Naive Bayes Lab (Lab 2)

1. Accidents Data E.D.A.

Obtain the accidents data (Accidents.xls) from www.dataminingbook.com, in the "First Edition Materials" section. This requires a free registration. Open the data set in Excel and explore the information available on the different tabs of the spreadsheet, including the meanings of the variables and their possible values. Click on the fifth tab, "Data", to display the data we will use, then do Save As and choose "CSV: (comma delimited)" as the file type. You will get several warnings; hit OK. Then read the data into R using the read.table command. Make sure that you handle the header in the csv file correctly, so that it does not turn into the first line of data in your data frame (see help(read.table)). Also, when you call read.table you will need to specify what character separates the entries in your csv file.

Check that the resulting object is a data frame. Make sure that the column names are basically correct. How many accidents are in the data set?

The predictor variables are those that we might use to predict the outcome, namely variables 1-19 in the data set. Variable 20 is the outcome that we are using and variables 21-24 are other quantities that could also be used as outcome variables, although we will ignore them here (e.g. whether there was a fatality). Obtain the joint distribution of our outcome variable with one of the predictor variables. To obtain the joint distribution of two variables, use the "table" command in R to obtain a table that shows the number of occurrences of each of the pairs of values, e.g.

```
> jointCounts = table( accid$INJURY_CRASH , accid$INT_HWY )
```

Divide these counts by the total count (total number of accidents) to get the frequencies. This can be done by simply calling

```
> jointCounts / nData
```

where nData is equal to the total count value. Note the more common and less common pairs of values in the joint distribution.

We will now obtain the distribution of the same predictor variable conditional on the value of the outcome variable. Recall that the conditional distribution is equal to the joint distribution of the two variables, divided by the marginal distribution of the variable being conditioned upon. You have found the joint distribution of the two variables. The marginal distribution of the outcome variable is just the frequency of each value of the outcome variable, as obtained from:

```
> table( accid$INJURY_CRASH ) / nData
```

You could find the conditional distribution by dividing each entry in the joint frequency table by the appropriate entry in the marginal frequency vector. However, there is an easier way. Take the original joint counts table for the two variables as obtained from the table function. Then for each possible value of the outcome variable, take the corresponding row or column in the joint counts table. Then scale so that this vector sums to one; in other words, divide the vector by its sum.

If the resulting distribution does not change much with the different values of the outcome variable, then the two variables (predictor and outcome) may be independent. If the resulting distribution is very different for different values of the outcome variable, then the two variables are probably dependent.

The naive Bayes classifier is based on these conditional distributions. The predictor variables that are highly dependent with the outcome variable are those that will be most influential in the classifier.

2. Naive Bayes Training on the Accident Data

Obtain the naiveBayes.R file from Blackboard. Open this file in R, and copy and paste its contents to the command line. This file defines the functions nb.train and nb.predict, which perform naïve Bayes training and prediction, respectively. These functions have been adapted from the nbc-r project on Google Code, and have a GPL-2 open-source license. Any distribution or use of these functions is governed by this license.

We will now apply the nb.train naive Bayes fitting function to the accident data. This function takes a data frame as input, where the rows are the records and the columns are the variables; all of the columns are assumed to be predictors except the last column, which is assumed to be the outcome. We must therefore remove the extra columns from our data set by calling

```
> accid = accid[,(1:20)]
```

Next we will apply the nb.train function to the accid data set. We do not have a separate test data set so let's leave aside the first 1,000 records in the data set when we fit the naïve Bayes model. First create a data frame, e.g., trainData, that contains all of the data except the first 1,000 records (accidents). Then call nb.train(trainData), and assign the result to another object. View the resulting object by typing its name at the command line.

The object is a list with two elements, class.dist and attr.dist. class.dist is the marginal distribution of the outcome variable and attr.dist is a list with elements equal to the conditional distributions of each of the predictor variables (conditioned on the outcome variable). Use these to check the marginal and conditional distributions that you obtained earlier for the INJURY_CRASH and INT_HWY variables.

