Convex Optimization
CMU-10725
Ellipsoid Methods

Barnabás Póczos & Ryan Tibshirani
Outline

- Linear programs
- Simplex algorithm
- Running time: Polynomial or Exponential?
- Cutting planes & Ellipsoid methods for LP
- Cutting planes & Ellipsoid methods for unconstrained minimization
Books to Read

David G. Luenberger, Yinyu Ye: Linear and Nonlinear Programming

Boyd and Vandenberghe: Convex Optimization
Back to Linear Programs

**Inequality form** of LPs using matrix notation:

\[
\begin{align*}
\min \text{[or } \max \text{]} & \quad c^T x \\
\text{s.t.} & \quad A x \leq b \\
& \quad l \leq x \leq u
\end{align*}
\]

\[
A \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m,
\]

\[
l \in \mathbb{R}^n, \quad u \in \mathbb{R}^n
\]

**Standard form** of LPs:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0 \\
& \quad l \leq x \leq u
\end{align*}
\]

\[
x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m
\]

We already know: Any LP can be rewritten to an equivalent standard LP
Motivation

Linear programs can be viewed in two somewhat complementary ways:

- **continuous optimization**: continuous variables, convex feasible region continuous objective function

- **combinatorial problems**: solutions can be found among the vertices of the convex polyhedron defined by the constraints

**Issues with combinatorial search methods**: number of vertices may be exponentially large, making direct search impossible for even modest size problems

\[ n \text{ variables and } m \text{ constraints: } \frac{n!}{m!(n-m)!} \text{ vertices.} \]
History
Simplex Method

**Simplex method:**

- Jumping from one vertex to another, it improves values of the objective as the process reaches an optimal point.

- It performs well in practice, visiting only a small fraction of the total number of vertices.

- **Running time?** Polynomial? or Exponential?
The Simplex method is not polynomial time

- Dantzig observed that for problems with $m \leq 50$ and $n \leq 200$ the number of iterations is ordinarily less than $1.5m$.

- That time many researchers believed (and tried to prove) that the simplex algorithm is polynomial in the size of the problem $(n,m)$.

- In 1972, Klee and Minty showed by examples that for certain linear programs the simplex method will examine every vertex.

- These examples proved that in the worst case, the simplex method requires a number of steps that is exponential in the size of the problem.
The Simplex method is not polynomial time

Klee–Minty example

\[
\begin{align*}
\text{MAX} & \quad \sum_{j=1}^{n} 10^{n-j} x_j \\
\text{subject to} & \quad 2 \sum_{j=1}^{n} 10^{i-j} x_j + x_i \leq 100^{i-1} \quad i=1, \ldots, n \\
& \quad x_j \geq 0 \quad j=1, \ldots, n
\end{align*}
\]

After standardizing this with slack variables:

2n nonnegative variables, n constraints

2^{n-1} pivot steps
**Klee–Minty example**

maximize \( 100x_1 + 10x_2 + x_3 \)
subject to \( x_1 \leq 1 \)
\( 20x_1 + x_2 \leq 100 \)
\( 200x_1 + 20x_2 + x_3 \leq 10000 \)
\( x_1, x_2, x_3 \geq 0 \)

Initial tableau:

<table>
<thead>
<tr>
<th>( z )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-100</td>
<td>-10</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0 = ( z )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1 = ( s_1 )</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100 = ( s_2 )</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10000 = ( s_3 )</td>
</tr>
</tbody>
</table>

First pivot: \( x_1 \) enters, \( s_1 \) leaves the basis.

<table>
<thead>
<tr>
<th>( z )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>-10</td>
<td>-1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>100 = ( z )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1 = ( x_1 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-20</td>
<td>1</td>
<td>0</td>
<td>80 = ( s_2 )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>20</td>
<td>1</td>
<td>-200</td>
<td>0</td>
<td>1</td>
<td>9800 = ( s_3 )</td>
</tr>
</tbody>
</table>

\( x_2 \) enters, \( s_2 \) leaves

Klee–Minty example

Second pivot: $x_2$ enters, $s_2$ leaves.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>−100</td>
<td>10</td>
<td>0</td>
<td>900</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>−20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>80</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>200</td>
<td>−20</td>
<td>1</td>
<td>8200</td>
</tr>
</tbody>
</table>

Third pivot: $s_1$ enters, $x_1$ leaves.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>−200</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−20</td>
<td>1</td>
<td>8000</td>
</tr>
</tbody>
</table>

