CS578 Empirical Methods in Machine Learning and Data Mining
Course Project

Predictions are due Tuesday December 11, 2007 at 11:59PM.
Reports are due Thursday December 13, 2007 at 11:59AM.
Late predictions and reports will *not* be accepted.
Turn in project write-ups to Melissa Totman in 4147 Upson.
Slide reports under Melissa's door if she is not there.

The goal of this project is to apply decision trees, neural nets, k-nearest neighbor, and/or SVMs to a data set using any/all of the methods from the course to improve performance. These include:

- bagging and boosting
- cross validation
- model averaging (combining predictions from two or more models or learning methods)
- early stopping
- feature re-coding
- feature selection
- feature weighting
- distance metric hacking
- ...

You will be given two data files, a train set and final test set. The train set will contain 25,000 cases and the test set will contain 50,000 cases. The targets (0 or 1) are in the *last* column. The first columns are the input attributes. A .attr file with the names and ranges of these attributes is included. The data set is from a real ornithology problem we are working on.

You can do anything you want with the training set. We strongly encourage you to use cross validation to create your own test sets from this training set so that you get unbiased estimates of the performance of the methods you try.

The test set will not contain targets! -- all of the targets have been replaced with 0. The last column, which would contain the targets, will instead contain all zeros. You will run your final model on the test set and submit predictions to us via a web interface. We will compute the performance of your method on the first 100 cases in the test set, and maintain a live results table with the performances of all groups on these 100 cases. Your performance on all of the test cases will be used as part of your grade for the project.

You are allowed *ten* submissions for each metric (see below). We will only consider your *last* submission for each metric, so be careful. We recommend that you only plan to use nine out of the ten allowed submissions and keep the tenth as a backup in case something goes wrong.

You may work on the project in groups of 1-4 students. If you work in a group, briefly document who does what. For example:

"X was responsible for decision trees, implementing cross validation for all the experiments, and preprocessing the data. Y did neural nets and k-NN and looked at feature weighting in k-NN (which helped, but not enough to make k-NN competitive with bagged trees and neural nets). Z implemented feature selection and bagging, and generated most of the graphs in this report. As a group we ran SVMs, decided how to do cross validation before starting the project, and decided which model performed best at the end of the project.

The project will be graded as follows:

50% TECHNICAL APPROACH:
How well did you tackle the problem?
What method(s) did you use to optimize performance?
How well did you do them?
How well did you interpret the results?
Can you tell us interesting findings about the data or the factors that affect the class the most?
The project is open ended and you are expected to think about how to find/train good models in the allotted time. You can't try all possible combinations of methods. It is important to create a plan for tackling the problem, and adjust the plan as you collect intermediate results.

25% WRITE-UP:
Is your report clear, concise, and complete? The write-up should outline your plan for tackling the problem, and summarize the performance of all the models you trained. The write-up should clearly state what model you think is best and how the final model is trained. You must include estimates of roughly how well you think the final model should perform on the final test set (based on the performance you observe on your own test sets). How accurate these estimates are will be part of your grade. Reports that are short will get better grades than reports that are long, rambling, or present too much detail.

25% PERFORMANCE ON THE FINAL TEST SETS:
We'll measure the accuracy, RMS, and ROC Area of your predictions. Because a model that optimizes accuracy might not be optimal for ROC Area or RMS, you will submit different predictions for accuracy, for RMS, and for ROC Area. It is OK if the predictions you submit for accuracy are the same as the ones you submit for RMS and ROC Area -- you don't have to submit *different* sets of predictions.

EXTRA CREDIT IF YOU USE SVMs:
To encourage you to try SVMs, we'll give you 5 points of extra credit if you do a reasonable set of experiments using SVMs. You do not have to use the SVMs as part of your final models when you make predictions. You get the extra points just for doing a good set of SVM experiments and showing us the results. We suggest you use Thorsten Joachims SVM code available at http://svmlight.joachims.org/. This package is easy to install on a number of platforms.

When submitting predictions via the web interface, they should be in the following format:

- The file should be plaintext, and contain exactly 50,000 numbers, with exactly one number per line.

  IMPORTANT! You must return predictions to us in the same order as the cases in the unlabeled final test sets!

  Sample Uploaded File:

  ```
  0.66
  0.09
  ...   67,511 more predictions
  0.59
  ```

Comments about the format:
- Probabilities can use any reasonable number of significant digits.
- The probability to give us is the probability the item is class 1!
- The order of the predictions is critical!