Problem Set 2

D Keck

February 01, 2018

Objectives

- Understand the KNN method via complete implementation
- Understand how to build, test, and predict with a KNN model using library functions
- Understand how to build, test, and predict with a multinomial ANN model using library functions

Due February 19

Requirements

- Problem sets are done in groups of 3 (21 groups)
- Each group member must be skilled with the language chosen at the start of problem set
- Each student must do approximately $\frac{1}{3}$ of the development
- The deliverable is single PDF or HTML that meets the standard of a reproducible research report with professional quality communication. Deployable code will not be a deliverable for any problem set.
- All code must be displayed
- Identify and explain each answer. Don't just print a number.
- Numeric output should be easy to read e.g., not 10 decimal places
- Graphics must be easy to read i.e., titles, axis labels, legends, colors, etc.
- The rows and columns of tables and the columns of matrices and data frames must be labeled

Question 1

Import the iris data set from the UCI Data Repository

Display the data frame dimensions, the structure, summary, the first 5 and last 5 observations. Which are the explanatory and response variables? Comment on the data.

Question 2

Display several 3D scatterplots using 3 different explanatory variables in each plot and different viewing angles. Color code the three iris species in the scatter plots. Comment. Suggest using scatterplot3d(…, angle = … , color = …)

Question 3

Graphically display the correlation among the features (explanatory variables.) Comment. Suggest using corrplot() and pairs().

Question 4

Since the observations are grouped by species, randomize the observations for subsequent use. Suggest using order()

Scale each feature (column of **X**) so that each feature observation lies between 0 and 1.

Verify the scaling. Suggest using summary() or str()

Question 5

Create a test set using 130 observations and a test set with the other 20 observations.

Confirm by displaying the dimensions of each set.

Question 6

Write code from scratch to predict the species of each observation in the test set using KNN.

Experiment with the prediction accuracy by changing K, the number of neighbors. You might include mention of bias and variance.

Display your comparisons as well as accuracy and error rates. Include a confusion matrix as well as a line plot for acccuracy rate (vertical axis) v. K (horizonatal axis). Suggest that table(… , …) be included.

(Accuracy is determined by comparing your species predictions to the known species.)

Question 7

Now repeat the process from Question 6, but use the knn() function from the *class* package.

Question 8

Develop an ANN model for iris species prediction using a library call to a neural network modeler. Suggest using neuralnet( ).

Since the iris prediction has 3 response categories instead of 2 (its multinomial, not binomial), the species factor variable must be split into 3 binary variables. The method is called *one hot encoding*.

Suggest using class.ind( ) from the *nnet* package for *one hot encoding*, but this is also easily hand coded.

Make a formula using as.formula( ) for use with neuralnet( ) of the form:

y1 + y2 + y3 ~ x1 + x2 + x3 + x4

Plot the model network

Question 9

Predict the species for each observation in the test set. Display the comparisons between actual and predicted. Use the numeric, not binary, version of predicted results. Choose a meaningful display type.

Display the RMSE error between actual and predicted for each category.

Display the actual (numeric) and the activated (0 or 1) predicted values for categories. Suggest using ifelse() with a threshold of .5 which is 'half way' between 0 and 1.

Display the confusion table for each species prediction. Suggest using table(actual = y1test, predicted = …)

Display accuracy and error rates from the confusion tables. Suggest using table(), diag(), and sum(). Note that table( ) values are stored columnwise.

Question 10

Using your ANN model, experiment with the number of hidden nodes. What structure of layers and nodes produces the best accuracy on the test set relative to the number of layers, nodes, and model solution time? Carry out this experiment any way you like. Summarize your observations.

Wrap up

At this point you have all of the expected skills to do KNN and *vanilla* neural network machine learning! You should be proud!

The next methods will be K means clustering, Naive Bayes, decision trees, along with the necessary background topics like information theory, principal component analysis, and more on Bayes Rule. Applications will include image classification.

Then we'll do linear and logistic regression with background topics like gradient descent solvers, ANOVA, feature selection, cross validation, and ROC curves.

Finally we'll do the specializations. I'll present the final rubic shortly, but the main evaluation criterion is whether **your classmates learned something useful from your lectures, and less whether you learned something useful !**

Careful if you're not using a language that the class knows! Classmates will contribute to evaluations. Attendence will be monitored with N cards and will be reflected in *your* final grade.