Prof. Ming Gu, 861 Evans, tel: 2-3145
Email: mgu@math.berkeley.edu
http://www.math.berkeley.edu/∼mgu/MA170F2015

Math170: Mathematical Methods for Optimization
Term Project II

In this project we solve the $1-$*norm regression problem*:

$$\min_x \ \|Ax - b\|_1. \tag{1}$$

In this problem, the matrix $A \in \mathbf{R}^{n\times m}$ and vector $b \in \mathbf{R}^n$ are given, with $n > m$. Our job is to find the optimal solution vector $x \in \mathbf{R}^m$ that minimizes the 1-norm $\|Ax - b\|_1$. For any given vector $y = (y_1, \cdots, y_n)^T$, its 1-norm is defined as

$$\|y\|_1 = \sum_{j=1}^{n} |y_j|.$$

The problem (1) can be recast as a linear program as

$$\begin{array}{ll} \min\limits_{x,u} & e^T u \\ s.t. & -u \le Ax - b \le u, \end{array}$$

where $e = (1, \cdots, 1)^T \in \mathbf{R}^n$. This linear program, in turn, has a dual

$$\begin{array}{ll} \max\limits_{u} & -b^T y \\ s.t. & A^T y = 0, \\ & |y| \le e. \end{array}$$

While problem (1) is equivalent to a linear program, it is typically much more efficient to solve it with a specialized simplex-type method. Below we discuss such a method, under the **Non-degeneracy assumptions:**

- The matrix $A(B,:)$ is invertible for every index set $B \subset \{1, \cdots, n\}$ with exactly m indexes.
- There does not exist an index set B with more than m indexes such that $A(B,:)x = b(B)$.

Under these assumptions, there exists a unique index set $B_{\mathbf{opt}} \subset \{1, \cdots, n\}$ with m indexes such that $x_{\mathbf{opt}} = A\left(B_{\mathbf{opt}}, :\right)^{-1} b\left(B_{\mathbf{opt}}\right)$ solves the problem (1).

To describe an algorithm for solving problem (1), we start with any given index set $B \subset \{1, \cdots, n\}$ with m indexes. Let $B^c = \{1, \cdots, n\} \backslash B$ be the complement set of B. Choosing $x = A(B, :)^{-1} b(B)$, we reach objective value $\|A(B^c, :)x - b(B^c)\|_1$ in problem (1). Below we exlpain how to update B in a fashion similar to the simplex method to reach a lower objective value in problem (1). Just like simplex method, we then repeat this procedure until we eventually reach the optimal index set $B_{\mathbf{opt}}$ and therefore the optimal solution $x_{\mathbf{opt}}$.

Define

$$x = A(B, :)^{-1} b(B) \quad \text{and} \quad h = Ax - b.$$

It follows that $h(B^c) = A(B^c, :)x - b(B^c)$. By the non-degeneracy assumptions none of the components in $h(B^c)$ is exactly zero. Now define $y \in \mathbf{R}^n$ as

$$\begin{aligned} y(B^c) &= \mathbf{sgn}\left(h(B^c)\right), \\ y(B) &= -A(B, :)^{-T} A(B^c, :)^T y(B^c), \end{aligned}$$

where **sgn** is the sign function, so $y(B^c)$ contains the signs of the $h(B^c)$ components. The components of $|y(B^c)|$ are all 1.

If all components of $|y(B)|$ are less than or equal to 1, then y is a feasible solution to the dual problem, and by the equilibrium conditions x and y are optimal solutions to problem (1) and the dual, respectively.

If, on the other hand, some components of $|y(B)|$ are greater than 1, then y is not dual feasible, and now we proceed to reduce the objective value in problem (1) as follows.

Choose an index $s \in B$ such that $|y_s| > 1$. Define

$$\begin{aligned} t(B^c) &= -\left(\mathbf{sgn}(y_s)\right)\left(y(B^c)\right) .* \left(A(B^c, :) A(B, :)^{-1} e_j\right), \\ r &= \underset{j}{\operatorname{argmin}} \{ \frac{|h_j|}{t_j}, \quad | \quad j \in B^c \quad \text{and} \quad t_j > 0, \} \end{aligned}$$

where e_j is the vector which is 0 everywhere except the j-th entry, which is 1; and where j is chosen so that s is the j-th entry in B. In other words, $A(B, :)^{-1} e_s$ is the s-th column of $A(B, :)^{-1}$. Then the new index set is

$$\widehat{B} = B \backslash \{s\} \cup \{r\}.$$

The new solution $\widehat{x} = A(\widehat{B}, :)^{-1} b(\widehat{B})$ will lead to a reduced objective value in problem (1).

Our job in this project is to develop the above idea into a simplex-type algorithm for solving problem (1) under the non-degeneracy assumptions, based on the revised simplex tableau on $M = A(B, :)$ and $y(B)$. Note that the Phase I calculations for this problem consists of picking up any initial index set B with m indexes and computing $M^{-1} = A(B, :)^{-1}$ and $A(B, :)^{-1} b(B)$.

Your program should read A and b from an input file called `infile.txt`. The first line of `infile.txt` looks like

```
n m
```

followed by $n \times m$ lines that look like

```
i j aij
```

and the last n lines of `infile.txt` look like

`i bi`

Your program should also report its results on an output file called `outfile.txt`:

- **Failure:** Your code had failed without producing any meaningful result. Report in which phase does the code fail and whether this is due to *degeneracy* or *unknown problem*.
- **Success:** Your code has come to a successful stop. Report the optimal primal and dual solutions and the optimal objective value.

Email your code to the instructor by 23:59PM, Dec. 3, 2015.