CMPS 142 Third Homework, Spring 2015

3 Problems, 12 pts, due start of class Tuesday 5/12 (revised)

This homework is to be done in groups of 2 or 3. Each group members should completely understand the group's solutions and *must* acknowledge all sources of inspiration, techniques, and/or helpful ideas (web, people, books, etc.) other than the instructor, TA, and class text. Each group should submit a single set of solutions containing the names and e-mail addresses of all group members. Although there are no points for "neatness", the TA may deduct points for illegible or poorly organized solutions.

1. (1 pt) Suppose that we have the following training set (where the instances have two features):

| $x_1$ | $x_2$ | label |
|---|---|---|
| 1 | 1 | +1 |
| 1 | 2 | +1 |
| 2 | 1 | +1 |
| 0 | 0 | -1 |
| 1 | 0 | -1 |
| 0 | 1 | -1 |

   Plot them and find the optimal separating line. What are the support vectors, and what is the margin?

2. (7 pts) Support Vector Machine Kernels
   We'll use LibSVM to explore the effects of kernels on SVM performance. Install LibSVM through the Weka package manager. It will be available in functions → LibSVM.

   There is a guide to using SVMs for classification at `http://www.csie.ntu.edu.tw/ cjlin/papers/g`
   They recommend beginners try the following first:

   - Transform data to the format of an SVM package
   - Conduct simple scaling on the data
   - Consider the RBF kernel $K(\mathbf{x}; \mathbf{y}) = e^{-\gamma\|\mathbf{x}-\mathbf{y}\|^2}$.
   - Use cross-validation to find the best parameter $C$ and $\gamma$.
   - Use the best parameter $C$ and $\gamma$ to train the whole training set.
   - Test

   In this exercise, we'll work with the UCI dataset spambase, which can be obtained from `https://archive.ics.uci.edu/ml/datasets/Spambase`. Open the file spambase.arff (there are several versions on the web, we will put one in the resources shortly). You may have to run the filter supervised → attribute → NominalToBinary to change the categorical attributes to numeric (binary) attributes. You will also want to normalize the data.

Use grid search (available under classifiers $\rightarrow$ meta) to run LibSVM and optimize the parameters classifier.cost and classifier.gamma. There is a page on using Weka's grid search at http://weka.wikispaces.com/Optimizing+parameters. There is a section on optimizing the parameters for an RBF kernel using Weka's SMO algorithm.

Pick some values for the minimum and maximum values the grid search will look at, and the step size. When grid search is ran, it will first build a model over the whole training set and report the values chosen. As SMO can take some time to run, don't use too many grid points in your grid search, *and get started early!*

Which values are chosen for $C$ and $\gamma$? Are these values at the boundary of what it could have selected? What is the accuracy?

Compare against a polynomial kernel of degree 2 in LibSVM. Make sure to optimize the $\gamma$ and "coef0" parameters. How did it compare?

Compare against the default Linear SVM, which uses a linear kernel (just the standard dot product. Make sure to optimize the C (cost) parameter. How did it compare?

3. (4 pts) Consider learning a decision tree with boolean features where the impurity criterion is the error rate. Therefore the attribute resulting in the fewest errors at the next level (when each subtree's examples are predicted with the majority label in the subtree) will be selected as the test.

   Construct a simple training set with binary features and labels where:

   (a) There is small decision tree that correctly classifies the training data,

   (b) The greedy decision tree construction algorithm finds a larger tree (i.e. more nodes), and

   (c) There are no ties in the construction process (i.e. the decision tree algorithm finds a single best attribute to test at each node).

   List your training set and draw the small decision tree that correctly classifies it. Illustrate how the decision tree algorithm runs on your data, and show the resulting tree. You may duplicate examples in your training set if you find it helpful. For full credit, construct a training set where the tree found by the decision tree algorithm has more than twice as many nodes as the smallest tree correctly classifying the data.