AM 221: Advanced Optimization Spring 2016

Prof. Yaron Singer *Problem Set 3 — Due Wed, Feb. 17th at 23:59*

Instructions: All your solutions should be prepared in LaTeX and the PDF and .tex should be submitted to canvas. For each question, the best and correct answers will be selected as sample solutions for the entire class to enjoy. If you prefer that we do not use your solutions, please indicate this clearly on the first page of your assignment.

The programming parts can be written in the programming language of your choice and the code should be submitted alongside your solutions.

1. Unboundedness. Let us consider the polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n \,|\, A\mathbf{x} \leq \mathbf{b}\}$ for some $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. In this problem, we are interested in the following linear program:

$$\max_{\mathbf{x} \in P} \mathbf{c}^\intercal \mathbf{x} \tag{LP}$$

We define the *recession cone* P^o associated with P by:

$$P^o \stackrel{\text{def}}{=} \{\mathbf{d} \in \mathbb{R}^n \,|\, \forall \mathbf{x} \in P, \forall \lambda \geq 0,\ \mathbf{x} + \lambda \mathbf{d} \in P\}$$

a. Show that $P^o = \{\mathbf{d} \in \mathbb{R}^n \,|\, A\mathbf{d} \leq 0\}$.

b. Show that P^o is a convex set.

c. Show that the linear program (LP) above is unbounded if and only if there exists $\mathbf{d} \in P^o$ such that $\mathbf{c}^\intercal \mathbf{d} > 0$.

2. Linearly Separable Datasets. In linear classification, we are given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. y_i is the *label* of data point i. Remember that we defined a dataset to be (strictly) linearly separable if and only if there exists $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that:

$$y_i(\mathbf{w}^\intercal x_i - b) > 0, \quad 1 \leq i \leq n$$

a. Show that the condition of being linearly separable is equivalent to the following condition: there exists $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

$$y_i(\mathbf{w}^\intercal x_i - b) \geq 1, \quad 1 \leq i \leq n$$

b. Let us define $X^+ \stackrel{\text{def}}{=} \{\mathbf{x}_i \,|\, y_i = +1,\ 1 \leq i \leq n\}$ and $X^- \stackrel{\text{def}}{=} \{\mathbf{x}_i \,|\, y_i = -1,\ 1 \leq i \leq n\}$. Using Farkas' lemma, show that if $\mathcal{D}$ is not linearly separable then $C(X^+) \cap C(X^-) \neq \emptyset$. Remember from the second problem set that $C(X)$ denotes the convex hull of X.

3. Max-Flow Min-Cut. In this problem we will study *flow networks*. A *flow network* is defined by a finite set of vertices V with two distinguished vertices, the *source* s and the *sink* t. Each edge $(u, v) \in V \times V$ has a *capacity* $c_{uv} \in \mathbb{R}^+$. Finally the source s has no incoming capacity, *i.e* $c_{us} = 0$ for $u \in V$ and the sink t has no outgoing capacity, *i.e* $c_{tu} = 0$ for $u \in V$.

A *flow* of the network is a family $(f_{uv})_{(u,v)\in V\times V}$ of real numbers satisfying the following constraints:

- *Positivity:* $f_{uv} \geq 0$ for all $(u, v) \in V \times V$.
- *Capacity constraint:* for all $(u, v) \in V \times V$, $f_{uv} \leq c_{uv}$.
- *Flow conservation:* for all $u \in V \setminus \{s, t\}$, $\sum_{v\in V} f_{vu} = \sum_{v\in V} f_{uv}$.

The amount of flow *leaving* s is $\sum_{u\in V} f_{su}$ and is called the *value of the flow* f. The goal of the *maximum flow problem* is to find a flow f satisfying the three conditions above whose value is maximal.

a. Show that the value of the flow f is equal to $\sum_{v\in V} f_{vt}$. Hence, the value of the flow could equivalently be defined as the amount of flow entering t.

b. Write the maximum flow problem as a linear program. Is this LP feasible? Is it bounded?

An $s-t$ *cut* of G is a partition of V into two sets: $V = S \cup T$ with $S \cap T = \emptyset$, such that $s \in S$ and $t \in T$. The *cost* c of an $s-t$ cut is defined by $c(S, T) := \sum_{(u,v)\in S\times T} c_{uv}$. The goal of the *minimum cut problem* is to find an $s-t$ cut of minimal cost.

d. Formulate a 0/1-integer program for the minimum cut problem (recall that 0/1-integer programs and their relaxations where covered in Section 1). Let us name this problem MinimumCut. Write the LP relaxation of MinimumCut. We will name this relaxation MinimumCut-LP.

e. Derive the dual of the maximum flow LP that you wrote in b. and show that it is equivalent to MinimumCut-LP. Show that the optimal value of MinimumCut is an upper bound on the value of the optimal flow.

f. Assuming that MinimumCut and MinimumCut-LP have the same optimal value [**bonus credits** if you prove this fact], prove the famous *max-flow min-cut theorem*:

The maximum value of a flow is equal to the minimal cost of an $s-t$ cut.

