Algorithmic Methods 08/11/2010

Lecture Notes 4: The Ellipsoid algorithm

Professor: Yossi Azar *Scribe:Guy Ilany*

1 Introduction

In this lecture we will see a polynomial running time algorithm for solving a general system of linear inequalities. The Ellipsoid algorithm was designed by KHACHIAN (1979). Here, we present a simplified version by Yamnitsky and Levin. The algorithm is applicable to problems where the input is rational. We would define and utilize the Simplex. We will see the general outline followed by more detailed description and analysis.

2 Reminder of previous lecture

In the previous lecture we have seen:

2.1 The Dual theorem

Given a Linear Programming problem $MIN(C^t x)$ with constraints:

- $Ax = b$
- $x \geq 0$

(the Primal problem), we can construct its Dual problem: $MAX(y^t b)$ with constraints:

- $y^t A \leq c_j$

such that:

- $C^t x \leq y^t b$ (The weak side of the theorem).
- $C^t x = y^t b$ (The strong side of the theorem).

We have then deduced both:

- The complementary slackness conditions.
- FARKAS' lemma - a geometric interpertation of the Dual problem (which, historically, preceded the Dual theorem).

2.2 The MinMax theorem (zero-sum games)

Given two player's strategies:

Player 1:
goal is $MAX(\alpha)$, with constraints:

- $A^t x \geq \alpha \cdot \overline{e}_m$
- $x \geq 0$
- $\overline{e}_n^t \cdot x = 1$ (this is due to x being 1st player probability distribution)

Player 2:
goal is $MIN(\beta)$, with constraints:

- $(y^t A) \leq \beta \cdot \overline{e}_n^t$
- $y \geq 0$
- $\overline{e}_m^t \cdot y = 1$ (this is due to y being 2nd player probability distribution)

We add α and β respectivly to the sets of variables x, y and reformulate as an (m+1) by (n+1) problem:

Player 1:
goal is $MIN(-\alpha)$, with constraints:

- $A^t x - \alpha \cdot \overline{e}_m \geq 0$
- $x \geq 0$
- $-\overline{e}_n^t \cdot x = -1$
- Find x, α

Player 2:
goal is $MAX(-\beta)$, with constraints:

- $(y^t A) - \beta \cdot \overline{e}_n^t \leq 0$
- $y \geq 0$
- $-\overline{e}_m^t \cdot y = -1$
- Find y, β

In case both are feasible, according to the Dual theorem, we conclude $\alpha = \beta$. Indeed $(1, 0, 0, 1, 0, 0)$ and large (in absolute value) negative α make the Primal feasible; $(1, 0, 0, 1, 0, 0)$ and large positive β make the Dual feasible.

2.3 Feasible $\Leftrightarrow$ Optimum

By a polynomial time reduction:
Suppose we have a black box that finds a feasible solution for a system of linear equalities. For a given Primal problem set for which we want to find an optimum solution, we look at the Dual problem, and add it's set of constraints. Now, the problem has $(n + m + 1)$ constraints:

- $Ax \geq b$
- $y^t A \leq C$
- $C^t x = y^t b$

The goal is: $MIN(C^t x)$. If there exist a feasible solution to the new problem, it implies, by the Dual theorem, the optimum of both the Primal and the Dual.

3 KHACHIAN algorithm

We will see a polynomial time algorithm for solving a system of linear inequalities of the form $MIN(C^t x)$ with constraints:

- $Ax \leq b$
- $x \geq 0$

where A, b are real-valued. Yamnitsky's and Levin's algorithm is a simplified version of the Ellipsoid algorithm of KHACHIAN, in the sense that algebraic manipulations are simpler due to usage of the Simplex construct. Note that although theoretically, the Simplex algorithm could have an exponential running time in worst case, in practice it may be much faster than the Ellipsoid algorithm.

