

# Optimization, Programming Assignment #6

June 8, 2008

## Description

In this assignment, you will implement the log-barrier method for optimizing convex problems with linear equality and inequality constraints.

You should turn in an archive containing all of your source code, plots, and a document containing a brief description of your code (how to run it, what parameters it takes, etc), and answers to all questions. Email this archive to your TA, at [cotter@tti-c.org](mailto:cotter@tti-c.org).

*Please remember to turn in a document, and to answer the questions (in particular, talk about what your plots mean)!*

## 1 Log-barrier method

### 1.1 Implementation

Implement the log-barrier method (algorithm 11.1 of Boyd and Vandenberghe), for optimizing a convex problem subject to linear constraints:

$$\begin{aligned} \text{minimize} & : f(x) \\ \text{subject to} & : Fx \preceq g \\ & Ax = b \end{aligned}$$

Use your implementation of the infeasible start Newton's method (from programming assignment 5) to solve each of the subproblems, and note that this means that the initial point provided to your log-barrier implementation need not satisfy the equality constraints (though it must satisfy the inequality constraints).

Your function should take 14 parameters:

- $\nabla f$  and  $\nabla^2 f$  (the gradient and Hessian of the objective function)
- $F$  and  $g$  (the inequality constraints)
- $A$  and  $b$  (the equality constraints)
- $x^{(0)}$  (the initial point)
- $\epsilon_{\text{lb}}$ ,  $\mu_{\text{lb}}$  and  $t_{\text{lb}}^{(0)}$  (the parameters to the log-barrier method)
- $\gamma_{\text{nm}}$  and  $\epsilon_{\text{nm}}$  (the stopping criterion parameters to the infeasible start Newton's method)
- $\alpha_{\text{nm}}$  and  $\beta_{\text{nm}}$  (the line search parameters for the infeasible start Newton's method)

Note that the description of the log barrier method in Boyd and Vandenberghe approximates the indicator  $\mathbf{1}[x < 0]$  as  $\log -\frac{1}{t} \log(-x)$ , with  $t$  being updated as  $t = \mu t$  at every iteration. In class, the parameterization was different, with  $-\alpha \log(-x)$  being the approximation to the indicator (and  $\alpha$  shrinking, rather than growing, at every iteration). Otherwise, the descriptions of the log-barrier method in the book, and in class, are identical.

Your function should return the list of iterates  $x^{(k)}$ , one for every iteration of the log-barrier method (throw away all iterates of the infeasible start Newton's method except the last). In addition to the checks which should already be present in your implementation of the infeasible start Newton's method, make sure to do the following:

- Make sure that your function smoothly handles the cases where  $F, g$  and/or  $A, b$  are empty
- Check that  $\epsilon_{\text{lb}}, t_{\text{lb}} > 0$ , and  $\mu_{\text{lb}} > 1$
- Check that the matrices and vectors  $x^{(0)}, F, g, A$  and  $b$  have compatible dimensions
- Check that the initial point  $x^{(0)}$  satisfies the inequality constraints  $Fx \leq g$
- Comment any portion of your code of which the interpretation is not obvious

Also modify your implementation of the infeasible start Newton's method to take  $F$  and  $g$  as parameters, and modify the backtracking line search so that it will also backtrack (that is, multiply the step length  $t$  by  $\beta$ ) if  $F(x + t\Delta x) \geq g$ , just to make sure that you never encounter an iterate which does not satisfy the inequality constraints (note that we have a non-strict inequality  $\geq$  here because, for the log-barrier method, the inequalities should never be satisfied with equality).

## 1.2 Experiments

Consider a probability density  $p_i$  supported on  $i \in \{0, 1, \dots, n-1\}$ . Consider the problem of finding the maximum-entropy probability density which has some specified mean  $\mu \in [0, n-1]$ :

$$\begin{aligned} \text{minimize} & : \sum_{k=0}^{n-1} p_k \log p_k \\ \text{subject to} & : p_k \geq 0 \\ & \sum_{k=0}^{n-1} p_k = 1 \\ & \sum_{k=0}^{n-1} k p_k = \mu \end{aligned}$$

Optimize this problem for various values of  $n$  and  $\mu$ , using your implementation from problem 1.1. Calculate the ratios  $\frac{p_k}{p_{k+1}}$  at the optimum. What do you find? Can you make a conjecture about the general form of the maximum entropy probability density supported on  $\{0, \dots, n-1\}$  with some prespecified mean? Optionally, prove your conjecture.

## 2 A general QP solver

### 2.1 Phase I implementation

Implement the “big-M”-style phase I method (section 11.4.2 of Boyd and Vandenberghe) for finding an initial feasible point.

