Homework 9: Classifiers

36-350, Data Mining, Fall 2009

Due Monday, 30 November 2009 at 5 pm

For this assignment, you will need the `zip.train` and `zip.test` data sets from the `ElemStatLearn` library. You used `zip.train` in a previous problem set.

We need to do a little manipulation to make sure fitting routines treat the digit label as a categorical variable ("factor") and not a number where fractional values make sense:

```
zip.train <- as.data.frame(zip.train)
zip.train[,1] <- as.factor(zip.train[,1])
```

and similarly for `zip.test`. Also, it is helpful to give the columns of the data sets names:

```
colnames(zip.train) <- c("Y",paste("X.",1:256,sep=""))
```

Include code or commands for everything you do. Do not report any numerical results to excess precision.

1. Fit a classification tree to `zip.train`. (Make sure that it is a classification tree and not a regression tree!) Include a plot of the tree. What is its in-sample error rate? Give a confusion matrix for real vs. predicted digits. (The `table` command is helpful here.) Are there any noteworthy patterns in the confusion matrix?

2. Use 10-fold cross-validation to prune the tree. How much does it shrink? Repeat the analysis of errors and confusion for the pruned tree.

3. Using the `svm` function from the `e1071` package, fit an SVM to `zip.train`, with `cost=1` (the default). (Make sure you are getting a classifier SVM and not one doing regression.) How many support vectors are there? What is the in-sample error rate? Make a confusion matrix; how does it differ from that of the tree?

4. Use 10-fold cross-validation to select a `cost` setting between 0.1 and 10 (inclusive) for the SVM. What is the selected cost? Repeat the analyses of the previous question.

5. Apply the CV-selected tree and SVM to the `zip.test` data. (If you could not get cross-validation to work, use the baseline versions.) What are the error rates? The confusion matrices? How do these differ from the error rates and confusion matrices you got by running these models on `zip.train`?

6. Extra Credit There are several packages on CRAN which do bagging and boosting, e.g., `adabag`. Use one of them to fit an ensemble of 50 trees to `zip.train`, using both bagging. How well does the ensemble perform in-sample? How different are its predictions from those of the single tree you fit by cross-validation? How well does it predict `zip.test`? Now repeat for boosting. (Both the fitting steps could take half an hour or more.)