Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
Ankur Moitra

Friday, March 18, 2016

Problem Set 6

Due: Wednesday, April 6, 2016 – 7 pm
Dropbox Outside Stata G5

Collaboration policy: collaboration is *strongly encouraged*. However, remember that

1. You must write up your own solutions, independently.

2. You must record the name of every collaborator.

3. You must actually participate in solving all the problems. This is difficult in very large groups, so you should keep your collaboration groups limited to 3 or 4 people in a given week.

4. Write each problem in a separate sheet and write down your name on top of every sheet.

5. **No bibles. This includes solutions posted to problems in previous years.**

1 Separation Oracles

Describe efficient separation oracles for each of the following families of convex sets. Here, "efficient" means linear time plus $O(1)$ calls to any additional oracles provided to you.

(a) The set $A \cap B$, given separation oracles for A and B.

(b) The ℓ_1 ball: $\|x\|_1 \leq 1$.

(c) Any convex set A, given a *projection oracle* for A. A projection oracle, given a point x, reports whether x is in A and if it is not additionally returns

$$\arg\min_{y \in A} \|x - y\|_2.$$

(d) The ϵ-neighborhood of a convex set A:

$$\{x \mid \exists y \in A, \|x - y\|_2 \leq \epsilon\}$$

given a projection oracle for A.

2 Optimization with the Ellipsoid Method

Here, we will show a way to optimize a linear function over a convex set using the ellipsoid algorithm. Recall that in class, we showed how to find a feasible point in a convex set given a separation oracle, and then used it to optimize a linear function by doing binary search over the optimum value, and turning the objective function into a linear constraint. Here we will explore an alternative method that uses the ellipsoid method directly without performing binary search. The basic idea is to optimize over the set $A \cap \{x \mid c^T x \leq v\}$ without actually knowing the function value v.

To avoid going through the analysis of the ellipsoid method again, we will give a more black-box description of the guarantees of the ellipsoid method. In particular, we will view it as an interactive process, producing output and reading input in each round. The process is initialized with a radius R. At the ith round, it outputs a point x_i. It then reads as input a direction d_i, and then goes on to round $i+1$.

Let $\widetilde{A}$ be *any* convex set contained in $B(0,R)$ such that $\widetilde{A}$ contains some ball of radius r. Furthermore, suppose that for all i with $x_i \notin \widetilde{A}$, d_i is a hyperplane that separates x_i from $\widetilde{A}$. Then we proved for $m = O(n^2 \log(R/r))$, some x_i, $i \leq m$ must satisfy $x_i \in \widetilde{A}$. Notably, this holds despite the ellipsoid method having no explicit knowledge of $\widetilde{A}$ except for the conditions on the vectors d_i. The idea is in some sense to change $\widetilde{A}$ as we go along.

Now consider the following algorithm for optimization. We will assume a set A with a separation oracle, a bounding radius R such that $A \subseteq B(0,R)$, and a linear objective to be minimized, $c^T x$.

Algorithm 1 Ellipsoid-Optimize

```
Initialize the ellipsoid method with radius R. Set v_best to ∞.
for i from 1 to m do
    Get x_i from the ellipsoid method.
    Use the separation oracle on x_i, finding if x_i ∈ A and a separating hyperplane if not.
    if x_i ∈ A then
        if c^T x_i < v_best then
            Set v_best to c^T x_i.
            Set x_best to x_i.
        end if
        Set d_i = c.
    else
        Set d_i to the separating direction returned by the oracle.
    end if
    Send the direction d_i back to the ellipsoid method.
end for
return x_best.
```

In other words, in each step we call the separation oracle for A, using the hyperplane it returns if $x_i \notin A$ and a hyperplane defined by the objective c otherwise (i.e. $x_i \in A$). After m iterations we simply return the best objective value out of all the points we have seen in A.

Prove that for any objective value v, as long as $A \cap \{x \mid c^T x \leq v\}$ contains a ball of radius r, Ellipsoid-Optimize will return an $x \in A$ satisfying $c^T x \leq v$. Note that this guarantee holds without any knowledge of the value v.

3 Two Player Zero Sum Games

This problem proves the von Neumann theorem on zero-sum games mentioned in class.

In a 0-*sum 2-player game*, Alice has a choice of n so-called *pure* strategies and Bob has a choice of m *pure* strategies. If Alice picks strategy i and Bob picks strategy j, then the *payoff* is a_{ij}, meaning a_{ij} dollars are transfered from Alice to Bob. So Bob makes money if a_{ij} is positive, but Alice makes money if a_{ij} is negative. Thus, Alice wants to pick a strategy that minimizes the payoff while Bob wants a strategy that maximizes the payoff. The matrix $A = (a_{ij})$ is called the *payoff matrix*.

