Project 3

due: 11/17/2015

KNN (40 points)
Download the data set in KNNdata.mat. This MATLAB file contain three variables: X, Y, and $Xtest$. X is 1,200 objects, each defined by 8 features. Y is the label of each X (there are 3 classes). $Xtest$ is 240 test vectors for which you do NOT have the label (I have the label, but will not release it to you).

Part 1) Build a k-nearest neighbor (KNN) classifier that uses the leave-one-out cross validation approach to determine best value for k. Your submission should be in the form of a MATLAB function with the following inputs / outputs (I/O):

- `[prediction, bestk, errors] = myKNN(X, Y, Xtest, k)`
- `X` is $[N \times d]$ matrix of training data
- `Y` is $[N \times 1]$ vector of training labels
- `Xtest` is $[Ntest \times d]$ matrix of test data
- `k` is a $[Nk \times 1]$ vector of candidate k values
- `prediction` is a $[Ntest \times 1]$ vector of label predictions for Xtest
- `bestk` is the value k chosen by the leave-one-out cross validation
- `errors` is a $[Nk \times 1]$ vector of the number of errors found at each k in the cross-validation

Part 2) Build a weighted KNN classifier, including a method for determining the best λ parameter in the weighting function (use the weighting function described in the class notes). Your submission should be in the form of a MATLAB function with the following I/O:

- `[prediction, bestlambda] = myWKNN(X, Y, Xtest)`
- Inputs and outputs are the same as the `myKNN` function

Not that this function does not have an input for the candidate λs. You will have to code that into your approach.

Your grade on this code will depend on the following:

10 pts Your functions run and return a proper output (the size of the outputs are correct)

15 pts Your outputs from `myKNN` are correct (note, there is a perfect value for k that produces the lowest leave-one-out cross-validation error)

15 pts Your outputs from `myWKNN` are correct (note, I am able to get down to about 100 training errors with the correct λ)

SVM (60 points)
Download the data sets in SVMdata.mat. This MATLAB file contains three variables for three different data sets: $K1$ and $Y1$, where $K1$ is the $[N \times N]$ training data kernel matrix and $Y1$ are the training labels for the first dataset (and similarly for $K2$, $Y2$, etc.). Implement the dual form of the SVM classifier in MATLAB using MATLAB's `quadprog` solver. Note, do not use MATLABs built-in SVM solver (you can use it to compare if you like). Your submission should be in the form of a MATLAB function with the following inputs / outputs (I/O):

- `[prediction, alpha, b] = mySVM(K, Y, Kt, dataset)`
- `K` is $[N \times N]$ kernel matrix of training data (i.e., $K_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$)
- `Y` is $[N \times 1]$ vector of training labels ($y \in \{-1, +1\}$)
- `Kt` is $[N \times Ntest]$ kernel matrix of test data (i.e., $[Kt]_{ij} = \phi(\mathbf{x}_i^{train}) \cdot \phi(\mathbf{x}_j^{test})$)
- `dataset` is the identifier of the datasets $\{1, 2, 3\}$. (Note that other values of `dataset` should provide a default box-constraint $C = 1$, see below)
- `prediction` is the $[Ntest \times 1]$ predicted labels of the test data
- `alpha` is the $[N \times 1]$ output of the SVM
- `b` is the scalar bias output of the SVM

The `dataset` input is for your code to set the box-constraint C for each of the three provided datasets. I recommend using cross-validation to choose a best value of C for each of the provided data sets. Make sure to include in your code a default value of $C = 1$ if `dataset` is set to something other than 1, 2, or 3.

Here are the equations of the SVM for your reference (also see the notes and textbooks, where appropriate).

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - 0.5 \sum i, j = 1^N y_i y_j \alpha_i \alpha_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j),$$
$$\text{s.t.} \quad 0 \geq \alpha_i \geq C,\ i = 1, \dots, N,$$
$$\sum_{i=1}^{N} \alpha_i y_i = 0.$$

The prediction equations are

$$\begin{aligned} f(\mathbf{x}^{test}) &= \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i^{train})^T \phi(\mathbf{x}^{test}) + b, \\ y^{test} &= \text{sgn}\{f(\mathbf{x}^{test})\}, \end{aligned}$$

where b can be found by

$$b = y_j - \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j), \quad \text{for any } j, \text{ s.t. } 0 < \alpha_j < C.$$

Your assignment will be graded on the following criteria:

20 pts Your `mySVM` code runs and returns a proper output (the size of the outputs are correct)

20 pts Your outputs are correct for the provided data sets (note, you can compare against MATLAB's SVM for verification). The number of points will be directly proportional to the error rate between my code (and my choice of C) and yours.

20 pts Your outputs are correct for the sequestered data sets. The number of points will be directly proportional to the error rate between my code and yours. Note, I will use $C = 1$ for these data sets.