Lecture 12: Convex optimization continued

Outline of the ellipsoid algorithm, and some links to the details. We then talk about the cone of PSD matrices and a separation oracle for it.

Efficient algorithms for membership

We saw last class that minimizing a convex function f over a convex domain D can be done if we can simply find a point x in $D \cap L_\tau(f)$ for any given τ, where $L_\tau(f)$ is the level set. (We can then perform a binary search over τ.)

Let us thus see how we can solve the membership problem for a convex set D. The main result is that in many cases, having a separation oracle $\mathcal{A}$ for D suffices to obtain a polynomial time algorithm for the membership problem. The idea is due to Khachiyan (1979), who used it to obtain the first polynomial time algorithm for Linear Programming.

Ellipsoid Algorithm

Let us suppose we have a separation oracle $\mathcal{A}$ for a convex set $D \subseteq \mathbb{R}^n$. Further, suppose we know a *bounding ellipsoid* E_0 for D that is not *too much larger*. More precisely, we have $D \subseteq E_0$ and

$$\frac{\mathrm{vol}(E_0)}{\mathrm{vol}(D)} < 2^{\mathrm{poly}(n)}.$$

(This is not possible when the feasible set is infinite. It turns out that for the proof, we do not need an ellipsoid that fully contains D. We only need to have $\frac{\mathrm{vol}(E_0)}{\mathrm{vol}(D \cap E_0)}$ to be bounded as above.)

Then the following iterative algorithm efficiently finds a point $x \in D$:

1. Start with the ellipsoid E_0, and suppose its center is c_0. For $i = 0, \ldots, T$, where $T \geq 4n \log(\mathrm{vol}(E_0)/\mathrm{vol}(D))$, do the following:[1]
2. Run the separation oracle on point c_i.
3. If it says $c_i \in D$, then we found a feasible point, so return it.
4. Else, we get a hyperplane $\langle a, x \rangle \geq b$, with the property that the center lies on one side and the entire set D lies on the other. In this case we construct a new ellipsoid E_{i+1} (described below) that contains D, and has a volume $\leq (1 - 1/2n)\mathrm{vol}(E_i)$.

Theorem 1. *Assuming step (4) is efficient (which we will prove momentarily), the algorithm runs in polynomail time and returns a feasible point.*

[1] The algorithm does not know the ratio of volumes. However, we assumed that there is an upper bound on the log of the ratio of volumes that we do know, and is polynomial in n.

Proof. First, note that our upper bound on the ratio of volumes implies that T is polynomial in n. The efficiency of the separation oracle, and the step (4) implies that the overall running time is polynomial in n.

Now, why does the algorithm always find a feasible point? Suppose it ran for T steps without finding a feasible point. Then, the final ellipsoid we construct has the property that $\text{vol}(E_T) \le (1 - 1/2n)^T \text{vol}(E_0)$, which by our choice of T is $< \text{vol}(D)$. But this cannot happen if E_T contains D!

Thus one of the centers in the process must have been contained in D, in which case we would have returned it. □

The volume reduction step. It now remains to see the key step in the algorithm – step 4 above. I.e., we have an ellipsoid E_i that contains D, and we have a hyperplane that separates the center from the entire set D. We may assume that the plane passes through the center (because otherwise we can move the plane closer to the center, while maintaining the property that the plane separates the center and the set D). Also, we can shift the axes, so that the center of E_i is the origin.

Thus, the problem is now the following: we have an ellipsoid E_i centered at the origin, and a hyperplane $\langle a, x\rangle \ge 0$ such that the set D is entirely contained in $E_i \cap \{x : \langle a, x\rangle \ge 0\}$. We want to construct an ellipsoid E_{i+1} that (a) has volume $\le (1 - 1/2n)\text{vol}(E_i)$, and (b) contains D.

We will, in fact, ensure that E_{i+1} contains the entire set $E_i \cap \{x : \langle a, x\rangle \ge 0\}$. Let us first pause and review some basic properties of ellipsoids. In fact, the key novelty in Khachiyan's algorithm is the *use of ellipsoids* – they are easy enough to work with, and still have the property that we can reduce the volume in every iteration by a multiplicative factor. In fact, the problem itself does not naturally suggest the use of bounding ellipsoids. (That said, there were occurrences of bounding ellipsoids of convex bodies studied earlier – e.g. the celebrated John's theorem.)

Abstractly, an ellipsoid centered at the origin is defined by a positive semidefinite matrix M (a symmetric matrix all of whose eigenvalues are non-negative), and the ellipsoid is simply $\{x : x^T M x \le 1\}$. The sphere corresponds to the case of M being the identity. A more geometric definition is as follows: an ellipsoid (in n dimensions) has n orthogonal *axes*; we can always rotate the coordinates so that they align with these axes, and once we do so, the ellipsoid can be written as the set of all x that satisfy

$$\frac{x_1^2}{\ell_1^2} + \frac{x_2^2}{\ell_2^2} + \cdots + \frac{x_n^2}{\ell_n^2} \le 1.$$

For some non-negative reals ℓ_i (which are called the *principal axes* of the ellipsoid – these turn out to be the inverse of the eigenvalues of M above). The volume of the ellipsoid is $\ell_1 \ell_2 \ldots \ell_n$.

Now, let us provide the details of step 4 in the algorithm above. Suppose we rotate the space so that E_i has a form as above. Further, suppose we rescale the coordinates, so that the ellipsoid simply becomes the sphere. (This can always be done – simply replace x_i by $\ell_i x_i'$ for a new variable x_i'.)

The key points about these transformations are the following. First, the half-space defined by $\langle a, x\rangle \ge 0$ above, maps to another half-space through the origin. Now by symmetry, we can in fact assume that the half-space is $x_1 \ge 0$. Second, invertible linear transforms always preserve the ratio of volumes of objects. I.e., if we have two convex sets A, B, and we apply a linear transformation T to both of them, then we have

$$\frac{\text{vol}(TA)}{\text{vol}(TB)} = \frac{\text{vol}(A)}{\text{vol}(B)}.$$

This is useful, because our algorithm will produce a new ellipsoid that has a smaller volume, and then

Ellipsoid Algorithm continued on next page...

map it back to the initial space. There, we need the volume ratio to be maintained.

Thus, we have reduced the problem to a simple geometric one. We have a unit sphere in $\mathbb{R}^n$, and we would like to find an ellipsoid that fully contains the hemisphere $\{x : x_1 \geq 0 \text{ and } \|x\| \leq 1\}$. This requires a bit of calculation. For the details, please take a look at pages 2-4 of:

http://www-math.mit.edu/∼goemans/18433S09/ellipsoid.pdf (replace the ∼ if you are copy-pasting).

More on separation oracles and the PSD cone

Another interesting convex set that has a non-trivial separation oracle is the so-called *PSD cone*. This is a subset of $\mathbb{R}^{n^2}$, and we think of every point in it as an $n \times n$ matrix. Formally, it is the *cone* formed by all matrices of the form $\mathbf{u}\mathbf{u}^T$, where $\mathbf{u} \in \mathbb{R}^n$ (each matrix is treated as a point in $\mathbb{R}^{n^2}$).[2]

Thus the PSD cone contains all the matrices M of the form $\sum_i v_i v_i^T$, for some $v_i \in \mathbb{R}^n$. This is precisely the set of matrices that have all their eigenvalues being nonnegative.

We discuss more on this in the next class.

[2]A cone formed by a set of points $\mathbf{v}_i$ is the set of all non-negative linear combinations of $\mathbf{v}_i$, i.e., $\{\sum_i \alpha_i \mathbf{v}_i : \alpha_i \geq 0\}$.