3. Naive Bayes Prediction on the Accidents Data

Now let's apply the nb.predict function to predict the outcome for the first 1000 rows in the data set:

```
> accidPred = nb.predict( D = accid[(1:1000),], nb = accidModel )
```

where accidModel is the object that resulted from the call to nb.train. This takes a minute or two to run! The object accidPred is a list with three elements: a vector of the predictions, a table of counts for the predicted vs. actual values, and the error rate as a decimal.

What is the error rate on the first 1000 data points in the data set?

Naive Bayes Homework (Homework 2)

1.
Please hand in your answer to the question at the end of the lab:
What is your error rate on the first 1000 data points in the data set? Express to three significant figures.

2. Fixing Zero Probabilities in the Prediction

Looking again at the conditional probabilities in the accidModel object, you will notice that there are some zeros. These zeros occur when there are no data points in the data set with that particular combination of values for the outcome and predictor variables. In what way can these zeros affect the predictions? How might these zeros occur just by chance?

In order to train the naive Bayes classifier, we had to estimate the marginal distribution of the outcome variable and the distributions of each of the predictor variables conditional on the outcome. We estimated them as being equal to the corresponding frequencies in the data set, which seems like a reasonable approach. These are called the "frequentist" estimates. However, this is not the only way to estimate the marginal and conditional distributions.

We have seen that when there are very few data points in the data set with a particular value for a predictor variable, the frequentist estimates can be very noisy. In order to mitigate the problems caused by this noise, it is common to instead use "Bayesian" estimates of the marginal and conditional distributions. Yes, you may think that we already doing something "Bayesian" since we are using the "naive Bayes" model, but in fact these are two different uses of the word.

Intuitively, a Bayesian estimate combines the information in the data with some prior information about the marginal and conditional distributions. The resulting estimate depends on the prior information that is specified, so there are multiple Bayesian estimates. One reasonable prior specification is to say that we want to treat all possible values for the marginal and conditional probabilities as equally probable before seeing the data. It turns out the resulting Bayesian estimates simply correspond to adding 1 to each count in the marginal and joint counts tables. In other words, the joint counts table from the data for a predictor and the outcome might look like:

	1	2	4
0	12	1	5
1	15	0	2

where the possible values of the outcome correspond to the rows (0 and 1) and the values for the predictor correspond to the columns (1, 2, and 4). Then a "prior count" of 1 would be added to each element of the table, yielding:

	1	2	4
0	13	2	6
1	16	1	3

In order to obtain the Bayesian estimated conditional distribution, one would then divide by the row counts in the updated table, yielding:

	1	2	4
0	0.619	0.095	0.286
1	0.800	0.050	0.150

Change the nb.train function to use the Bayesian estimates rather than the frequentist estimates (you don't have to change how the marginal distribution of the outcome is estimated). Again train on all the data except the first 1,000 records, and test on the first 1,000 records.

To be handed in:
After changing the nb.train function to use the Bayes estimates, what is the error rate on the first 1000 data points in the data set? Express to three significant figures. Was there improvement over using the frequentist estimates? Hand in your altered nb.train function.

3. True or false:

a. ROC curves are a measure of how well the model assumptions hold.

b. The overall misclassification rate cannot be greater than the maximum of the false positive rate and the false negative rate.

4. Say the application of a classification tree to the test data results in the following "classification matrix" (matrix of predicted vs. actual counts):

	Predict Y=0	Predict Y=1
Actually Y=0	271	345
Actually Y=1	592	520

What is the false positive rate? The false negative rate? The overall error rate?

5. Say we have a classifier that, if X1=0, flips a biased coin; with probability p0 the coin is heads. If the coin is heads we predict Y=1; otherwise, we predict Y=0. If X1=1, we flip a different biased coin; with probability p1 the coin is heads. If this coin is heads we predict Y=1 and otherwise we predict Y=0. Say that Pr(X1 = 1 | Y=1) =0.3 and Pr(X1=1 | Y=0) = 0.6. What is the false positive probability of our classifier? What is the false negative probability?