Convex Optimization
CMU-10725

Ellipsoid Methods

Barnabás Póczos & Ryan Tibshirani

MACHINE LEARNING DEPARTMENT
Carnegie Mellon
School of Computer Science

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:

$$\text{MIN [OR MAX]} \quad c^T x \qquad c \in \mathbb{R}^n$$

$$\text{S.T} \quad Ax \le b \qquad A \in \mathbb{R}^{m \times n} \quad b \in \mathbb{R}^m$$

$$l \le x \le u \qquad l \in \mathbb{R}^n \quad u \in \mathbb{R}^n$$

$$x \in \mathbb{R}^n$$

Standard form of LPs:

$$A \in \mathbb{R}^{m \times n} \qquad n \ge m$$

$$\text{MIN} \quad c^T x \qquad c \in \mathbb{R}^n$$

$$\text{S.T} \quad Ax = b$$

$$x \ge 0 \qquad x \in \mathbb{R}^n_{+}$$

$$b \in \mathbb{R}^m_{+}$$

SOMETIMES $b \ge 0$ IS NOT REQUIRED

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 variables and m constraints: $\frac{n!}{m!(n-m)!}$ 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

$$\max_{x_1 \ldots x_n} \sum_{j=1}^{n} 10^{n-j} x_j$$

SUBJECT TO

$$2 \sum_{j=1}^{i} 10^{i-j} x_j + x_i \le 100^{i-1} \qquad i = 1 \ldots n$$

$$x_j \geq 0 \qquad j = 1 \ldots n$$

After standardizing this with slack variables:

2n nonnegative variables, n constraints

2^n-1 pivot steps

Klee–Minty example

$$
\begin{array}{ll}
\text{maximize} & 100x_1 + 10x_2 + x_3 \\
\text{subject to} & x_1 \le 1 \\
& 20x_1 + x_2 \le 100 \\
& 200x_1 + 20x_2 + x_3 \le 10000 \\
& x_1, x_2, x_3 \ge 0
\end{array}
$$

Initial tableau:

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	-100	-10	-1	0	0	0	0	$=$	z
0	1	0	0	1	0	0	1	$=$	s_1
0	20	1	0	0	1	0	100	$=$	s_2
0	200	20	1	0	0	1	10000	$=$	s_3

First pivot: x_1 enters, s_1 leaves the basis.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	0	-10	-1	100	0	0	100	$=$	z
0	1	0	0	1	0	0	1	$=$	x_1
0	0	1	0	-20	1	0	80	$=$	s_2
0	0	20	1	-200	0	1	9800	$=$	s_3

x_2 enters, s_2 leaves

http://www.math.ubc.ca/~israel/m340/kleemin3.pdf

Klee–Minty example

Second pivot: x_2 enters, s_2 leaves.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	0	0	-1	-100	10	0	900	=	z
0	1	0	0	1	0	0	1	=	x_1
0	0	1	0	-20	1	0	80	=	x_2
0	0	0	1	200	-20	1	8200	=	s_3

Third pivot: s_1 enters, x_1 leaves.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	100	0	-1	0	10	0	1000	=	z
0	1	0	0	1	0	0	1	=	s_1
0	20	1	0	0	1	0	100	=	x_2
0	-200	0	1	0	-20	1	8000	=	s_3

Fourth pivot: x_3 enters, s_3 leaves.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	-100	0	0	0	-10	1	9000	=	z
0	1	0	0	1	0	0	1	=	s_1
0	20	1	0	0	1	0	100	=	x_2
0	-200	0	1	0	-20	1	8000	=	x_3

x_1 enters, s_1 leaves

Klee–Minty example

Fifth pivot: x_1 enters, s_1 leaves.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	0	0	0	100	-10	1	9100	$=$	z
0	1	0	0	1	0	0	1	$=$	x_1
0	0	1	0	-20	1	0	80	$=$	x_2
0	0	0	1	200	-20	1	8200	$=$	x_3

Sixth pivot: s_2 enters, x_2 leaves

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	0	10	0	-100	0	1	9900	$=$	z
0	1	0	0	1	0	0	1	$=$	x_1
0	0	1	0	-20	1	0	80	$=$	s_2
0	0	20	1	-200	0	1	9800	$=$	x_3

Seventh pivot: s_1 enters, x_1 leaves.

z	x_1	x_2	x_3	s_1	s_2	s_3	rhs		
1	100	10	0	0	0	1	10000	$=$	z
0	1	0	0	1	0	0	1	$=$	s_1
0	20	1	0	0	1	0	100	$=$	s_2
0	200	20	1	0	0	1	10000	$=$	x_3

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 for LP

The feasibility problem:

$$\Omega = \{ y \in \mathbb{R}^m : y^T a_j \le c_j , \; j = 1 \dots n \}$$

Goal: finding a point of a polyhedral set Ω given by a system of linear inequalities.