3.1 Overview

The algorithm finds a feasible solution which, by the reduction we saw on the last lecture, implies the existence of an optimum solution. The concept is to find a Simplex S that encloses all the polytope vertices, then check if the mid point of S is a feasible solution. If not - find another S with smaller volume, and that still encloses the polytope. This scheme should eventually lead to a feasible solution (a small enough Simplex which mid point lays inside the polytope), or reach the step where the volume is so small that a feasible polytope could not be enclosed within it. This in turn would lead to the conclusion that there is no solution to the problem.

3.2 The input

Let $B \in \mathbb{N}$ be the maximum absolute value in the input matrix A. Furthermore, W.L.O.G. we assume that $B \in \mathbb{N}$, otherwise we multiply by the minimum common divisor of the input. This in turn could change the polynomial degree by some factor, but still the running time remains polynomial in the size of the input. Obviously, the size of the input is $O(n \cdot m \cdot \log B)$ bits but not less than $\Omega(n + m + \log B)$. Specifically define by L the size of the input in bits, then $2^L \geq n^{n+1} B^{n+1} m$. Taking logarithm, we have $L \approx (n+1)\log n + (n+1)\log B + \log m$.

3.3 Feasible solution

The goal of the algorithm is to find some point (if exists) of the polytope $P = \{x | Ax \leq b\}$.

Theorem: All vertices $x \in P$ could be written as: $x = (\frac{d1}{d}, \frac{d2}{d}, \frac{d3}{d}, \ldots \frac{dn}{d})$, where both $0 \leq |d_i| \leq 2^L$ and $0 \leq |d| \leq 2^L$. This means that although the input constitute of

whole numbers, the solution could be rational (but certainly not irrational). Nevertheless, the denominator is not too big.
Proof: By Cramer's rule for solving $M_{nxn}x = b$, each entry of the solution vector is of the form: $x_i = \frac{det(M_i)}{det(M)}$ where $|det(M)| \leq n!\cdot MAX(M)^n$ and $|det(M_i)| \leq n!\cdot MAX(M)^n$ for all i. This produces $0 \leq |d| \leq n!\cdot MAX(A)^n = n!\cdot B^n \leq 2^L$ as well as $0 \leq |d_i| \leq n!\cdot MAX(A)^n = n!\cdot B^n \leq 2^L$.

3.4 Linear Strict Inequalities (LSI)

$Definition$: LSI is a system of linear strict inequalities of the form $Ax < b$.
Theorem: Let A be a matrix $s.t.$ $a^i_j \in \mathbb{Z}$ for all i, j. Then there exists a solution to the system $Ax \leq b$ if and only if there exists a solution to the LSI $Ax < b + \varepsilon$, where $\varepsilon = 2^{-2L}$.
Proof - sketch: While one direction is trivial, the other requires more work, but the concept is that since ε so small relative to A's entries, it doesn't cause significant pertubations in x. Morover, there exists a polynomial time algorithm that given a solution to one case, finds a solution to the other. Thus, given a non-strict problem $Ax \leq b$ with $a_{i,j} \in \mathbb{Z}$, we could solve the LSI $Ax < b+2^{-2L}$ and by the theorem above, find a solution to the non-strict problem. Of course we need to manipulate the LSI (to make it in integer form) to be: $2^{2L}\cdot Ax < b\cdot 2^{2L} + 1$. Notice that the amount of input bits has increased. Notice that the increase in input is only polynomial, which keeps the running time polynomial. According to the prior analysis, $2^{L'} = n^{n+1}(2^{2L}B)^{n+1}m$, and after taking logarithm, $L' = O((n+1)\log n + (n+1)\cdot 2L + (n+1)B+)$ and indeed $L' \approx L^2$.

