Assignment 2

Assigned: Tue Jan 23, 2001
Due: Tue Feb 6, 2001

In this assignment, you will have the opportunity to review your understanding of probability and explore two simple machine learning techniques (Naive Bayes classifier, *k*-Nearest Neighbor classifier). Your investigations will utilize two different data sets.

- **titanic**: The titanic data set gives the values of four categorical attributes for each of the 2201 people on board the Titanic when it struck an iceberg and sank. The attributes are social class (first class, second class, third class, crew member), age (adult or child), gender, and whether or not the person survived. The titanic data set is available at
  `http://www.cs.toronto.edu/~delve/data/titanic/desc.html`
- **credit-approval**: The credit-approval data set contains information about 690 applicants for credit and whether they were approved or rejected. Each application is described by 15 attributes and classified as approved ("+") or rejected ("–"). Information about the raw data base can be obtained at
  `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/crx.names`
  The raw data base contains some missing values, and it also contains numerical attributes. I removed the cases with missing values, and I discarded the numerical attributes to simplify your task, leaving 9 categorical attributes. Each of the 9 attributes is a one-letter symbol that is a shorthand for a more meaningful English-language description, but you needn't worry about the meaning of the attribute. I split the data set into training and testing components.You should retrieve the data from my site:
  `http://www.cs.colorado.edu/~mozer/courses/5622/crx.train`
  `http://www.cs.colorado.edu/~mozer/courses/5622/crx.test`

Part 1

For the titanic set, compute the statistic *p(death | class, age, gender)* for each combination of class, age, and gender. One way to display this statistic is to construct two tables, one for men and one for women. The rows of each table represent the different classes and the columns the different ages. In each cell of the table, insert the probability. Warning: Be alert to the possibility of a cell containing no data.

After you've built the probability table, make a second table that indicates classifications for each feature combination. If your estimate of *p(death | class, age, gender)* > .5, then label that cell as *death*; otherwise label that cell as *survival*.

Part 2

For the titanic set, build a Naive Bayes classifier. To build the classifier, you must first construct six one-dimensional tables: *p(class | death)*, *p(age | death)*, *p(gender | death)*, *p(class | survive)*, *p(age | survive)*, *p(gender | survive)*. To be clear on this notation, for *p(age |death)*, your table should have two rows, one for adult and one for child, and you should compute, for each age group, the probability of the deceased being in that age group. From these probability distributions, and the estimate of *p(death)* and *p(survival)*, build the same sort of table as you did in Part 1, indicating for each cell whether *death* or *survival* is more likely.

How well do the classification tables in Part 1 and Part 2 match?

Which table would you recommend using for prediction in case of another disaster like the Titanic (assuming it occurred at the same time in history)?

Part 3

Construct a Naive Bayes classifier based on the credit-application training data. Classify the data in the test set and report accuracy. In case *p(reject | application data) = p(accept | application data)*, choose the class that is most prevalent in the data.

Part 4

Construct a *k*-Nearest Neighbor classifier based on the credit-application training data. Classify the data in the test set for $k = 1, 3, 5, 11, 51$, and 101. I have chosen odd k so that there will be no ties (i.e., half of the neighbors are accepted applications and half are rejected applications). Make a graph showing, as a function of the smoothing parameter k, the proportion of errors on the test set.

You must decide on a distance metric. Describe the distance metric you chose. One simple approach is to assign a "0" if a feature of a training example matches a feature of a test case, or "1" otherwise, and simply sum up the number of feature mismatches. Using this distance metric, you may find that more than one training example is equidistant from the test case. You may select one at random for the classification.