Fourth pivot: $x_3$ enters, $s_3$ leaves.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>−100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−10</td>
<td>1</td>
<td>9000</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>−200</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>−20</td>
<td>1</td>
<td>8000</td>
</tr>
</tbody>
</table>

$x_1$ enters, $s_1$ leaves
Klee–Minty example

Fifth pivot: $x_1$ enters, $s_1$ leaves.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>-10</td>
<td>1</td>
<td>9100 =  $z$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1 =  $x_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-20</td>
<td>1</td>
<td>0</td>
<td>80 =  $x_2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>200</td>
<td>-20</td>
<td>1</td>
<td>8200 =  $x_3$</td>
</tr>
</tbody>
</table>

Sixth pivot: $s_2$ enters, $x_2$ leaves

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>-100</td>
<td>0</td>
<td>1</td>
<td>9900 =  $z$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1 =  $x_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-20</td>
<td>1</td>
<td>0</td>
<td>80 =  $s_2$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>20</td>
<td>1</td>
<td>-200</td>
<td>0</td>
<td>1</td>
<td>9800 =  $x_3$</td>
</tr>
</tbody>
</table>

Seventh pivot: $s_1$ enters, $x_1$ leaves.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>rhs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10000 =  $z$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1 =  $s_1$</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100 =  $s_2$</td>
</tr>
<tr>
<td>0</td>
<td>200</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10000 =  $x_3$</td>
</tr>
</tbody>
</table>

This is optimal.
Ellipsoid methods

Is it possible to construct polynomial time algorithms?

1979 Khachiyan’s ellipsoid method:

- It constructs a sequence of shrinking ellipsoids
- each of which contains the optimal solution set
- and each member of the sequence is smaller in volume than its predecessor by at least a certain fixed factor.

Khachiyan proved that the ellipsoid method is a polynomial-time algorithm for linear programming!

Practical experience, however, was disappointing. ..
In almost all cases, the simplex method was much faster than the ellipsoid method!

Is there an algorithm that, in practice, is faster than the simplex method?
Polynomial time methods

1984 Karmarkar:
a new polynomial time algorithm, an interior-point method, with the potential to improve the practical effectiveness of the simplex method

It is quite complicated, and we don’t have time to discuss it.
Patent issues...
The Ellipsoid method for Linear Programs
The feasibility problem: 

$$\Omega = \{ y \in \mathbb{R}^m : y^T \alpha_j \leq c_j, \ j = 1 \ldots n \}$$

**Goal**: finding a point of a polyhedral set $\Omega$ given by a system of linear inequalities.
Solving LP = Solving feasibility problem

The feasibility problem:

$$\Omega = \{ y \in \mathbb{R}^m : y^T a_j \leq c_j \quad j = 1 \ldots n \}$$

One can prove that finding a point $y$ in $\Omega$ is equivalent to solving a linear programming problem.

- It is trivial that LP can be used to solve the feasibility problem
- We already know that Simplex method Phase 2 can be used to solve the feasibility problem
- From duality theory we will see later the feasibility problem can indeed be used to solve arbitrary LPs
Two assumptions:

(A1) \( \Omega \) can be covered with a finite ball of radius \( R \)

\[ \Omega \subseteq \left\{ y \in \mathbb{R}^m : |y - y_0| \leq R \right\} = S(y_0, R) \]

\( R \) is known to us

\( y_0 \) is known to us

(A2) There is a ball with radius \( r \) inside of \( \Omega \)

\[ S(y^*, r) \subseteq \Omega \]

We don’t need to know \( y^* \)

\( r \) only need to decide when to stop the algorithm
Ellipsoids

In what follows we will need ellipsoids.

**Definition**: [Ellipsoid]

\[
E = \left\{ x \in \mathbb{R}^m : (x - z)^\top Q (x - z) \leq 1 \right\}
\]

\(z \in \mathbb{R}^m\) CENTER

\(Q \geq 0\) POS DEF \(Q \in \mathbb{R}^{m \times m}\)

**Properties**:

Axes of ellipsoid:

EIGENVECTORS OF \(Q\)

\(\lambda_1^{-\frac{1}{2}}, \lambda_2^{-\frac{1}{2}}, \ldots, \lambda_m^{-\frac{1}{2}}\)

\(\{\lambda_i\}_{i=1}^m\) EIGENVALUES OF \(Q\)

Volume of ellipsoid:

\[
Vol(E) = Vol(s(0, I)) \prod_{i=1}^m \lambda_i^{-\frac{1}{2}} = Vol(s(0, I)) \det(Q^{-\frac{1}{2}})
\]
Cutting Plane method and covering ellipsoid

\[ \Omega = \{ y \in \mathbb{R}^n : y^T \alpha_j \leq c_j \} \quad j = 1 \ldots n \]

In the ellipsoid method, a series of ellipsoids \( \mathcal{E}_k \) is defined.