3.5 Definitions

1. A set of $n+1$ points, $s.t.$ $v_i \in \mathbb{R}, i = 0, 1, 2..., n$ are said to be in a *general position* if $Rank(v_1 - v_0, v_2 - v_0 ..., v_n - v_0) = n$. In other words, there is no hyperplane that passes through all the $n+1$ points.
2. *n-Simlex* is a *convex hull* of $n+1$ points in a general position (i.e. in 2 dimensions it is a triangle, in 3 dimensions it is a tetrahedron and so on).
3. The center of an n-Simplex is the algebric mean of it's n vertices: $Center(S) = \frac{1}{n+1}\cdot \sum_{i=0}^{n} v_i$.

4. The volume of a Simplex: $Vol(S) = \frac{1}{n!}\cdot Det \begin{pmatrix} v_1 - v_0 \\ v_2 - v_0 \\ \cdots \\ v_n - v_0 \end{pmatrix}$
5. Define a *Half-Simplex* as the intersection of a Simplex with a half-space, where *Center(S)* lies on the supporting hyperplane of the half space.

3.6 Prequisites for the algorithm

Theorem 1: Let $P = \{x \in \mathbb{R}^n | Ax < b\} \neq \emptyset$ be a polytop (Notice the LSI form), then there exist a simplex $S \subset P$ s.t. $Vol(S) > 0$.
Remark: A polytope could have 2^n vertices, while a simplex has only $n+1$ vertices. This is what simplifies the algorithm.
Proof: By the help of the next basic theorem (which we would prove on the next lecture):
Theorem D (basic theorem): Each vertex $x \in P$ of a polytop could be described as a convex combination of at most $n+1$ vertices of P.
According to theorem D, P is covered by the union of all convex hull with at most $n+1$ vertices. Thus $Vol(P) > 0$. Moreover, because P could be viewed as a union of Simplexes, there must exists at least one such simplex S with $Vol(S) > 0$ (otherwise all the volumes are 0, and the total volume is 0 which means that the volume of P is 0 in contrast to the assumption). This simplex has $n+1$ vertices, otherwise it's volume would have been 0.
Remark: The key to having a non-empty set of feasible solutions is the LSI form which guarantees that the feasible space is of full dimension n. *Remark:* The volume of this simplex would be exponentially small, but not too small: $Vol(S) \geq n^{-n \cdot O(1)} \cdot B^{-n \cdot O(1)}$. This is due to the same bound over determinants.

Theorem 2: Each polytope could be enclosed by a *not too large* a Simplex. More formally, given a polytope $P = \{x \in \mathbb{R}^n | Ax < b\}$, and assume W.L.O.G that $x \geq 0$ (if this is not the case, we could shift the polytope by the maximum negative value), then there exists a simplex S such that $P \subseteq S$ and $Vol(S) \leq n^{n \cdot O(1)} \cdot B^{n \cdot O(1)}$.
Proof: Taking n vertices $\{M \cdot e_i | i = 1...n\}$ along with the zero vector, define a Simplex. Taking M to be large engough such that $M/n \geq 2^L$ guarentees that all the polytope's vertices are enclosed by a 'cube' $0 \leq x_i \leq M/n$ for all i, which implies that all the polytope is enclosed by the cube and the enclosing simplex S. The volume of such a simplex could increase by a factor the power of n, but is is still gets 'swollwed' by the $O(1)$.

4 The algorithm

4.1 Outer loop

Precondition: Assume that there is a feasible set P for the LSI form $Ax < b + \varepsilon$ that could be enclosed by a simplex S as shown in the last section. If no feasible solution exists, the algorithm would stop with no solution after a polynomial amount of iterations.
1. $S \leftarrow Enclosing - simplex(P)$.

2. While $Vol(S)$ is no smaller than $n^{-n^{o(1)}} \cdot B^{-n}$ do:
   2.1. if $Center(S)$ is feasible, then **return with** $Center(S)$.
   2.2. else, find a contradicting constraints, and a half-simplex $S_{1/2}$.
   2.3. Find a smaller simplex $\overline{S}$ such that $S_{1/2} \subseteq \overline{S}$ and set $S \leftarrow \overline{S}$.
3. **Return unfeasible**.

