The Ellipsoid algorithm

May 29, 2002

The Ellipsoid algorithm was developed by (formerly) Soviet mathematicians (Shor (1970), Yudin and Nemirovskii (1975)). Khachian (1979) proved that it provides a polynomial time algorithm for linear programming. The average behavior of the Ellipsoid algorithm is too slow, making it not competitive with the simplex algorithm. However, the theoretical implications of the algorithm are very important, in particular, providing the first proof that linear programming (and a host of other problems) are in P.

Consider a generalization of the "20 questions" game, from one dimension to many dimensions. The input to the game is an n dimensional ball B of radius $R > 1$. In it, an adversary hides a unit ball U. The purpose of the algorithm is to find a point in U. The game proceeds in rounds. In each round, the algorithm specifies one point $p \in B$. If p happens to be in U, the game ends and the algorithm wins. If not, the adversary provides a hyperplane that separates between p and U, and the game proceeds with the next round. There are algorithms that win the game within $O(n \log R)$ rounds, but their complexity per round is rather high. The Ellipsoid algorithm wins the game within $O(n^2 \log R)$ rounds, with a rounghly similar complexity per round.

We shall discuss (briefly) the following issues:

1. How linear programming is reduced to the above game.
2. Dependency on R, and the concept of strongly polynomial time algorithms.
3. Indepence of m, the number of constraints. The notion of a separation oracle.
4. Algebraic representations of ellipsoids. Positive definite matrices.
5. What needs to be proven in order to show that the ellipsoid algorithm runs in polynomial time.

Homework

Prove that for any point in a triangle, there is a line that passes through the point and separates the triangle into two regions, one of which has at least a 5/9-fraction of the area of the triangle. (Hint: consider what happens if the point is the center of gravity of the triangle.)