Solving LP = Solving feasibility problem

The feasibility problem:

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

One can prove that finding a point y in Ω 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

The Ellipsoid method

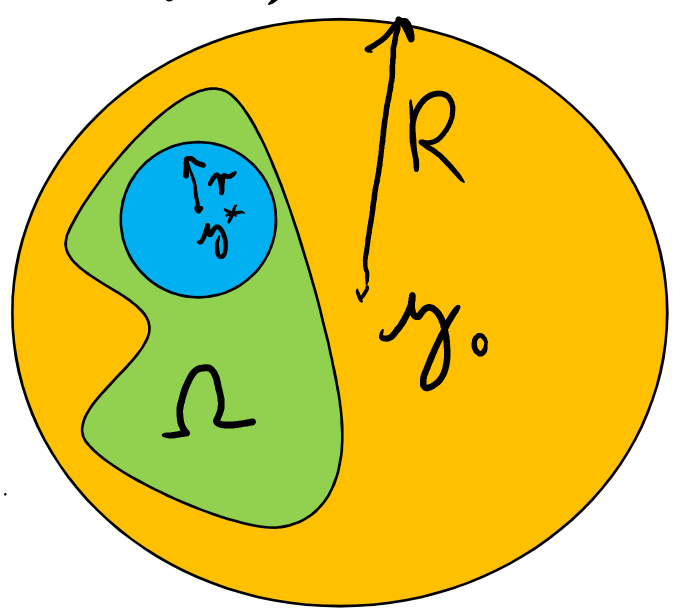
Two assumptions:

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

$$\Omega \subseteq \{y \in \mathbb{R}^m : |y - y_0| \le R\} = S(y_0, R)$$

R IS KNOWN TO US

y_0 IS KNOWN TO US



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

$$S(y^*, r) \subset \Omega$$

WE DON'T NEED TO KNOW y^*

r ONLY NEED TO DECIDE WHEN TO STOP THE ALG

Ellipsoids

In what follows we will need ellipsoids.

Definition: [Ellipsoid]

$$\mathcal{E} = \{ y \in \mathbb{R}^m : (y - z)^T Q (y - z) \leq 1 \}$$

$z \in \mathbb{R}^m$: CENTER

$Q \succ 0$ POS DEF $Q \in \mathbb{R}^{m \times m}$

Properties:

Axes of ellipsoid: EIGENVECTORS OF Q

Lengths of the axes: $\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots \lambda_m^{-1/2}$

$\{\lambda_i\}_{i=1}^m$: EIGENVALUES OF Q

Volume of ellipsoid:

$$\mathrm{VOL}(\mathcal{E}) = \mathrm{VOL}(S(0,1)) \prod_{i=1}^m \lambda_i^{-1/2} = \mathrm{VOL}(S(0,1)) \, \mathrm{DET}(Q^{-1/2})$$

Cutting Plane method and covering ellipsoid

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

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 Ω as required.

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

Suppose it is the jth constraint: $a_j^T y_k > c_j$

Then,

$$\Omega \subset \mathcal{E}_k \cap \underbrace{\{ y : a_j^T y \le a_j^T y_k \}}_{\text{HALF SPACE BORDER ON } y_k}$$

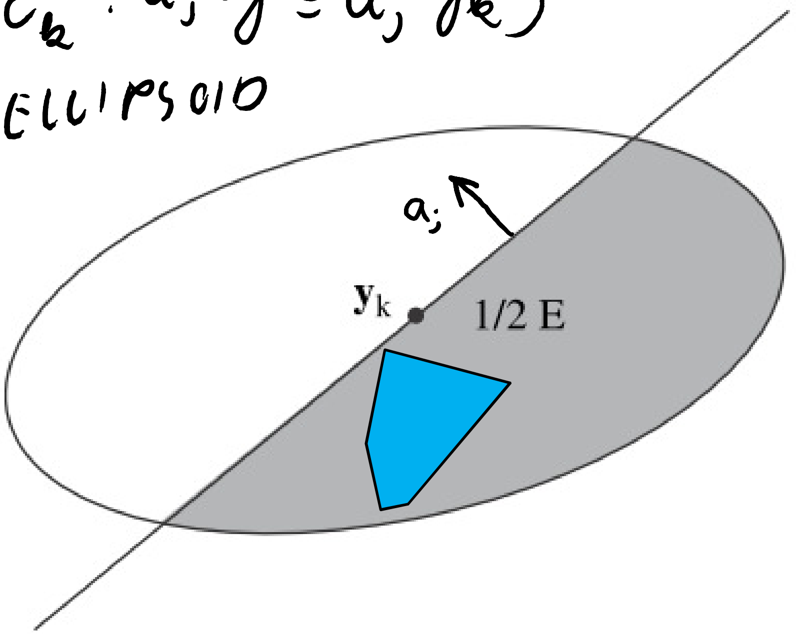
Cutting Plane and New Containing Ellipsoid