It is well known that to play these games well, you need to use a *mixed* strategy—a random choice from among pure strategies. A mixed strategy is just a particular probability distribution over pure strategies: you flip coins and then play the selected pure strategy. If Alice has mixed strategy x, meaning he plays strategy i with probability x_i, and Bob has mixed strategy y, then it is easy to prove that the expected payoff in the resulting game is xAy. Alice wants to minimize this expected payoff while Bob wants to maximize it. Our goal is to understand what strategies each player should play.

We'll start by making the pessimal assumption for Alice that whichever strategy she picks, Bob will play best possible strategy against her. In other words, given Alice's strategy x, Bob will pick a strategy y that achieves $\max_y xAy$. Thus, Alice wants to find a distribution x that minimizes $\max_y xAy$. Similarly, Bob wants a y to maximize $\min_x xAy$. So we are interested in solving the following 2 problems:

$$\min_{\sum x_i = 1} \max_{\sum y_j = 1} xAy$$

$$\max_{\sum y_j = 1} \min_{\sum x_i = 1} xAy$$

Unfortunately, these are nonlinear programs!

(a) Show that if Alice's mixed strategy is known, then Bob has a pure strategy serving as his best response.

(b) Show how to convert each program above into a linear program, and thus find an optimal strategy for both players in polynomial time.

(c) Give a plausible explanation for the meaning of your linear program (why does it give the optimum?)

(d) Use strong duality (applied to the LP you built in the previous part) to argue that the above two quantities are *equal*.

The second statement shows that the strategies x and y, besides being optimal, are in *Nash Equilibrium*: even if each player knows the other's strategy, there is no point in changing strategies. This was proven by Von Neumann and was actually one of the ideas that led to the discovery of strong duality.

4 Weak Duality for Nonlinear Programs

Here, we will look at a simple class of *second-order cone programs*, which are *nonlinear* convex programs. Your task will be to show that a form of weak duality still holds for these programs.

The below convex programs are defined for an n-dimensional vector c, an $m \times n$ matrix A, an $l \times n$ matrix B, and an m-dimensional vector b.

The primal problem is to find an n-dimensional vector x:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b, \\ & \|Bx\|_2 \leq 1. \end{aligned}$$

This is similar to a linear program except for the extra ℓ_2 ball constraint $\|Bx\|_2 \leq 1$.

The dual problem is to find an m-dimensional vector y and an l-dimensional vector z:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & b^T y - \|z\|_2 \\ \text{subject to} \quad & A^T y + B^T z = c, \\ & y \geq 0. \end{aligned}$$

Prove that for any primal and any dual solution, the objective value of the primal is at least the objective value of the dual solution (i.e. weak duality).

5 Two Definitions of Submodular

Let N be a set (the "universe") and F a function mapping subsets of N to real numbers.

There are two different standard definitions of what it means for F to be *submodular*:

(i) For all A, B,

$$F(A \cap B) + F(A \cup B) \leq F(A) + F(B).$$

(ii) For all $A \subseteq B$, $j \notin B$

$$F(A \cup \{j\}) - F(A) \geq F(B \cup \{j\}) - F(B).$$

Your task is to prove these definitions are equivalent:

(a) Show that (i) implies (ii).

(b) Show that (ii) implies (i).

6 Fun with Lovasz Extensions

Let $F(S)$ be a submodular function. Recall that the *Lovasz extension* of F, $f(x)$ is defined for nonnegative vectors x as

$$f(x) = F(\emptyset) + \int_0^\infty (F(\{i : x_i \geq a\}) - F(\emptyset))\, da.$$

Note that this definition applies to all $x \in R_+^n$, not just $x \in [0,1]^n$.

We will be looking at an interesting optimization problem involving the Lovasz extension:

$$\widehat{x} = \arg\min_{x \geq 0} f(x) + \frac{1}{2}x^2.$$

Recall that f is not necessarily monotonically increasing. $\widehat{x}$ will always be unique (due to the "strong convexity" of the objective).

The aim will be to relate $\widehat{x}$ to the minimizers of $F_a(S) = F(S) + a|S|$. Note that each F_a is itself a submodular function.

(a) Show that although $F_a(S)$ may have multiple minimizers, there is a unique set S_a that minimizes $F_a(S)$ and has a higher cardinality than any other minimizer.

(b) Show that for all $b > a$, $S_b \subseteq S_a$.

(c) For any positive vector x, define

$$X_a = \{i : x_i \geq a\}.$$

Show that

$$f(x) + \frac{1}{2}x^2 = F(\emptyset) + \int_0^\infty (F(X_a) - F(\emptyset) + a|X_a|)\, da.$$

(d) Show that for any $a > 0$, $\{i : \widehat{x}_i \geq a\} = S_a$.

This means that solving one convex optimization problem lets us minimize *all* the submodular functions F_a simultaneously!

You may assume without proof that the optimization problem defining $\widehat{x}$ has a unique optimum.