Stony Brook University
CSE592 – Convex Optimization – Spring 18
Homework 2, Due: March, 6, 2018, 11:59PM

February 20, 2018

Instructions

- The homework is due on March 3, 2018. Anything that is received after the deadline will not be considered.
- The write-up **must** be prepared in Latex or Word and converted to pdf. No scanned hand-written notes!
- We can use any Latex class you like, just report question number and your answer.
- If the question requires you to implement a Python function, you must also submit a file with the implementation. Make sure it is sufficiently well documented that the TAs can understand what it is happening.

1 Duality

In this problem we will complete the derivation of the dual of a regression problem. In a linear regression model we would like to explain binary labels (responses) $y_1, \ldots, y_m$ using a linear function of input points (feature vectors, covariate vectors) $x_1, \ldots, x_m \in R^d$. In particular, we would like to find $w \in \mathbb{R}^d$ such that the sign of $w^\top x_i$ matches the label y_i. We quantify this by minimizing $g(y_i - w^\top x_i)$, where $g : \mathbb{R} \to \mathbb{R}$. Fitting a regression model therefore corresponds to optimizing the following unconstrained convex optimization problem:

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^{m} g(y_i - w^\top x_i) \tag{1}$$

In order to be able to derive a meaningful dual, e.g. in order to be able to obtain certificates of suboptimality, we instead rewrite (1) as:

$$\begin{aligned} \min_{w\in\mathbb{R}^d, z\in\mathbb{R}^m} \quad & \sum_{i=1}^{m} g(z_i) \\ \text{s.t.} \quad & z_i = y_i - w^\top x_i \qquad i = 1, \ldots, m. \end{aligned} \tag{2}$$

1. Use the general form for the dual of a problem with linear equalities and an arbitrary objective to write down the dual of (2) in terms of the conjugate of the f_0. If necessary, simplify the dual by eliminating unnecessary variables.

2. Write down the KKT conditions for a pair of primal and dual optimal solutions of (2). Explain how to use the KKT conditions to easily obtain a primal optimal solution if you are given a dual optimal solution.

Answer the following questions with the two choices of $g(\cdot)$.

$$g(z) = \frac{1}{2}z^2$$
$$g(z) = \max(|z| - 1, 0)$$

3. Derive the Fenchel conjugate of $g(z)$.

4. Write down the Fenchel conjugate of the objective $f_0(z, w)$ of (2) (Hint: express the objective as an independent sum of functions of each of the optimization variables).

5. Write down the dual.

Suggested review questions (please do not turn these in): 5.13, 5.41.

2 Programming Exercise

In this exercise, we will experiment with bisection search algorithm and get familiar with ways to implement the oracle access. The included archive contains partial Python code. In order to compile and run the code, you need to install Python3 and you will need NumPy package for the future exercises. One easy way to do set up both of them is to install Anaconda which is a free Python distribution that includes many Python packages for science, math, engineering, data analysis.

In this assignment, you will experiment with bisection, gradient descent, and Newtons method. The included archive contains partial python code, which you must complete. Areas that you will fill in are marked with "TODO" comments. **For this section, you should turn in *only* the file algorithms.py.**

2.1 Algorithms

All algorithms are implemented in `algorithms.py`.

Complete the implementation of the BFGS method. Complete the implementations of BFGS method in `algorithms.py`, algorithm 6.1 of Nocedal and Wright. Use the backtracking line search. Your implementation should terminate once $\|\nabla f(x)\|_2^2 \leq \epsilon$. Note that in order to guarantee always convergence, we should use a line-search procedure that guarantees the Wolfe condition (see discussion in the implementation section of the Nocedal and Wright). These kind of line-search procedures are very complex and behind the scope of this class. So, for simplicity, just skip the update of the inverse Hessian if $y_k^\top s_k$ is negative or too small (e.g. less than $1E-9$). **This is strategy might fail sometimes, but it is enough for our needs.**

2.2 Example Functions

The file `main.py` and `hw2_function.py` contains a number of functions on which you can try your algorithms. The Rosenbrock is a nasty non-convex function that is usually used to benchmark optimization algorithms. The optimum is in $(1, 1)$ and $p^* = 0$. The main algorithm will also plot the suboptimality with a logarithmic scale on the y axis. Note how bad is GD compared to Newton and BFGS even if all the three algorithms are initialized very close to the optimum.

The visualization part is commented out: feel free to use it to get a feeling of how the algorithms work.

2.3 Implementation Tips

For simplicity, I suggest to define matrix and vectors as NumPy matrices, so that you can use the "*" for matrix-vector multiplication, ".T" for the transpose, and "$.I$" for the matrix inverse.