Your function should take 11 parameters:

- $\nabla f$  and  $\nabla^2 f$  (the gradient and Hessian of the objective function)
- $F$  and  $g$  (the inequality constraints)

- $A$  and  $b$  (the equality constraints)
- $t_{\text{lb}}$  (the scale of the log-barriers)
- $\gamma_{\text{nm}}$  and  $\epsilon_{\text{nm}}$  (the stopping criterion parameters to the infeasible start Newton's method)
- $\alpha_{\text{nm}}$  and  $\beta_{\text{nm}}$  (the line search parameters for the infeasible start Newton's method)

Your function should return a feasible initial weight vector  $x^{(0)}$ . Make sure to do the following:

- Make sure that your function smoothly handles the cases where  $F, g$  and/or  $A, b$  are empty
- Check that  $t_{\text{p1}} > 0$
- Check that the matrices and vectors  $F, g, A$  and  $b$  have compatible dimensions
- Comment any portion of your code of which the interpretation is not obvious

## 2.2 QP solver implementation

Create a wrapper function around your phase I and log-barrier implementations to solve a general quadratic program:

$$\begin{aligned} \text{minimize} & : \frac{1}{2}w^T Q w + w^T r \\ \text{subject to} & : Fx \preceq g \\ & Ax = b \end{aligned}$$

Your function should take 13 parameters:

- $Q$  and  $r$  (the objective function)
- $F$  and  $g$  (the inequality constraints)
- $A$  and  $b$  (the equality constraints)
- $\epsilon_{\text{lb}}, \mu_{\text{lb}}$  and  $t_{\text{qp}}$  (the parameters to the log-barrier method)
- $\gamma_{\text{nm}}$  and  $\epsilon_{\text{nm}}$  (the stopping criterion parameters to the infeasible start Newton's method)
- $\alpha_{\text{nm}}$  and  $\beta_{\text{nm}}$  (the line search parameters for the infeasible start Newton's method)

Your function should call your phase I implementation with  $t_{\text{p1}} = t_{\text{qp}}$ , then call your log-barrier implementation with the resulting initial point, and  $t_{\text{lb}}^{(0)} = \mu t_{\text{qp}}$ . Finally, return the list of iterates returned by the log-barrier method. Also do the following:

- Make sure that your function smoothly handles the cases where  $F, g$  and/or  $A, b$  are empty
- Check that the matrix  $Q$  and vector  $r$  have compatible dimensions
- Comment any portion of your code of which the interpretation is not obvious

## 2.3 Experiments

Consider the problem of optimizing a support vector machine for binary classification, with an unregularized bias term  $b \in \mathbb{R}^n$  (which is the only change from the description in class):

$$\begin{aligned} \text{minimize} & : \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{subject to} & : y_i (w^T x_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned}$$

Download the UCI “Pima Indians Diabetes Data Set” from <http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>. Read the data set description (the most important thing to get out of the description is which columns contain features, and which contains the labels). Note that the dataset file contains the classes  $\{0, 1\}$ , while for the SVM problem described above, we have  $y_i \in \{-1, 1\}$ , so be sure to map all 0s to  $-1$ s before attempting to optimize. The matlab file [http://bleu.uchicago.edu/CMSC34500/pa6/read\\_data.m](http://bleu.uchicago.edu/CMSC34500/pa6/read_data.m) demonstrates how the data file should be processed.

Write the above problem in the standard form of a QP:

$$\begin{aligned} \text{minimize} & : \frac{1}{2} w^T Q w + w^T r \\ \text{subject to} & : F x \preceq g \\ & A x = b \end{aligned}$$

### 2.3.1 Dependence of performance on $\mu_{lb}$

For this part of the problem, *use only the first 256 elements of the data set*. Use your implementation from problem 2.2 to find the weights  $w$  and bias  $b$  which minimize the SVM objective function above, using the following values of the constants:

- $C = 1$
- $\epsilon_{lb} = 0.1$  and  $t_{qp} = 1$
- $\gamma_{nm} = 10^{-8}$  and  $\epsilon_{nm} = 0.1$
- $\alpha_{nm} = 0.1$  and  $\beta_{nm} = 0.9$

Perform this optimization for all nine of the values  $\mu_{lb} \in \{2, 2^2, 2^3, \dots, 2^9\}$ , keeping track of the runtime of each. Create a plot of runtime versus  $\mu_{lb}$ , where the  $\mu_{lb}$  axis is in log scale. What is the shape of this plot? Can you (roughly) choose an optimal value for  $\mu_{lb}$ ?

### 2.3.2 Optimizing the full problem

*On the full dataset*, use your implementation from problem 2.2 to find the weights  $w$  and bias  $b$  which minimize the SVM objective function above, using the same values of the constants as in the previous section (you should choose the value of  $\mu_{lb}$ ):

In your writeup, include the optimal values of  $w$  and  $b$  which you found, as well as the classification error which your SVM experiences on the training set (that is, the proportion of elements for which  $\text{sgn}(w^T x_i + b) \neq y_i$ ).