Suppose $a_j^T y_k > c_j$ $\quad \Omega \subset \mathcal{E}_k.$

$$\Omega = \{ y \in \mathbb{R}^m : y^T a_j \le c_j , \; j = 1 \dots n \}$$

$$\Omega \subset \tfrac{1}{2}\mathcal{E}_k \doteq \{ y \in \mathcal{E}_k : a_j^T y \le a_j^T y_k \}$$

HALF ELLIPSOID



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

Cutting Plane and New Containing Ellipsoid

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

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

$$\mathbf{y}_{k+1} = \mathbf{y}_k - \frac{\tau}{(\mathbf{a}_j^T \mathbf{B}_k \mathbf{a}_j)^{1/2}} \mathbf{B}_k \mathbf{a}_j$$

$$\mathbf{B}_{k+1} = \delta \left(\mathbf{B}_k - \sigma \frac{\mathbf{B}_k \mathbf{a}_j \mathbf{a}_j^T \mathbf{B}_k}{\mathbf{a}_j^T \mathbf{B}_k \mathbf{a}_j} \right)$$

Cutting Plane and New Containing Ellipsoid

Theorem: [Ratio of volumes]

THE ELLIPSOID $\mathcal{E}_{k+1} = \mathcal{E}(y_{k+1}, B_{k+1}^{-1})$ DEFINED AS ABOVE CONTAINS THE SET $\frac{1}{2}\mathcal{E}_k$

MOREOVER,

$$\frac{\mathrm{VOL}(\mathcal{E}_{k+1})}{\mathrm{VOL}(\mathcal{E}_k)} = \left(\frac{m^2}{m^2-1}\right)^{(m-1)/2} \frac{m}{m+1} < \exp\left(-\frac{1}{2(m+1)}\right) < 1$$

Convergence

Initial step

$$\Omega \subseteq \{y \in \mathbb{R}^m : |y - y_0| \leq R\}$$

R IS KNOWN TO US

y_0 IS KNOWN TO US

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

$$\frac{\mathrm{VOL}(\mathcal{E}_{k+1})}{\mathrm{VOL}(\mathcal{E}_k)} < \exp\left(-\frac{1}{2(m+1)}\right)$$

$$\frac{\mathrm{VOL}(\mathcal{E}_{2m})}{\mathrm{VOL}(\mathcal{E}_0)} = \frac{\mathrm{VOL}(\mathcal{E}_{2m})}{\mathrm{VOL}(\mathcal{E}_{2m-1})} \cdots \frac{\mathrm{VOL}(\mathcal{E}_1)}{\mathrm{VOL}(\mathcal{E}_0)} < \exp\left(\frac{-2m}{2(m+1)}\right) < \frac{1}{2}$$

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]

$VOL(\mathcal{E}_0) = cR^m$ [WE START HERE]

$VOL(\mathcal{E}_k) \leq c\,r^m$ [WE WANT TO GET HERE]

$$\frac{VOL(\mathcal{E}_k)}{VOL(\mathcal{E}_0)} \leq \left(\frac{r}{R}\right)^m \leq \left(\frac{1}{2}\right)^{\frac{k}{m}} \Rightarrow m \log\frac{r}{R} \leq \frac{k}{m}\log\frac{1}{2}$$

$$\Rightarrow m\log\frac{R}{r} \geq \frac{k}{m}\log 2$$

THIS IS WHAT WE WANT

IN m STEPS WE CAN HALF THE VOLUME

WE USE k STEPS

$$\Rightarrow k \leq O\left(m^2 \log\frac{R}{r}\right)$$

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: $A \in \mathbb{R}^{m \times n}$

(P) MAX $c^T x$
SUBJECT TO $Ax \le b$
$x \ge 0$

(D) MIN $y^T b$
SUBJECT TO $y^T A \ge c^T$
$y \ge 0$

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

$$c^T x \le b^T y \quad \forall x, y \text{ FEASIBLE}$$

$$c^T x^* = b^T y^*$$

$$\Rightarrow \begin{cases} -c^T x + b^T y \le 0 \\ Ax \le b \\ -A^T y \le -c \\ x, y \ge 0 \end{cases}$$

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

$$O\left((m+n)^4 \log(R/r)\right)$$

Ellipsoid method for Unconstrained Convex Opt.

Ellipsoid method for Unconstrained Convex Opt.

