CPTS 570 Machine Learning, Fall 2016
Homework #2
Due Date: Oct 6

NOTE 1: Please use a word processing software (e.g., Microsoft word or Latex) to write your answers and submit a printed copy to me at the begining of the class on Oct 6. The rationale is that it is sometimes hard to read and understand the hand-written answers. Thanks for your understanding.

NOTE 2: Please ensure that all the graphs are appropriately labeled (x-axis, y-axis, and each curve). The caption or heading of each graph should be informative and self-contained.

1. (**10 points**) Suppose we have n_+ positive training examples and n_- negative training examples. Let C_+ be the center of the positive examples and C_- be the center of the negative examples, i.e., $C_+ = \frac{1}{n_+}\sum_{i:\ y_i=+1} x_i$ and $C_- = \frac{1}{n_-}\sum_{i:\ y_i=-1} x_i$. Consider a simple classifier called CLOSE that classifies a test example x by assigning it to the class whose center is closest.

   - Show that the decision boundary of the CLOSE classifier is a linear hyperplane of the form $sign(w \cdot x + b)$. Compute the values of w and b in terms of C_+ and C_-.
   - Recall that the weight vector can be written as a linear combination of all the training examples: $w = \sum_{i=1}^{n_+ + n_-} \alpha_i \cdot y_i \cdot x_i$. Compute the dual weights (α's). How many of the training examples are support vectors?

2. (**10 points**) Suppose we use the following radial basis function (RBF) kernel: $K(x_i, x_j) = exp(-\frac{1}{2}\|x_i - x_j\|^2)$, which has some implicit unknown mapping $\phi(x)$.

   - Prove that the mapping $\phi(x)$ corresponding to RBF kernel has infinite dimensions.
   - Prove that for any two input examples x_i and x_j, the squared Euclidean distance of their corresponding points in the higher-dimensional space defined by ϕ is less than 2, i.e., $\|\phi(x_i) - \phi(x_j)\|^2 \leq 2$.

3. (**5 points**) The decision boundary of a SVM with a kernel function (via implicit feature mapping $\phi(.)$) is defined as follows:
   $w \cdot \phi(x) + b = \sum_{i \in SV} y_i \alpha_i K(x_i, x) + b = f(x; \alpha, b)$
   , where w and b are parameters of the decision boundary in the feature space phi defined by the kernel function K, SV is the set of support vectors, and α_i is the dual weight of the i^{th} support vector.

   Let us assume that we use the radial basis function (RBF) kernel $K(x_i, x_j) = exp(-\frac{1}{2}\|x_i - x_j\|^2)$; also assume that the training examples are linearly separable in the feature space ϕ and SVM finds a decision boundary that perfectly separates the training examples.

   If we choose a testing example x_{far} that is far away from any training instance x_i (distance here is measured in the original feature space $\Re^d$). Prove that $f(x_{far}; \alpha, b) \approx b$.

4. (**5 points**) The function $K(x_i, x_j) = -\langle x_i, x_j \rangle$ is a valid kernel. Prove or Disprove it.

5. (**5 points**) You are provided with n training examples: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n,)$, where x_i is the input example, y_i is the class label (+1 or -1). The teacher gave you some additional information by specifying the costs for different mistakes C_+ and C_-, where C_+ and C_- stand

for the cost of misclassifying a positive and negative example respectively.
a. How will you modify the Soft-margin SVM formulation to be able to leverage this extra information? Please justify your answer.

6. **(5 points)** Consider the following setting. You are provided with n training examples: $(x_1, y_1, h_1), (x_2, y_2, h_2), \cdots, (x_n, y_n, h_n)$, where x_i is the input example, y_i is the class label (+1 or -1), and $h_i > 0$ is the importance weight of the example. The teacher gave you some additional information by specifying the importance of each training example.
a. How will you modify the Soft-margin SVM formulation to be able to leverage this extra information? Please justify your answer.
b. How can you solve this learning problem using the standard SVM training algorithm? Please justify your answer.

7. **(25 points)** You are provided with a set of n training examples: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n,)$, where x_i is the input example, y_i is the class label (+1 or -1). Suppose n is very large (say in the order of millions). In this case, standard SVM training algorithms will not scale due to large training set.
Tom wants to devise a solution based on "Coarse-to-Fine" framework of problem solving. The basic idea is to cluster the training data; train a SVM classifier based on the clusters (coarse problem); refine the clusters as needed (fine problem); perform training on the finer problem; and repeat until convergence. Suppose we start with k_+ positive clusters and k_- negative clusters to begin with (a cluster is defined as a set of examples). Please specify the mathematical formulation (define all the variables used in your formulation) and concrete algorithm for each of the following steps to instantiate this idea:
a) How to define the SVM training formulation for a given level of coarseness: a set of k_+ positive clusters and a set of k_- negative clusters?
b) How to refine the clusters based on the resulting SVM classifier?
c) What is the stopping criteria?
Optional question: For what kind of problems will this solution fail?

8. (**25 points**) Empirical analysis question.

   Download and install the LibSVM software: http://www.csie.ntu.edu.tw/c̃jlin/libsvm/

   You are provided with a training set, and testing set of multi-class classification examples (Please use the 10 folds data from the first homework). Please divide the training data into sub-train (80 percent) and validation (20 percent) for your experiments. You will run the LibSVM software on the training data to answer the following questions. Repeat each experiment for all the 10 different folds and compute the averaged results.

   (a) Using a linear kernel (-t 0 option), train the SVM on the training data for different values of C parameter(-c option): $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}, 10^{4}$. Compute the training accuracy, validation accuracy, testing accuracy for the SVM obtained with different values of the C parameter. Plot the training accuracy, validation accuracy, and testing accuracy as a function of C (C value on x-axis and Accuracy on y-axis) – one curve each for training, validation, and testing data. Also, plot the number of support vectors (see the training log at the end of SVM training) as a function of C. List your observations.

   (b) Select the best value of hyper-parameter C based on the accuracy on validation set and train a linear SVM on the *combined* set of training and validation examples. Compute the testing accuracy and the corresponding confusion matrix: a $k \times k$ matrix, where k is the total number of classes.

   (c) Repeat the experiment (a) with polynomial kernel (-t 1 -d option) of degree 2, 3, and 4. Compare the training, validation, testing accuracies, and the number of support vectors for differnt kernels (linear, polynomial kernel of degree 2, polynomial kernel of degree 3, and polynomial kernel of degree 4). List your observations.