Ellipsoid algorithm, the Lovasz Theta function – Handout4

May 31, 2015

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 (such an adversary is referred to as a *separation oracle*), and the game proceeds to 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. It is a polynomial time algorithm, though not *strongly polynomial time*.

Let Q be an n by n nonsingular real matrix and $t \in R^n$. The mapping $T(x) = Qx + t$ is called an *affine transformation.* A *unit ball* $S(0,1)$ in R^n is the set $\{x|x^T x \leq 1\}$. An *ellipsoid* is the image of a unit ball under an affine transformation. Observe that $y = Qx+t$ implies $x = Q^{-1}(y-t)$. Hence using the notation $B = QQ^T$, an ellipsoid is

$$T(S(0,1)) = \{y|(Q^{-1}(y-t))^T Q^{-1}(y-t) \leq 1\} = \{y|((y-t)^T B^{-1}(y-t) \leq 1\}.$$

The matrix B is *positive definite* meaning that it is real and symmetric, and satisfies the following conditions (all of which are equivalent):

- $x^T Bx > 0$ for all nonzero $x \in R^n$.
- all its eigenvalues are real and positive.
- there exists a matrix Q with linearly independent rows such that $B = QQ^T$. (One possible choice for Q is to have column i equal to $\sqrt{\lambda_i} v_i$, where v_i is the ith eigenvector of B, and λ_i is its eigenvalue.)

The eigenvectors of B are the principle axes of the ellipsoid, the square roots of the eigenvalues are their lengths, and the square root of the determinant gives the volume (scaled by the volume of the unit ball).

In the ellipsoid algorithm we construct a sequence of ellipsoids $E_k = (B_k, t_k)$. If t_k violates the constraint $a_i^T x \leq b_i$ then we take E_{k+1} to be an ellipsoid that contains $\frac{1}{2}E_k = \{y \in E_k : a_i^T y \leq a_i^T t_k\}$, for which there are the following formulas:

$$t_{k+1} = t_k - \frac{1}{n+1}\frac{B_k a_i}{\sqrt{a_i^T B_k a_i}} \qquad B_{k+1} = \frac{n^2}{n^2-1}(B_k - \frac{2}{n+1}\frac{B_k a_i a_i^T B_k}{a_i^T B_k a_i})$$

It can be shown that $\mathrm{vol}(E_{k+1}) < e^{-1/2(n+1)}\mathrm{vol}(E_k)$. The proofs start with the simplest ellipsoid, the unit ball, and then use linear transformations.

Positive semidefinite matrices form a convex set, and the ellipsoid algorithm can be used to optimize over them (*positive semidefinite programming*) up to arbitrary precision. As examples, we shall consider the problem of embedding a finite metric space in Euclidean space with minimum distortion, and the Lovasz theta function.

Let $\omega(G)$, $\alpha(G)$, $\chi(G)$, $\bar{\chi}(G)$ denote the maximum clique size, maximum independent set size, chromatic number, and clique cover number of graph G, respectively. A graph G is *perfect* if $\omega(G') = \chi(G')$ for every vertex induced subgraph G' of G. Lovasz proved (the weak perfect graph conjecture) that the complement of a perfect graph is perfect. Hence for perfect graphs, $\alpha(G') = \bar{\chi}(G')$ for every vertex induced subgraph G' of G. Many classes of graphs (and their complements) are known to be perfect, including bipartite graphs, line graphs of bipartite graphs, interval graphs, comparability graphs, chordal graphs (the above classes are not disjoint). A chordless odd cycle of length at least five is not a perfect graph, and neither is its complement. Chudnovsky, Robertson, Seymour and Thomas proved (the strong perfect graph conjecture) that every nonperfect graph must contain one of the above as a vertex induced subgraph.

Let f be a function on graphs with the following "sandwich" property: $\alpha(G) \leq f(G) \leq \bar{\chi}(G)$. If f is computable in polynomial time, it can be used to compute $\omega(G)$, $\alpha(G)$, $\chi(G)$, $\bar{\chi}(G)$ for perfect graphs. In fact, it can be used iteratively in order to find maximum independent sets and cliques in perfect graphs. The ϑ function of Lovasz has the sandwich property, and can be computed with arbitrary precision in polynomial time (via a semidefinite program). In can also be used to find a minimum coloring and minimum clique cover of perfect graphs in polynomial time.

Homework: Hand in by June 21 (no class on June 14).

1. 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 and the line is parallel to an edge of the triangle.)

2. Show (without using the strong perfect graph theorem) that in every perfect graph G, if some vertex v is duplicated (a vertex v' is added, and for every vertex u with an edge (u, v) we also put in the edge (u, v')), then the resulting graph is also perfect.

3. Show that every interval graph is the complement of some comparability graph.

4. Give an example of a nonperfect graph G for which $\omega(G) = \chi(G)$ (and say how you know the values of $\omega(G)$ and $\chi(G)$ in this graph, and how you know that it is not perfect).

5. Show that the question of whether an input graph G has the property $\omega(G) = \chi(G)$ is NP-complete (namely, both in NP and NP-hard).