AM 221: Advanced Optimization Spring 2016

Prof. Yaron Singer *Problem Set 8 — Due Wed, March. 30th 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. Gradient descent with a noisy oracle. In this problem, we will show that the gradient descent algorithm can be used when optimizing a strongly convex function given an approximate oracle for its gradient.

Let us consider a twice-differentiable strongly convex function $f : \mathbb{R}^n \to \mathbb{R}$, *i.e* we have:

$$mI_n \preceq \nabla^2 f(\mathbf{x}) \preceq MI_n,\ \mathbf{x} \in \mathbb{R}^n$$

for some constants $m, M > 0$. The function f is unknown to us. Instead, for any $\mathbf{x} \in \mathbb{R}^n$, we can query an oracle for the value of the gradient of f at $\mathbf{x} \in \mathbb{R}^n$. The oracle is erroneous in the following sense: let us denote by $\tilde{\nabla} f(\mathbf{x})$ the value returned by the oracle at point $\mathbf{x}$, then we have:

$$\nabla f(\mathbf{x})^\intercal \mathbf{y} - \delta \leq \tilde{\nabla} f(\mathbf{x})^\intercal \mathbf{y} \leq \nabla f(\mathbf{x})^\intercal \mathbf{y} + \delta,\ \ \mathbf{y} \in \mathbb{R}^n$$

for some $\delta > 0$. Such an oracle is called δ-erroneous.

We now consider the gradient descent algorithm from Lecture 9 where the update at each iteration is computed using the erroneous oracle: denoting by $\mathbf{x}^{(k)}$ the solution at iteration k, the solution at iteration $k+1$ is given by:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t\tilde{\nabla} f(\mathbf{x}^{(k)})$$

where the step size is constant set to $t = \frac{1}{M}$.

We say that a solution $\mathbf{x}^{'}$ has accuracy ε for f if $f(\mathbf{x}^{'}) - f(\mathbf{x}^*) \leq \varepsilon$, where $\mathbf{x}^*$ is the minimizer of f. By adapting the analysis of the gradient descent algorithm from Lecture 9, prove the following statement:

Theorem. *There exists a constant $C > 0$, such that for any $\varepsilon > 0$, the gradient descent algorithm using a δ-erroneous oracle with $\delta \leq \frac{C\varepsilon m}{M}$ computes a solution of accuracy 2ε in as many iterations as the standard gradient descent algorithm (*i.e *the one which uses the exact gradient).*

2. Duality and SVMs. In this problem, we will refine our analysis of the primal and dual formulations of the support vector machine optimization problem of Lecture 13. Remember that in this problem we have a dataset consisting of two clusters of data points in $\mathbb{R}^d$: positively labeled data points $\{\mathbf{x}_i,\ i \in I\}$ and negatively labeled data points $\{\mathbf{x}_j,\ j \in J\}$. The goal is to find an affine hyperplane $(\mathbf{a}, b) \in \mathbb{R}^d \times \mathbb{R}$ such that:

$$\mathbf{a}^\intercal \mathbf{x}_i + b > 0,\ i \in I$$
$$\mathbf{a}^\intercal \mathbf{x}_j + b < 0,\ j \in J$$

The primal problem is written as:

$$\begin{aligned}\min_{\mathbf{a}\in\mathbb{R}^d,\, b\in\mathbb{R}} \quad & \frac{\|\mathbf{a}\|^2}{4} \\ \text{s.t. } & \mathbf{a}^\intercal \mathbf{x}_i + b \geq 1,\ i \in I \\ & \mathbf{a}^\intercal \mathbf{x}_j + b \leq -1,\ j \in J\end{aligned}$$

and we computed the dual in class:

$$\begin{aligned}\max_{\lambda\in\mathbb{R}^{|I|},\, \mu\in\mathbb{R}^{|J|}} \quad & \frac{1}{\left\| \sum_{i\in I} \lambda_i \mathbf{x}_i - \sum_{j\in J} \mu_j \mathbf{x}_j \right\|^2} \\ \text{s.t. } & \sum_{i\in I} \lambda_i = \sum_{j\in J} \mu_j = 1 \\ & \lambda \geq 0,\ \mu \geq 0\end{aligned}$$

We assume that Slater's condition hold so that we have strong duality.

a. Let us denote by (λ, μ) a dual-optimal solution and by $(\mathbf{a}, b)$ a primal-optimal solution. Show that:
$$\text{either } \lambda_i = 0 \quad \text{or} \quad \mathbf{a}^\intercal \mathbf{x}_i + b = 1,\ \ i \in I$$
similarly, show that:
$$\text{either } \mu_j = 0 \quad \text{or} \quad \mathbf{a}^\intercal \mathbf{x}_j + b = -1,\ \ j \in J$$
Give a geometric interpretation.

b. Using part a., explain how a primal-optimal solution could be computed from a dual-optimal solution.

