Optimization Methods		Date:	*16 Feb '18*
Instructor: *Sujit Prakash Gujar*		Scribes:	Aditya Bharti Avijit Dasgupta Srija Bhupathiraju

Lecture 10: Introduction to the Ellipsoid Method

1 Recap

1.1 Dual of a Linear Program

- Every linear program (referred to as the *primal* form) can be formulated in an equivalent *dual* form.
- The number of variables in the primal equals the number of constraints in the dual.
- The number of constraints in the primal equals the number of variables in the dual.
- The conversion is useful when the time complexity of a particular solution depends upon the number of variables in a program, and reducing the number of variables would be advantageous.
- The conversion from primal to dual can be easily done using the dualization recipe in Table 1.

1.2 Weak Duality

Theorem 1.1 (Weak Duality)**.** *Let P denote the primal form of a linear program and let D be its dual. Further, let $F(P)$ denote the set of feasible solutions of the primal and $F(D)$ be the set of feasible solutions of the dual.*
For any $x \in F(P)$ and $y \in F(D)$, we have

$$c^T x \leq b^T y,$$

that is, the primal objective is always upper bounded by the dual objective.

Since optimal solutions are also feasible solutions, we have $c^T x^* \leq b^T y^*$, where x^* is optimal solution to primal and y^* is optimal solution to dual.

Table 1: Dualization Recipe

	Primal Linear Program	Dual Linear Program
Variables	$x_1, x_2, \ldots, x_n$	$y_1, y_2, \ldots, y_m$
Matrix	A	A^T
Right-hand side	$\mathbf{b}$	$\mathbf{c}$
Objective function	max $\mathbf{c}^T\mathbf{x}$	min $\mathbf{b}^T\mathbf{y}$
Constraints	i^{th} constraint has $\leq$	$y_i \geq 0$
	$\geq$	$y_i \leq 0$
	$=$	$y_i \in \mathbb{R}$
	$x_j \geq 0$	j^{th} constraint has $\geq$
	$x_j \leq 0$	$\leq$
	$x_j \in \mathbb{R}$	$=$

1.3 Strong Duality

Theorem 1.2 (Strong Duality)**.** *Let P be the primal form of a linear program and D be its corresponding dual. Then, the following 4 possibilities arise:*

1. *Neither P nor D has a feasible solution*
2. *P is unbounded and D is infeasible*
3. *P is infeasible and D is unbounded*
4. *Both P and D have feasible solutions*

In the last case, let x^* be the primal-optimal solution and let y^* be the dual optimal solution. We have

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

The optimal values of the primal and dual objective functions are equal.

1.4 Difficulty of finding an optimal vs. a feasible solution

Consider the following primal linear program:

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b \\ & \quad x \geq 0 \end{array}$$

with the following dual formulation:

$$\begin{array}{ll} \text{minimize} & b^T y \\ \text{subject to} & A^T y \geq c \\ & \quad y \geq 0 \end{array}$$

Consider another linear program formulated as:

$$\begin{array}{lll} \text{maximize} & c^T x & \\ \text{subject to} & Ax & \leq b \\ & -A^T y & \leq c \\ & b^T y - c^T x & \leq 0 \\ & x & \geq 0 \\ & y & \geq 0 \end{array}$$

Notice that the number of variables in the new program has increased from n variables in the primal to $n + m$ variables, where m is the number of constraints in the primal. Thus, the problem size has increased only polynomially, not exponentially.

- Let $(\widetilde{x}, \widetilde{y})$ be a feasible solution to this new linear program, found by some algorithm. Since it must satisfy the constraints, we have $b^T \widetilde{x} \leq c^T \widetilde{y}$.
- Now, $\widetilde{x}$ is a feasible solution to primal and $\widetilde{y}$ is a feasible solution to primal. From theorem 1.1 of weak duality, we get $c^T \widetilde{x} \leq b^T \widetilde{y}$.
- Therefore, $c^T \widetilde{x} = b^T \widetilde{y}$ and $\widetilde{x}$ and $\widetilde{y}$ are optimal solutions to primal and dual respectively.
- Therefore, any algorithm for finding feasible solutions to a linear program can be adapted to find optimal solutions to a linear program.
- The problem of finding feasible solutions is as hard as finding optimal solutions.