4.2 Analysis

We would see how to find the half simplex $S_{1/2}$ of step 2.2, how to find the simplex $\overline{S}$ of step 2.3 which is smaller than S, as well as bounding the iterations of the main loop of step 2.
First, we find (arbitrarily) one contradicting constraint, i.e. one that violates $ax < a \cdot Center(S)$. We define $S_{1/2}$ by the intersection of S and a hyperplane that passes thtough $Center(S)$ and that is parallel to the contradicting constraint.
Key theorem: Let $S_{1/2}$ be as described above in step *2.2*, then there exists a simplex $\overline{S}$ s.t. $S_{1/2} \subseteq \overline{S}$ and $\frac{Vol(\overline{S})}{Vol(S)} < e^{-\frac{1}{4n^2}} \approx 1 - \frac{1}{4n^2}$. Let us see that this imposes a polynomial bound on the number of iterations: The initial volume is $n^{n^{o(1)}} B^{n^{o(1)}}$. Each step decreases the volume by $1 - \frac{1}{4n^2}$. Let t be the iterations count. The algorithm terminates when $Vol(S) < n^{-n^{o(1)}} B^{-n^{o(1)}}$. Solving for t, we get $t \approx 4n^2 \cdot n^{o(1)} \cdot \log B$ which is polynomial in n.
Proof of the key theorem: Let the vertices of S be $\{v_0, v_1 ..., v_n\}$ and let $\overline{ax} < \overline{b}$ be the hyperplane that passes through $Center(S)$. Define a linear function $e(x) = \overline{b} - \overline{ax}$. Let k be the index that maximize $e(v_i)$ (v_k is the most distant vertex from $Center(S)$ and it would be the 'anchor' for computing $\overline{S}$). In addition, since $Vol(S_{1/2}) > 0$ we know that $e(v_k) > 0$. Now define $\{u_0, u_1, ..., u_n\} u_i = v_k + \frac{v_i - v_k}{d_i}$, $d_i = 1 - \frac{e(v_i)}{n^2 \cdot e(v_k)}$. What we basically do here is stretching (contracting) each vertex of S that is on the positive (negative) side of the hyperplane $\overline{b}$ with proportion to its relative distance from the anchor and the distance of the anchor from $Center(S)$, and in the direction $v_i - v_k$. The set of $n+1$ points define the new simplex $\overline{S}$.
Looking at the new simplex $\overline{S}$, we see that the new vertices u_i lie on the same side of the hyperplane $\overline{b}$ as the old vertices v_i (this is due to $d_i \geq 0$ since $e(v_i) \leq e(v_k)$).
It is left to see two things:

- 1. $S_{1/1} \subseteq \overline{S}$

- 2. $\frac{Vol(\overline{S})}{Vol(S)} < 1 - \frac{1}{4n^2}$

1. Suppose $x \in S_{1/2}$, then $x = \Sigma t_i v_i$, where $\Sigma t_i = 1, t_i \geq 0$. Also, $e(x) \geq 0$. Now define $\overline{t_i} = \begin{cases} d_i t_i, i \neq k \\ d_k t_k + \frac{e(x)}{n^2 \cdot e(v_k)}, i = k \end{cases}$

First, $\bar{t}_i \geq 0$ simply because all terms are positive.

Second, $\sum_{i=0}^{n} t_i = 1$. This is because:

$$\sum_{i=0}^{n} \bar{t}_i = \tag{1}$$

$$\sum_{i=0}^{n} t_i \cdot d_i + \frac{e(x)}{n^2 e(v_k)} = \tag{2}$$

$$\sum_{i=0}^{n} t_i \left(1 - \frac{e(v_i)}{n^2 e(v_k)}\right) + \frac{e(x)}{n^2 \cdot e(v_k)} = \tag{3}$$

$$\sum_{i=0}^{n} t_i - \frac{1}{n^2 \cdot e(v_k)} \cdot \sum_{i=0}^{n} e(v_i) t_i + \frac{e(x)}{n^2 \cdot e(v_k)} = \tag{4}$$

$$1 - \frac{1}{n^2 \cdot e(v_k)} \cdot \sum_{i=0}^{n} e(t_i v_i) + \frac{e(x)}{n^2 \cdot e(v_k)} = 1 \tag{5}$$

Where, on the last step, the second and the third terms cancel each other by means of linearity of e.