3. Revisiting the Maximum Coverage Problem Remember the Maximum Coverage Problem from Section 1. In this problem there is a universe of elements $\mathcal{U} = \{1, \dots, m\}$ and you are given as input a collection of n subsets $S_1, \dots, S_n$ of $\mathcal{U}$ ($S_i \subseteq \mathcal{U}$) and a budget $k \in \mathbb{N}$. The goal is select a collection $\mathcal{S}$ of at most k of the sets $S_1, \dots, S_n$ such as to maximize the number of elements of $\mathcal{U}$ contained in the union of the sets in $\mathcal{S}$. In other words, the goal is to solve:

$$\max_{|\mathcal{S}|\leq k} \left| \bigcup_{S_i\in\mathcal{S}} S_i \right|$$

One of the relaxations of this problem we considered in Section 1 was the following:

$$\begin{aligned}\max_{\mathbf{x}} \quad & \sum_{j=1}^{m} \min\left\{1, \sum_{i:j\in S_i} x_i\right\} \\ \text{s.t} \quad & x_i \geq 0 \quad 1 \leq i \leq n \\ & \sum_{i=1}^{n} x_i \leq k\end{aligned} \tag{P}$$

a. Show that the objective function in problem (P) is concave. How could you use an algorithm for minimizing a convex function to solve this relaxation?

b. Show that problem (P) can be reformulated as a linear program.

4. Barrier method. Let us briefly review the barrier method seen in section 8. Consider the following optimization problem with m inequality constraints:

$$\begin{aligned}\min_{\mathbf{x}\in\mathbb{R}^n} \ & f(\mathbf{x}) \\ \text{s.t. } & f_i(\mathbf{x}) \leq 0,\ 1 \leq i \leq m\end{aligned}$$

This problem is transformed into the following unconstrained problem using the barrier function $\hat{I}_t(u) = -\frac{1}{t}\log(-u)$:

$$\min_{\mathbf{x}\in\mathbb{R}^n} tf(\mathbf{x}) - \sum_{i=1}^{m} \log\big(-f_i(\mathbf{x})\big)$$

Let us denote by $\mathbf{x}^*(t)$ the optimal solution to this problem. Then the barrier method simply consists in solving the transformed problem for increasing values of t:

Algorithm 1 Barrier method with parameter $\mu > 1$

```
1: $t \leftarrow 1$, $\mathbf{x} \leftarrow$ feasible solution
2: while $\frac{m}{t} \geq \varepsilon$ do
3:     $\mathbf{x} \leftarrow \mathbf{x}^*(t)$    (the $\mathbf{x}$ from the previous iteration is used as the starting feasible solution)
4:     $t \leftarrow \mu t$
5: end while
6: return $\mathbf{x}$
```

In this problem we will use the barrier method to compute the support vector machine classifier for the forged banknotes dataset available at: http://thibaut.horel.org/banknotes.data. The setup is exactly identical to Problem 3 in Problem Set 7. The optimization problem is the following:

$$\begin{aligned}\min_{\mathbf{w}\in\mathbb{R}^d,\,\xi\in\mathbb{R}^n} \ & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda\sum_{i=1}^{n}\xi_i \\ \text{s.t } & \mathbf{w}^\intercal\mathbf{x}_i + \xi_i \geq 1,\ 1 \leq i \leq n \\ & \xi_i \geq 0,\ 1 \leq i \leq n\end{aligned} \tag{1}$$

where $\mathbf{x}_i,\ 1 \leq i \leq n$ are the transformed data points.

a. Write code to implement the barrier method described in Algorithm 1 for the optimization problem (1). For the internal optimization $\mathbf{x} \leftarrow \mathbf{x}^*(t)$, you are free to reuse either your implementation of the gradient descent algorithm or of Newton's method from previous problem sets. In fact you are encouraged to experiment with both!

b. Report the accuracy (fraction of correctly classified data points) for the optimal hyperplane found by the code you wrote in part a. For the penalty parameter λ, reuse the optimal value you found in Problem Set 7 (or experiment with different values if you haven't done Problem Set 7). What is the impact of the parameter μ on the convergence of your implementation?