**Centers:** \( y_k \). **Parameter matrix:** \( Q = B_k^{-1} \), where \( B_k \succ 0 \).

At each iteration of the algorithm, we have \( \Omega \subset \mathcal{E}_k \).

It is then possible to check whether \( y_k \in \Omega \). [Center of \( \mathcal{E}_k \)]

If so, we have found an element of \( \Omega \) as required.

If not, there is at least one constraint that violates \( y_k \).

Suppose it is the \( jth \) constraint: \( a_j^T y_k \geq c_j \)

Then, \( \Omega \subset \mathcal{E}_k \cap \{ y : a_j^T y \leq a_j^T y_k \} \)

Then, \( \Omega \subset \mathcal{E}_k \cap \text{Half Space Border on } y_k \)
Suppose $a_j^T y_k > c_j \quad \Omega \subset \mathcal{E}_k$.

$$\Omega = \{ y \in \mathbb{R}^n : y^T a_j \leq c_j \mid j = 1 \ldots n \}$$

$$\Omega \subset \frac{1}{2} \mathcal{E}_k = \left\{ y \in \mathcal{E}_k : a_j^T y \leq a_j^T y_k \right\}$$

The successor ellipsoid $\mathcal{E}_{k+1}$ is defined to be the minimal-volume ellipsoid containing $\frac{1}{2} \mathcal{E}_k$. 
The successor ellipsoid $\mathcal{E}_{k+1}$ is defined to be the minimal-volume ellipsoid containing $\frac{1}{2} \mathcal{E}_k$.

**It is constructed as follows:**

Define

$$\tau = \frac{1}{m+1}, \quad \delta = \frac{m^2}{m^2 - 1}, \quad \sigma = 2\tau.$$  

$$y_{k+1} = y_k - \frac{\tau}{(a_j^T B_k a_j)^{1/2}} B_k a_j$$  

$$B_{k+1} = \delta \left( B_k - \sigma \frac{B_k a_j a_j^T B_k}{a_j^T B_k a_j} \right)$$
Theorem: [Ratio of volumes]

The ellipsoid $E_{q+1} = E(y_{b+1}, B_{q+1}^{-1})$ defined as above contains the set $\frac{1}{2} E_b$

Moreover,

$$\frac{\text{Vol}(E_{q+1})}{\text{Vol}(E_b)} = \left(\frac{m^2}{m^2-1}\right)^{(m-1)/2} \frac{m}{m+1} < \exp\left(-\frac{1}{2(m+1)}\right) < 1$$
Convergence

Initial step

\[ \mathcal{O} \leq \{ y \in \mathbb{R}^m : |y - y_0| \leq R \} \]

- \( R \) is known to us
- \( y_0 \) is known to us

We start the ellipsoid method from the \( S(y_0,R) \) ellipsoid \([=\text{sphere}]\)

\[
\frac{\text{Vol}(E_{a+1})}{\text{Vol}(E_a)} < \exp\left(-\frac{1}{2(2m+1)}\right)
\]

\[
\frac{\text{Vol}(E_{2m})}{\text{Vol}(E_0)} \quad \frac{\text{Vol}(E_{2m})}{\text{Vol}(E_{2m-1})} \quad \frac{\text{Vol}(E_{1})}{\text{Vol}(E_{0})} < \exp\left(-\frac{2m}{2(2m+1)}\right) < \frac{1}{2}
\]

in \( O(m) \) iterations the ellipsoid method can reduce the volume of an ellipsoid to one-half of its initial value.
Convergence rate of the ellipsoid method

How many iterations we need to get into $S(y^*, r)$?

We start the ellipsoid method from the $S(y_0, R)$ ellipsoid [sphere]

$$\text{Vol}(E_0) = C R^n$$  \text{[WE START HERE]}

$$\text{Vol}(E_k) \leq C \tau^m$$  \text{[WE WANT TO GET HERE]}

$$\frac{\text{Vol}(E_0)}{\text{Vol}(E_k)} \leq \left( \frac{\tau}{R} \right)^m \leq \left( \frac{1}{2} \right)^m \Rightarrow m \log \frac{R}{r} \leq \frac{b_2}{m} \log \frac{1}{2}$$

This is what we want.

