| | | | |
|---|---|---|---|
| Optimization Methods | | Date: | *16/02/2018* |
| Instructor: *Sujit Prakash Gujar* | | Scribes: | Amrit Chatterjee (201450802)<br>Poojitha Nandigam (201402182)<br>Madhur Aggarwal (201564235) |

# Lecture 10: Linear Programming - Ellipsoid Method

## 1 Recap

In the previous lecture, the concepts of strong and weak duality were covered. Strong Duality was proven using the Simplex Method. The Dualization recipe which can be used to convert a primal to a dual by following a specific set of rules was also explained. From the Dualization recipe one can easily prove the property that Dual of the Dual is the Primal itself. Finally, it was proven that finding the feasible solution of a Linear program is as hard as finding the optimal solution.

## 2 Introduction

The Ellipsoid Method can be best understood by an example of finding a Lion in the Sahara. We assume there is a lion in the Sahara desert and we need to find where it is, if at all there is any. We put boundaries on the whole desert so that the size of the desert is restricted by us. Using Ellipsoid Algorithm, we can find the lion, if it is there at all, if we follow the following steps:

1. Divide the Sahara in half with a fence.
2. Pick any one half and search for the lion. If you find the lion, our task is done.
3. Else, this half does not contain the lion. So, repeat with the nonempty half, divide it with a fence.
4. Continue until the fenced area is small enough (area smaller than the lion) so that either you will find the lion or there is no lion in the desert, as there is nowhere left for the lion to escape.

Let $V_0$ be the initial volume of the desert. After first step, the new volume of the fenced part of the desert under consideration will be $V_0/2$. Let $k$ be the number of iterations. So, after $k$ steps, the volume will be $V_0/2^k$. Let $\epsilon$ be the volume of the lion, Our stopping condition will be : $V_0/2^k < \epsilon$ Either we would have found out the lion by now or there is no lion as there is nowhere the lion can escape.

## 3 Definition of Ellipsoid

A simple definition of the ellipsoid can be given below.

$$E = \left\{\mathbf{x} \in \mathbb{R}^n : \left(\mathbf{x} - \mathbf{x}_0\right) A \left(\mathbf{x} - \mathbf{x}_0\right) \leq 1\right\} \tag{1}$$

Another way to look at an Ellipsoid is by considering it to be a stretched transformation of a sphere.

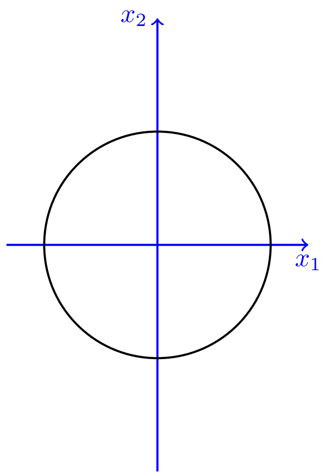



Figure 1: A sphere

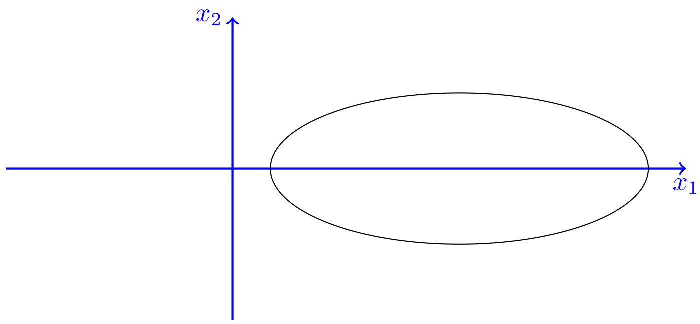



Figure 2: Stretched sphere after the transformation $M\mathbf{x}+\mathbf{s}$

The region inside a hypersphere centered at $\mathbf{0}$ and having a radius $R$ can be defined by

$$B_n(\mathbf{0}, R) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T\mathbf{x} \leq R^2\} \tag{2}$$

An $n$-dimensional ellipsoid is a set of the form

$$E = \left\{M\mathbf{x} + \mathbf{s} : \mathbf{x} \in B^n\right\} \tag{3}$$

where $M$ is a nonsingular $n \times n$ matrix and $s \in \mathbb{R}^n$ is a vector.