Goal: $\min_x f_0(x)$

Start from a big enough initial ellipsoid:

$$\mathcal{E}_0 = \left\{ z \in \mathbb{R}^n : (z - x_0)^T P_0^{-1} (z - x_0) \le 1 \right\} \subset \mathbb{R}^n$$

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

where $P_0 \succ 0$ and x_0 is the center of $\mathcal{E}_0$.

At the kth iteration of the algorithm, we have

x_k: the the center of an ellipsoid

$P_k \succ 0$

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

Ellipsoid method for Unconstrained Convex Opt.

Observation: [subgradient]

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

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!

Ellipsoid method for Unconstrained Convex Opt.

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,

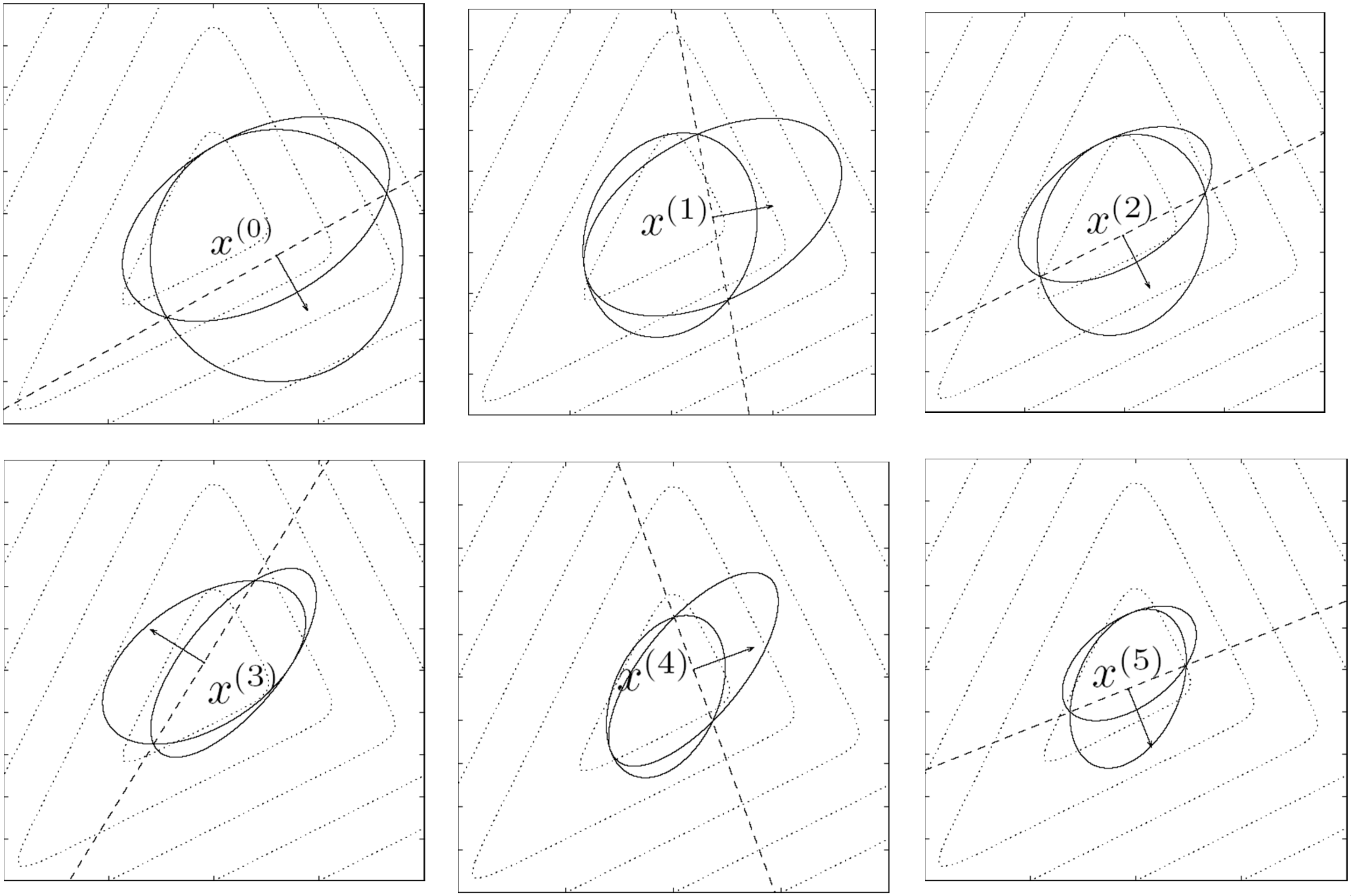
$$x^* \in \mathcal{E}_k \cap \{z : g^{(k+1)T}(z - x^{(k)}) \leq 0\}$$

ELLIPSOID HALF SPACE

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



http://en.wikipedia.org/wiki/Ellipsoidal_method

Ellipsoid method for Unconstrained Convex Opt.

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 for Constrained minimization

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