4. Boosting I: Weak Learners and Decision Stumps. In problems 4. and 5., we will use boosting to construct a strong classifier from weak learners. The dataset is available at `http://thibaut.horel.org/banknotes.data`: in each line, the first four columns contain measurements from a banknote (real numbers) and the last column is a binary (0 or 1) variable indicating if the banknote was forged. Denoting by $\mathbf{x}^i \in \mathbb{R}^4$ the measurements from banknote i, the goal is to construct a classifier which takes $\mathbf{x}^i$ as input and predicts the last column $y^i \in \{0, 1\}$.

A common choice for weak learners is to consider *decision stumps*: a decision stump is a simple classifier which classifies by simply comparing the values along one axis to a threshold. Formally, for each coordinate j and each threshold value t we define the following two decisions stumps:

$$s^-_{j,t}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \mathbf{x}_j < t \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad s^+_{j,t}(\mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \mathbf{x}_j \geq t \\ 0 & \text{otherwise} \end{cases}, \quad 1 \leq j \leq 4,\ t \in \mathbb{R},\ \mathbf{x} \in \mathbb{R}^4$$

Let us denote by n the number of samples in the dataset, the accuracy of a classifier s is defined by:

$$A(s) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[s(\mathbf{x}^i) = y^i]$$

This definition can be extended to any probability distribution on samples. Writing:

$$\Delta_n \stackrel{\text{def}}{=} \left\{ \mathbf{p} \in \mathbb{R}^n_+ \mid \sum_{i=1}^{n} \mathbf{p}_i = 1 \right\},$$

then for $\mathbf{q} \in \Delta_n$, we define:

$$A_{\mathbf{q}}(s) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \mathbf{q}_i \mathbf{1}[s(\mathbf{x}^i) = y^i]$$

Note that $A(s)$ is the special case where $\mathbf{q}$ is the uniform distribution.

a. Prove that for any coordinate j, $1 \leq j \leq 4$, any threshold value t, and any distribution $\mathbf{q} \in \Delta_n$, at least of one the following two conditions holds:

$$A_{\mathbf{q}}(s^+_{j,t}) \geq \frac{1}{2} \quad \text{or} \quad A_{\mathbf{q}}(s^-_{j,t}) \geq \frac{1}{2}$$

From now on, we will assume that the dataset is such that: $i_1 \neq i_2 \Rightarrow \mathbf{x}^{i_1}_j \neq \mathbf{x}^{i_2}_j$, $1 \leq j \leq 4$. That is, no two banknotes have similar measurements for any coordinate.

b. Prove that for any coordinate j, $1 \leq j \leq 4$ and any probability distribution $\mathbf{q} \in \Delta_n$, there exists a threshold t such that one of the following two conditions is true:

$$A_{\mathbf{q}}(s^+_{j,t}) > \frac{1}{2} \quad \text{or} \quad A_{\mathbf{q}}(s^-_{j,t}) > \frac{1}{2}$$

c. [for **bonus credits**] Define S_j to be the set of all decision stumps for coordinate j:

$$S_j \stackrel{\text{def}}{=} \left\{ s^-_{j,t} \mid t \in \mathbb{R} \right\} \bigcup \left\{ s^+_{j,t} \mid t \in \mathbb{R} \right\}$$

Does the result of part b. imply that S_j satisfies the weak learning property seen in class?

d. In practice, we will not use all decision stumps from S_j (there are infinitely many of them!), but only the best one. Describe a procedure which finds the element $s \in S_j$ for which $A(s)$ is maximum. Implement this procedure and report your code as well as the accuracy of the best decision stump for each coordinate j, $1 \leq j \leq 4$.

5. Boosting II: Aggregating Weak Learners. In this problem, we will use boosting to aggregate the decision stumps found in part 4.d. Remember from class that this is done by constructing a weighted average of the weak classifiers. Specifically we want to construct $\mathbf{p} \in \Delta_4$ solution to the following optimization problem:

$$\max_{\mathbf{p}\in\Delta_4} \min_{\mathbf{q}\in\Delta_n} \mathbf{p}^\intercal M \mathbf{q}$$

where $M \in \mathbb{R}^{4\times n}$ is such that M_{ji} is -1 when stump j is wrong on sample i and $M_{ji} = +1$ otherwise. We also saw in class that this problem can be rewritten as:

$$\begin{aligned} \max_{t,\,\mathbf{p}} \; & t \\ \text{s.t } & t \leq (\mathbf{p}^\intercal M)_i, \; 1 \leq i \leq n \\ & \mathbf{p} \in \Delta_n \\ & t \in \mathbb{R} \end{aligned}$$

which is an LP.

a. Write code to solve the above LP. We highly recommend using the CVXOPT library and more specifically the function cvxopt.solvers.lp whose documentation can be found at `http://cvxopt.org/userguide/coneprog.html#cvxopt.solvers.lp`. Note that the function expects the LP to be given in a specific form (all the inequalities must be in "less than" form). Report the code you wrote as well as the accuracy of the aggregate learner. How does it compare to the best individual weak learner?