In $m$ steps, we can halve the volume.

$$\Rightarrow m \log \frac{R}{r} > \frac{b_2}{m} \log 2$$

Hence we can reduce the volume to less than that of a sphere of radius $r$ in $O(m^2 \log(R/r))$ iterations.

A single iteration of the ellipsoid method requires $O(m^2)$ operations. Hence the entire process requires $O(m^4 \log(R/r))$ operations.
Ellipsoid Method for General LP

**Linear programs:** \( \Delta \in \mathbb{R}^{m \times n} \)

(P) \[ \max_{x} \mathbf{c}^T x \quad \text{subject to} \quad A x \leq b \quad \mathbf{c}^T x \geq 0 \]

(O) \[ \min_{y} \mathbf{y}^T b \quad \text{subject to} \quad \mathbf{y}^T A \geq \mathbf{c}^T \quad \mathbf{y} \geq 0 \]

From duality theory we know that both problems can be solved by finding a feasible point to inequalities

\[ \mathbf{c}^T x \leq \mathbf{c}^T y \ \forall x, y \ \text{feasible} \]

\[ \mathbf{c}^T x^* = \mathbf{w}^T y^* \]

Thus, the total number of arithmetic operations for solving a linear program is bounded by:

\[ \mathcal{O} \left( (m+n)^4 \log \left( \frac{R}{\epsilon} \right) \right) \]
Goal: \[ \min_x f_0(x) \]

Start from a big enough initial ellipsoid:
\[ \mathcal{E}_0 = \left\{ z \in \mathbb{R}^n : (z - x_0)^T P_0^{-1} (z - x_0) \leq 1 \right\} \subset \mathbb{R}^n \]

such that \( x^* \in \mathcal{E}_0 \)

where \( P_0 > 0 \) and \( x_0 \) is the center of \( \mathcal{E}_0 \).

At the \( k \)th iteration of the algorithm, we have
\[ x_k: \text{ the the center of an ellipsoid} \]
\[ P_k > 0 \]
\[ x^* \in \mathcal{E}_k = \left\{ x \in \mathbb{R}^n : (x - x_k)^T P_k^{-1} (x - x_k) \leq 1 \right\} \]
**Observation:** [subgradient]

If \( g_{k+1} \in \partial f(x_k) \), then \( f(y) \geq f(x_k) + g_{k+1}^T(y - x_k), \quad \forall y \)

Therefore,

\[
g_{k+1}^T(x^* - x_k) \leq f(x^*) - f(x_{k+1}) \leq 0.
\]

If we have access to the subgradient, then we can use this as a normal vector of the cutting plane!
We already know:

\[ x^* \in \mathcal{E}_k = \left\{ x \in \mathbb{R}^n : (x - x_k)^T P_k^{-1} (x - x_k) \leq 1 \right\} \]

\[ g_{k+1}^T (x^* - x_k) \leq f(x^*) - f(x_{k+1}) \leq 0. \]

Therefore,

\[ x^* \in \mathcal{E}_k \cap \{ z : g^{(k+1)}^T (z - x^{(k)}) \leq 0 \} \]

Set \( \mathcal{E}_{k+1} \) to be the ellipsoid of minimal volume containing this half-ellipsoid. \( \Rightarrow x_{k+1}, P_{k+1} \)

Stop when \( f(x_k) - f(x^*) \leq \epsilon \)
Ellipsoid method


Update rule:

\[ x_{k+1} = x_k - \frac{1}{n+1} P_k \tilde{g}_{k+1} \]
\[ P_{k+1} = \frac{n^2}{n^2 - 1} \left( P_k - \frac{2}{n+1} P_k \tilde{g}_{k+1} \tilde{g}_{k+1}^T P_k \right) \]

where \( \tilde{g}_{k+1} = \frac{g_{k+1}}{\sqrt{g_{k+1}^T P g_{k+1}}} \).
Cutting planes can be used for constrained problems as well.

We don’t have time to discuss them...

Instead we will use penalty and barrier functions to handle constraints.
Summary

- Linear programs
- Simplex algorithm
- Klee–Minty example
- Cutting planes & Ellipsoid methods for LP
- Polynomial rate
- Cutting planes & Ellipsoid methods for unconstrained minimization