# CS726 - Nonlinear Optimization I - Homework 9

November 26, 2012

This assignment is due at 5PM on Friday, December 14.

Note that this assignment is worth *two* homework grades. But you have 3 weeks to complete it. This is a programming assignment and should be submitted electronically using learn@uw. The assignment name is hwk9 and your code should be runnable from exactly one script called `run_hwk9.m` (or `run_hwk9.py` if running python). That is, I should be able to type `run_hwk9.m` (or `python run_hwk9.py`) and have all of your experiments executed and appropriate plots produced. Also, make a file `hwk9_commentary.pdf` with all of the plots and write-ups of your findings. In the pdf report, describe your implementation methodology. You will be partially graded on the computation time required to achieve the desired accuracy. Describe in detail what optimizations you implement to reduce computation time.

1. **Comparing classifiers.** Write an algorithm, based on ADMM, to solve the general risk minimization problem

$$\text{minimize} \quad \textstyle\sum_{k=1}^{n} \text{loss}(w^T x_k + b, y_k) + \lambda \|w\|_2^2$$

   Use the splitting based on introducing the equality constraint $z_k = w^T x_k + b$. Implement the following three loss functions:

   (a) $\text{loss}(z, y) = \max(1 - yz, 0)$.

   (b) $\text{loss}(z, y) = \frac{1}{2}(y - z)^2$.

   (c) $\text{loss}(z, y) = \log(1 + \exp(z)) - \left(\frac{y+1}{2}\right) z$.

   In your code use the stopping criteria proposed by Boyd *et al* in Equation (3.12) (page 19).

   Test your classification code on the`iris`, `breast_cancer.dat`, `adult.dat` data sets, available for download on the class website. In all cases, each row indexes an example. The first column corresponds to the label $y$, and the remaining columns are the $x$ data.

   In each case, tune the ADMM algorithm and regularization parameter $\lambda$ to get the best error over ten-fold cross-validation: break the data set up into 10 random chunks of equal size. For each chunk, use the other 90% of the data to train an SVM and then evaluate its performance on the held out chunk. The cross-validation error is the average number of misclassified data points over these 10 runs.

   How do the different loss-functions compare to one another after tuning? Describe how the different loss functions fare on each data set.

2. **Compressed Sensing.** Write an algorithm, based on projected gradient or ADMM to solve the $\ell_p$ regularized problem

$$\text{minimize} \quad \|Ax - b\|_2^2 + \mu \|x\|_p$$

Where $A$ is $m \times n$ for $p = 1, 2$. You may use whatever tricks you would like to make this algorithm as fast as possible. The goal will be to see how this optimization fares on random compressed sensing problems.

Generate your data as

$$b_i = A_{i,\cdot}x_0 + \omega_i$$

where $\omega_i \sim \mathcal{N}(0, \sigma^2)$. Generate $x_0$ to be the sparse vector equal to $1$ in $s$ randomly selected locations and $0$ everywhere else.

Test your algorithm on the following four random matrices

(a) $A_{ij} \sim \mathcal{N}(0, 1/n)$.

(b) $A_{ij} = \sqrt{3/n}$ with probability $1/6$, $-\sqrt{3/n}$ with probability 1/6, and $0$ with probability $2/3$.

(c) $A$ equals $m$ random rows from the discrete cosine transform, normalized so that each row has norm $1$.

Record the *prediction error* $\|A\hat{x} - b\|^2$ and *recovery error* $\|x_0 - \hat{x}\|^2$ where $\hat{x}$ is the estimate returned by your algorithms for $\ell_2$ and $\ell_1$.

For $n = 1000$, $m = 250$ and $s = 50$, find the value of $\mu$ that achieves the lowest errors possible, plotting the errors and computation time as a function of $\mu$. Make these plots using the values $\sigma = 0, 0.001, 0.1$ for the noise. When does $\ell_1$ outperform $\ell_2$ and vice versa?