36-350: Data Mining

Lab 12

Date: November 15, 2002 **Due: end of lab**

1 Introduction

This lab teaches you how to construct and evaluate logistic regression classifiers.

There are 5 questions. For each one, submit your commands and a response from R demonstrating that they work. (Only hand in commands relevant to the question.) To submit a plot, click on the plot window and select

```
File -> Save as -> Postscript...
```

This saves the plot to a file which can be printed, incorporated into a Word document, or mailed to us as an attachment.

2 Starting R

Start R as in lab 1. On the class web page, go to "computer labs" and download the files for lab 12 into your work folder. Read the special functions into your running R application via the commands

```
source("lab11.r")
```

If this fails, check that the files were downloaded correctly.

3 The data

The dataset used in this lab is the same as in lab 11. It describes the repayment behavior of 1000 individuals who acquired loans from a bank. The bank would like to use this data to decide which customers in the future are likely to repay a loan. Each individual is described by 21 variables, the most important being `Class` which classifies the loan as good or bad. Load this data via

```
load("Credit.rda")
```

This defines a matrix of training data called `x.tr` and a matrix of test data called `x.te`. Your job is to construct a classifier from the training data which has high accuracy in predicting `Class` on the test data.

4 Description of functions

The format of this lab resembles a real-life situation in which you have a set of tools at your disposal and must figure out how to combine them to solve problems. Here the R functions you need are all listed up front.

For this lab, it will probably be useful to make a notepad file containing your R code, which you can edit ahead of time and then cut and paste into the R command window.

In the descriptions below, angle brackets denote code that you must fill in (such as `<formula>`). What you fill in can be any piece of R code, such as variable name or a mathematical expression.

These functions create classifier objects (tree, nearest-neighbor, and logistic regression, respectively):

```
tree(<formula>,<data>)
knn.model(<formula>,<data>)
logistic(<formula>,<data>)
```

The formula specifies the predictor/response variables and data is a matrix whose columns are the variables. The output is an object which can be placed into variables and passed to other functions. (If you don't assign to a variable, the object will simply print out to the screen.)

Cross-validation:

```
best.size.tree(<tree object>)
best.k.knn(<knn object>)
```

These functions accept a classifier object and return a new classifier object whose parameters have been optimized via cross-validation.

Quadratic expansion:

```
expand.quadratic(<object>)
```

This function returns a formula which contains all cross terms and squared terms of the predictors used by the object. (Object can be a classification object or a formula). It is similar to `expand.cross` from lab 10.

Examining classifier objects:

```
plot.graph.tree(<tree object>)
summary(<logistic object>)
```

Evaluating performance:

```
misclass(<object>,<data>)
confusion(<object>,<data>)
confusion(<object>,<data>,<prob>)
```

`misclass` returns the number of times the classifier chose a class different than the true class given in the data matrix. `confusion` is described in section 6.

5 Constructing classifiers

Question 1: These are the proportions of good and bad loans in the test set:

```
  Bad  Good
0.286 0.714
```

What is the misclassification rate (the number of errors divided by the test set size) of a classifier which always reports "Good"? What is the misclassification rate of a classifier which always reports "Bad"? The minimum of these two is the *baseline rate.* Any classifier which does worse than the baseline is essentially worthless.

Question 2: (a) Submit code to train a pruned classification tree, a k-nearest-neighbor classifier with best k, a linear classifier, and a quadratic classifier on the training set. (b) Compute the misclassification rate of each classifier on the test set. (You will want to keep a copy of these for the homework.) Which are below baseline?

Homework tip: you may find it easier to answer question 2(a) on homework 12 if you examine the tree classifier and linear classifier using `plot.graph.tree` and `summary`. What is the main difference?

6 Incorporating misclassification costs

Evaluating classifiers according to misclassification rate assumes that both kinds of misclassifications are equally bad. In reality, it costs more to grant a bad loan than not to grant a loan at all. Suppose the bank has the following cost matrix:

```
      predicted
truth  Bad Good
  Bad    0    5
  Good   1    0
```

This table specifies the cost of each *(predicted class, true class)* combination. It is available in R as the variable `costs`. (This is a simplification, since in reality it depends on the amount and duration of each loan.)

Question 3: What is the average cost (total cost divided by the test set size) of a classifier which always reports "Good"? What is the average cost of a classifier which always reports "Bad"? The minimum of these two is the *baseline cost.*

To evaluate classifiers according to cost, use the function `confusion`. `confusion` is like `misclass`, but gives more detail. It returns a matrix counting all of the *(predicted class, true class)* combinations on the data set. This matrix is set up so that, if you directly multiply the confusion matrix and the cost matrix, the sum is the total cost on the data. `confusion` has an optional third argument `p` which will be discussed later.

Question 4: For each of the four classifiers in Question 2, submit code to compute the average cost on the test set. (Recall that vectors and matrices in R can be multiplied, divided, etc. just like numbers. Also recall the function `sum`.) Which are below baseline?

You will find that the classifiers perform rather poorly. Why is this? By default, each classifier reports the class which is most probable. But when misclassification costs are not symmetric, this is not the best strategy. Instead, a loan should be classified as "Good" only if its probability is above some threshold p, not necessarily 0.5. This threshold is the third argument to `confusion`. Not specifying p is equivalent to specifying $p = 0.5$. Try running `confusion` with different values of p to get a feel for its effect.

The optimal value of p in this problem is 5/6. This comes from the formula for expected cost:

```
Cost(predict Good) = Cost(predict Good | truth Bad) p(truth Bad) +
                     Cost(predict Good | truth Good) p(truth Good)
Cost(predict Bad) = Cost(predict Bad | truth Bad) p(truth Bad) +
                    Cost(predict Bad | truth Good) p(truth Good)
```

The optimal p is such that, if the probability of "Good" is p, then the expected cost of predicting "Good" is the same as the expected cost of predicting "Bad".

Question 5: Using the optimal threshold above, recompute the average cost of each classifier on the test set. Which are below baseline now? (To check that you have the right threshold, you should notice that the confusion matrices are biased away from costly events.)