The fact that an algorithm for finding feasible solutions can be used to find optimal solutions will be used in the ellipsoid method.

2 Ellipsoid Method

2.1 Motivation

- The simplex method has exponential worst case complexity. It can visit all possible vertices of the polyhedron before finding optimal solution.
- For example, using Dantzig's original pivot rule, the simplex method visits every vertex of the Klee-Minty cube. Further, for every pivot rule, problems have been formulated which demostrate the worst case complexity of the simplex algorithm.
- Instead of visiting vertices to find the optimal solution, the ellipsoid method finds feasbile solutions by partitioning the solution search space. From the discussion in the recap, the feasible solutions can then be used to find optimal solutions for other linear programs.
- In every iteration of the algorithm, some portion of the search space is discarded as having no feasible solutions.
- The algorithm halts when the search space has been reduced to less than a threshold volume (no feasible solution) or when a feasible solution has been found out.

2.2 Formalism

In this section, we define an ellipsoid and describe some of its useful properties.

Ball: A ball $B(c, r)$ in $\mathbb{R}^n$ centered at $c \in \mathbb{R}^n$ with radius r is defined as follows:

$$B(c, r) := \{x \in \mathbb{R}^n : (x - c)^T (x - c) \leq r^2\}, \tag{1}$$

If $c = 0$ and $r = 1$, then we call $B(0, 1)$ a unit ball. A ball is a convex set.

Ellipsoid: An ellipsoid E centered at the origin is an image $L(B(0, 1))$ of the unit ball under an invertible linear transformation $L : \mathbb{R}^n \to \mathbb{R}^n$. Mathematically,

$$\begin{aligned} E = L(B(0,1) &= \{Lx : x \in B(0,1)\}) \\ &= \{y : L^{-1}y \in B(0,1)\} \\ &= \{y : (L^{-1}y)^T L^{-1}y \leq 1\} \\ &= \{y : y^T (LL^T)^{-1} y \leq 1\} \\ &= \{y : y^T Q^{-1} y \leq 1\} \end{aligned} \tag{2}$$

,

where $Q = LL^T$. Q is a positive definite matrix. However, linear transformations are not enough to describe all ellipsoids as the center of the ellipsoid does not necessarily have to be at the origin. To account for the translation of the center, we require and *affine* rather than a linear mapping.

Since translation cannot be written as a matrix multiplication in n-dimensions, the equation of the general ellipsoid centered at $c \in \mathbb{R}^n$ is obtained by a shifting of coordinates:

$$E(c, Q) := \{y : (y - c)^T Q^{-1} (y - c) \leq 1\} \tag{3}$$

2.3 Why Ellipsoids

In this section, some explanations are given as to why ellipsoids were chosen as the basis for this algorithm. This list is not rigorous.

- In a linear program, the feasible region is bounded (not necessarily *completely* bounded) by a set of hyperplanes defined by the constraints.
- The feasible region is thus the intersection of a set of half-spaces, which themselves are convex sets. Thus the feasible region itself is a convex set.
- At every iteration, the search space represents an estimate of the feasible region. Thus, the search space must be a convex set.
- In the ellipsoid algorithm described in the following section 2.4, a separating hyperplane is used to reduce the search space and discard the infeasible region. Since convexity is a necessary requirement for the separation hyperplane theorem, again the search space needs to be a convex set.
- An ellipsoid is an affine transformation of a sphere, which is a convex set. Hence, an ellipsoid is a convex set and fulfills the requirements of a candidate search space.
- In addition, ellipsoids have a mathematical representation that is simple and yet general enough so that the minimal ellipsoid covering a convex space can be calculated using a formula.

2.4 Ellipsoid Algorithm

A linear program (LP) solvability problem can be reduced to the LP feasbility problem in polynomial time as discussed in section 1.4. Since the ellipsoid method finds feasible solutions, it can be used to solve LPs. The ellipsoid algorithm also checks if a given convex set is empty.