The mapping $\mathbf{x} \mapsto M\mathbf{x} + s$ consists of a linear function and a translation; this is called an affine-map. Based on the $\mathbf{y} = M\mathbf{x} + \mathbf{s}$ transformation, we can describe the ellipsoid as

$$\begin{aligned} E &= \{\mathbf{y} \in \mathbb{R}^n : M^{-1}\big(\mathbf{y} - \mathbf{s}\big) \in B^n\} \\ &= \{\mathbf{y} \in \mathbb{R}^n : (\mathbf{y} - \mathbf{s})^T\big(M^{-1}\big)^T M^{-1}\big(\mathbf{y} - \mathbf{s}\big) \leq R^2\} \end{aligned} \tag{4}$$

Substituting $Q = MM^T$ and normalizing by $R^2$ gives us the following definition.

$$E = \{\mathbf{y} \in \mathbb{R}^n : \big(\mathbf{y} - \mathbf{s}\big)^T Q^{-1}\big(\mathbf{y} - \mathbf{s}\big) \leq 1\} \tag{5}$$

Here, $Q$ is a positive definite matrix; this means that for all nonzero vectors $\mathbf{x}$ the symmetric square matrix $Q$ satisfies $\mathbf{x}^T Q\mathbf{x} > 0$.

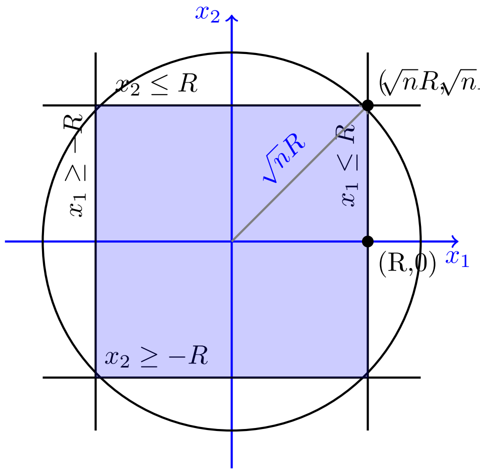
# 4 Key Concepts

There are certain decisions one needs to make before initiating the Ellipsoid method. Two of the most important ones are explained below:

1. Choosing an Initial Ellipsoid - An ellipsoid must be chosen such that it completely covers the convex polyhedron which is formed by the constraint hyperplanes from all sides. An example is shown below.

   Let the constraints be $|x_i| \leq R$, then our starting Ellipsoid should be

   $$E_0 = B_n(\mathbf{0}, \sqrt{n}R) \tag{6}$$

   We can generalize the above by taking the constraints $Ax \leq b$ and the parameter L which is the total number of bits required to represent each entry in the matrices.

   $$L = \langle A \rangle + \langle b \rangle \tag{7}$$

   It has been shown that there exists a feasible solution with $R = 2^L$.

2. Separation Oracle - This is an algorithm which returns a Yes if the input $\mathbf{x} \in \mathbb{R}^n$ is a solution, that is if $\mathbf{x} \in F(P)$ where $F(P)$ denotes the feasible region, and if $\mathbf{x}$ is not a solution, it returns one hyperplane that separates $\mathbf{x}$ and $F(P)$.

# 5 Ellipsoid Method

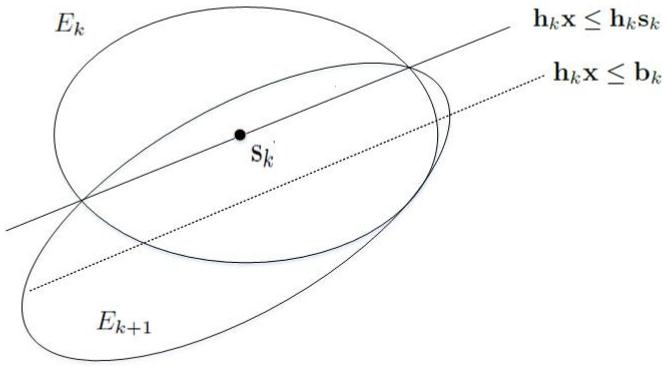
1. Start with a Basic Ellipsoid with center around the origin, also denoted here by $s_k$, where k=0 and $S_0$=0.

