1 The Ellipsoid Method

Khachian showed in 79 that linear programming can be solved in polynomial time. We give a brief account of his method and refer the reader to the discussions in [Chv93, Section Appendix] and [GT89] for details.

We consider LPs in inequality form:

$$\max\left\{c^T x \; ; \; Ax \le b\right\} \tag{LP}$$

The entries of A, c, and b are assumed to be integral. Define

$$L = \sum_{0 \le i \le m, 0 \le j \le n} 1 + \log(1 + |a_{ij}|))$$

where $a_{i0} = b_i$ and $a_{0j} = c_j$ and $a_{00} = 0$.

Theorem 1 *The Ellipsoid method solves linear programs in time polynomial in L*

The proof is through a sequence of lemmas.

Lemma 1 *If one can decide feasibility of LPs in polynomial time then one can compute optimal solutions in polynomial time.*

So we need to decide feasibility of $Ax \le b$. The solution to $Ax \le b$ might be a single point. In the Ellipsoid method one argues about the volume of the set of solutions and hence we want the situation that either there is no solution or that the solutions set has a certain minimum volume. This is easy to achieve by perturbation.

Lemma 2 (Minimum Volume) *Let $\epsilon = 1/(mn2^L)$. $Ax \le b$ is feasible if $Ax \le b + \epsilon e$ is feasible (e is the vector of ones). Moreover, if the latter problem is feasible, the set of solutions inside the ball with radius 4^{nL} centered at the origin, has volume at least $2^{-(n+1)L}$.*

In the sequel I use S to denote the set of solutions of the perturbed system inside the ball with radius 4^{nL} centered at the origin. Then S is either empty or has volume at least $2^{-(n+1)L}$.

I view the Ellipsoid method as a generalization of binary search. In binary search we maintain an interval that contains the solution. In each iteration, we test whether the midpoint of the interval is a solution. If not, we proceed with one of two half-intervals. The Ellipsoid method generalizes this strategy to arbitrary dimensions.

An ellipsoid in $\mathbb{R}^n$ consists of all points $x \in \mathbb{R}^n$ satisfying

$$(A(x-c))^T(A(x-c)) \le 1$$

where $c \in \mathbb{R}^n$ is the center of the ellipsoid and A is any $n \times n$ non-singular matrix. For $A = I$, we have the unit ball centered at c, for $A = diag(a_1, \ldots, a_n)$, we have an ellipsoid with center c and axes of length $1/a_i$. An ellipsoid can also be viewed as the image of the unit ball under the affine transformation $x \mapsto A(x-c)$. A one-dimensional ellipsoid is an interval.

The ellipsoid method maintains an ellipsoid that is known to contain S.

1. Initialize an ellipsoid E to the ball with radius $4nL$ centered at the origin.

2. If the center c of E is feasible, stop and declare the problem feasible.

3. If the volume of E is less than $2^{-(n+1)L}$, stop and declare the problem infeasible.

4. Select an inequality $a_i^T x \le b_i$ among the inequalities in $Ax \le b$ which is violated by c, i.e., $a_i^T c > b_i$. For every point $x \in S$, we have $a_i^T x \le b_i \le a_i^T c$.

5. Consider the "half-ellipsoid" $(1/2)E = E \cap \left\{ x \ ; \ a_i^T x \le a_i^T c \right\}$ which is the intersection of E with a half-space whose boundary passes through c and replace E by the smallest (in volume) ellipsoid containing $(1/2)E$. Goto step 2.

Lemma 3 *In every iteration, the volume of E shrinks by at least the factor*

$$f = \exp(-\frac{1}{2(m+1)} < 1 \ .$$

The polynomiality of the method follows easily. We start with an ellipsoid whose volume is bounded by $(8nL)^n$. Observe that the ellipsoid is contained in a box with side length $8nd$. In each iteration the volume is shrunk be a factor f and hence the volume of the ellipsoid after k iterations is at most $f^k(8nL)^n$. We stop, if the volume goes below $2^{-(n+1)L}$ and hence any iteration index must satisfy $f^k(8nL)^n \ge 2^{-(n+1)L}$ or $k \log f \ge -(n+1)L - n\log(8nd)$ or

$$k \le (-(n+1)L - n\log(8nd))/\log f \le 4(m+1)((n+1)L + n\log(8nd)) \ .$$

We have not given the formulae for how to update the ellipsoid in each iteration. The formulae involve additions, multiplications, divisions, and roots. It can be shown the the claims above stay true, if all calculations are carried out with 8^L bits of precision.

The ellipsoid method is not a practical method. However, its theoretical interest is immense.

- First, it shows that linear programming is a polynomial time problem.

- Second, it shows that it suffices to have a "separation oracle" for solving LPs. A separation oracle takes a point c and tells whether c is feasible. If c is infeasible it also provides a violated inequality. Observe that a separation oracle is all that is needed in step 4.

2 Interior Point Methods

Interior point methods were introduced by Karmakar in 1984. It is also polynomial time and it is a practical method. There are problems where it is superior to the simplex algorithm. Links to implementations can be found on Michael Trick's web page. We give a brief account and refer the readers to [GT89] for more details.

The simplex method walks along the boundary of the feasible region, the interior point algorithm walks in the interior. It is applied to the problem

$$\text{minimize } c^T x \quad \text{subject to} \quad Ax = 0, x \in \Delta$$

where $\Delta = \{x \; ; \; \sum_i x_i = 1, x \geq 0\}$ is the standard simplex in $I\!R^n$. It is further assumed that A is an $m \times n$ matrix of rank m, that $Ae = 0$ (e is the vector of ones) and hence $x^0 = e/n$ is feasible and that the optimal objective value is zero. Any LP can be transformed into this standard form.

Karamakar's algorithm generates a sequence x^k, $k \geq 0$ with

$$c^T x^k \leq \exp(\frac{-k}{5n}) c^T x^0 \ ,$$

i.e., the sequence of objective values converges exponentially to zero. After $k = 2nL$ iterations (same L as in preceding section) the objective value is so small that any nearby vertex of the feasible region is optimal.

How is x^{k+1} generated from x^k. We discuss how x^1 is found. The idea is simple. We replace the constraint $x \in \Delta$ by the stronger but smoother constraint $x \in B$ where

$$B = \left\{ x \; ; \; e^T x = 1 \text{ and } ||x - e/n|| \leq (n(n-1))^{-1/2} \right\} \ .$$

It can be shown that B is contained in Δ. In other words, we draw a ball with radius $r = (n(n-1))^{-1/2}$ around e/n. The optimization problem

$$\text{minimize } c^T x \quad \text{subject to} \quad Ax = 0, x \in B$$

is easy to solve. The solution is a point on the boundary of B (which is easy to compute given c and A). Let x^1 be the solution. Karmakar made two crucial observations.

1. The move from x^0 to x^1 decreases the objective value by a constant factor.
2. The procedure can be continued. The argument is extremely elegant. He argues that the transformation

   $$x \mapsto \bar{x} \quad \text{with} \quad \bar{x}_i = \frac{x_i / x_i^k}{\sum_i x_i / x_i^k}$$

   maps Δ onto itself and the point x^k to e/n. Hence the same reasoning as for the first step applies.

References

[Chv93] V. Chvatal. *Linear Programming.* Freeman, 93.

[GT89] D. Goldfarb and M.J. Todd. Linear programming. In G.L. Nemhauser et. al., editor, *Handbooks in Combinatorial Optimization and Management Science.* Elsevier, 1989.