Third, we want to see that $\sum_{i=0}^{n} \bar{t}_i u_i = x$ (x is a convex sum of $\overline{S}$):

$$\sum_{i=0}^{n} \bar{t}_i u_i = \tag{6}$$

$$\sum_{i=0}^{n} t_i d_i \left(v_k + \frac{v_i - v_k}{d_i}\right) + \frac{e(x)}{n^2 \cdot e(v_k)} \cdot v_k = \tag{7}$$

$$\sum_{i=0}^{n} t_i v_i + v_k \left(\sum_{i=0}^{n} t_i d_i\right) - v_k \cdot \sum_{i=0}^{n} t_i + v_k \frac{e(x)}{n^2 \cdot e(v_k)} = x \tag{8}$$

Where, on the last step, the sum of 2nd and 4th terms are canceled by the 3rd term.

2. Regarding the volume:

$$n! \cdot Vol(\overline{S}) = Det \begin{pmatrix} \frac{v_0 - v_k}{d_0} \\ \cdots \\ \frac{v_i - v_k}{d_i} \\ \cdots \\ \frac{v_n - v_k}{d_n} \end{pmatrix}_{i \neq k} = \frac{1}{\prod_{i \neq k} d_i} \cdot Det \begin{pmatrix} v_0 - v_k \\ \cdots \\ v_i - v_k \\ \cdots \\ v_n - v_k \end{pmatrix}_{i \neq k} = \frac{1}{\prod_{i \neq k} d_i} \cdot Vol(S) \cdot n!$$

Now, let us check what is the minimum of $\prod_{i \neq k} d_i$ because this imposes the lower bound on the volume ratio $fracVol(S)Vol(\overline{S})$. We know that $d_i \geq 1 - \frac{1}{n^2}$. Let us start by looking at the sum

$$\sum_{i \neq k} d_i = \sum_{i \neq k} \left(1 - \frac{e(v_i)}{n^2 e(v_k)}\right) = n - \sum_{i=0}^{n} \frac{e(v_i)}{n^2 e(v_k)} + \frac{e(v_k)}{n^2 e(v_k)}$$

by adding and subtracting the kth term.
Looking at the middle term, we can see that it is a product of some constant by $e(Center(S))$ which is 0: $C = Center(S) = \sum_{i=0}^{n} \frac{1}{n+1} \cdot v_i$, so $0 = e(C) = \frac{1}{n+1} \cdot \sum_{i=0}^{n} e(v_i)$.
Thus, we have $\sum_{i \neq k} d_i = n + \frac{1}{n^2}$.
Back to the product $\prod_{i \neq k} d_i$ - the minimum value is reached when all terms but one are at the minimal value: $1 - \frac{1}{n^2}$. This is due to the sum, and the last term must be $1 + \frac{1}{n}$. So $\prod_{i \neq k} d_i \geq \left(1 - \frac{1}{n^2}\right)^{n-1} \left(1 - \frac{1}{n}\right) =$ by multiplying and dividing by the conjugate $= \frac{\left(1 - \frac{1}{n^2}\right)^n}{1 - \frac{1}{n}} \geq e^{-\frac{1}{4n^2}}$ where the last transition is via Taylor expansion.

5 Summary

We have seen a simplified version of KHACHIAN's Ellipsoid algorithm by Yamnitsky and Levin which solves a general system of inequalities in a polynomial bounded running time. The algorithm utilizes the Simplex construct by a scheme of enclosing the feasible space and decreasing, on each iteration step, the enclosing Simplex S, until the center of the S is a feasible solution, or until S is so small, that the algorithm concludes that there is no feasible solution.