Consider a general LP (L1) in standard form:

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b \\ & \;\; x \geq 0 \end{array}$$

Here, x is the vector of decision variables. Suppose, we are given the matrix A, b, and starting radius of a ball $R > \epsilon > 0$ where ϵ is some minimum threshold radius. We also assume a ball $P := \{x \in \mathbb{R}^n : Ax \leq B\}$ with radius R and centered at 0 as the starting search space.

Now, if P becomes a ball of radius ϵ then the ellipsoid algorithm has to return a point $y \in P$. If P contains no ball of radius ϵ, then the algorithm should either return some $y \in P$, or the answer NO SOLUTION.

The ellipsoid method generates a sequence of ellipsoids $E_0, E_1, ..., E_t$ where $P \subseteq E_k$ for each k. Let us denote the total number of bits required to represent a value x as $\langle x \rangle$. For example, $\langle A \rangle$ is the total number of bits required to represent each entry a_{ij} in A.

Let $L = \langle A \rangle + \langle b \rangle$ denote the input size for the system $Ax \leq b$.

If a solution to the LP exists and the solution is not too large, then this system has a solution if an only if the system

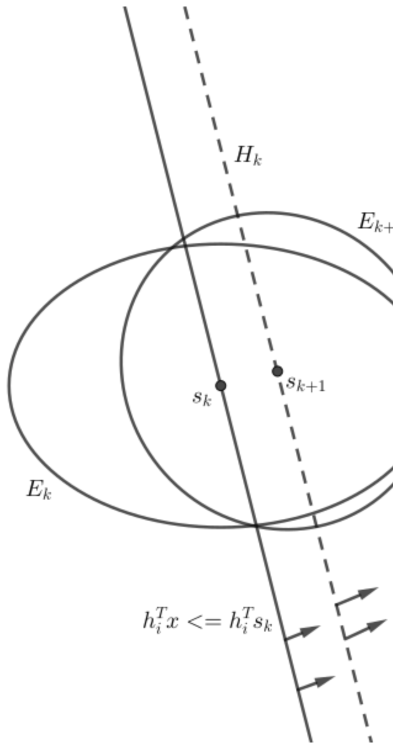

Figure 1: Evolution of ellipse in the ellipsoid method

$$
\begin{aligned}
Ax &\leq b \\
-K \leq x_1 &\leq K \\
-K \leq x_2 &\leq K \\
-K \leq x_3 &\leq K \\
&. \\
&. \\
&. \\
-K \leq x_n &\leq K
\end{aligned}
$$

has a solution, where $K = 2^L$. So, all the solutions of this system are contained in the ball $B(0, \sqrt{n}R)$, where the starting radius $R = K$. We can start from this ball.

The overall algorithm is as follows:

1. Set $k = 0$ and $E_0 = B(0, \sqrt{n}R)$
2. The current ellipsoid E_k be of the form $E_k := \{y : (y - s_k)^T Q^{-1}(y - s_k) \leq 1\}$. If s_k satisfies all inequalities of the system $Ax \leq b$ return s_k as a solution; stop. Otherwise, let h_k be the normal to a separating hyperplane (problem specific).
3. Otherwise, choose an inequality of the system that is violated by s_k. If the ith constraint is violated, define a new ellipsoid E_{k+1} which is the smallest possible volume containing the "half-ellipsoid" $H_k = E_k \cap \{x \in \mathbb{R}^n : h_i^T x \leq h_i^T s_k\}$ (see Figure 1).
4. If the volume of E_{k+1} is smaller than the volume of a ball of radius ϵ, return NO SOLUTION; stop. Otherwise, increase k by 1 and go to step 2.

It can be shown that we always have

$$\frac{vol(E_{k+1})}{vol(E_k)} \leq e^{\frac{1}{(2n+2)}} \tag{4}$$

The proof is beyond the scope of this course. Hence the volume of the ellipsoid E_k is at least $e^{\frac{k}{(2n+2)}}$ times smaller than the initial ball. Since the volume of an n-dimensional ball is proportional to the n^{th} power of the radius, for k satisfying $Re^{\frac{k}{(2n+2)}} \leq \epsilon$, the volume of E_k is smaller than that of a ball of radius ϵ. Therefore, the index k provides an upper bound of $\lceil n(2n+2)log(\frac{R}{\epsilon}) \rceil$ on the maximum number of iterations.