   $$E_0 = B_n(\mathbf{0}, \sqrt{n}R) \tag{8}$$

2. Does $s_k \in$ F(P) ?
   If Yes, then Stop and return the feasible solution.
   If No, then return the separating hyperplane $h_k$ which separates the feasible and the infeasible regions in the current Ellipsoid.
   Now, let us try to understand how we get this hyperplane $h_k$. Typically, the separation oracle is used in this step. A simple implementation of the separation oracle is to take $s_k$ and exercise all constraints. The first constraint which is not satisfied is the one that needs to be returned.

We need to do one more simple transformation to the hyperplane before we proceed to the next step. We need to draw a hyperplane parallel to $h_k$ and passing through $s_k$. This step is done to simplify the analysis of the algorithm.

$$\mathbf{h}_k\mathbf{x} \leq \mathbf{h}_k\mathbf{s}_k \tag{9}$$




3. The region below the hyperplane passing through the center in the figure above represents the region where the feasible region may be confined to. This is also called a half Ellipsoid and can be represented by the equation given below.

$$H_{k+1} = E_k \cap \{\mathbf{x} \in \mathbb{R}^n : \mathbf{h}_k\mathbf{x} \leq \mathbf{h}_k\mathbf{s}_k\} \tag{10}$$

We now need to draw a new Ellipsoid $E_{k+1}$ which can contain this half Ellipsoid completely. The equation of the this new Ellipsoid is given below.

$$\mathbf{s}_{k+1} = \mathbf{s}_k - \frac{1}{n+1}\frac{Q_k\mathbf{h}_k}{\sqrt{\mathbf{s}_k^T Q_k \mathbf{s}_k}} \tag{11}$$

$$Q_{k+1} = \frac{n^2}{n^2-1}\left(Q_k - \frac{2}{n+1}\frac{Q_k\mathbf{h}_k\mathbf{h}_k^T Q_k}{\mathbf{h}_k^T Q_k \mathbf{h}_k}\right) \tag{12}$$

4. If Vol$(E_k + 1) < \epsilon$, then Stop and return infeasible. Otherwise go back to step 2 with the Ellipsoid $E_{k+1}$ with center $S_{k+1}$.

# 6 Complexity of Ellipsoid Method

Mathematicians have proved that by using the Ellipsoid method, the volume is reduced by a constant factor between two consecutive steps.This can be expressed by the inequality below.

$$\frac{Vol(E_{k+1})}{Vol(E_k)} \leq e^{\frac{-1}{2n+2}} \tag{13}$$

Now, through a series of equations we will try to reach to an upper bound value for the maximum iterations for the Ellipsoid method. First, through an expansion of Step 4 of the Ellipsoid method, we can say that the terminating point of the algorithm comes when we draw a new ellipsoid smaller than a sphere of the radius $\epsilon$. For a $n$-dimensional sphere, the volume is proportional to n-times the radius. Thus, we can write the equation below.

$$Vol(E_{k+1}) \leq R^n e^{\frac{-1}{2n+2}} \leq \epsilon^n \tag{14}$$

If k $\geq n\big(2n+2\big)ln\big(R/\epsilon\big)$, we have either found a feasible solution or the problem is infeasible. Therefore the time complexity of the Ellipsoid method can be expressed as given below.

$$\bigcirc(n^2 ln(\frac{R}{\epsilon}) * Cost(SeparationOracle)) \tag{15}$$

From the above equation, we get a polynomial time complexity since the cost of Separation Oracle is less or a polynomial itself. Though, theoretically this algorithm should outperform the Simplex Method, but thats rarely the case in practical applications.

# References

[1] Bernd Grtner and Ji Matouek, *Understanding and Using Linear Programming*, Chapter 7: Not Only the Simplex Method.