Defence

Chris Meade

12/7/2016

I chose to analyze the Pima Indian Diabetes dataset available at the UCI Machine Learning Repository. This dataset has 768 observations of 9 variables. 8 of the variables give biological information about a given person, such as age, body mass index, and number of times pregnant, while the 9th variable is a class label that indicates whether or not a person has diabetes. The purpose of my project is to train a model using this data that can accurately diagnose diabetes in Pima Indian women. The three machine learning algorithms I compared were a random forest decision tree, a support vector machine with a radial bias function kernel, and a multilayer perceptron neural network.

After importing data into R, I began with my exploratory analysis. Although the dataset is not explicitly encoded with missing values, or NA's, several of the variables had values of 0 where a 0 would be biologically impossible, such as body mass index and blood glucose concentration. So I assumed that the dataset creator implicitly encoded missing values with 0. I corrected this by explicitly encoding these 0's with NAs.

I then calculated that almost half of all observation have at least one missing values. I didn't want to drop nearly half of the data from analysis, so I imputed the missing values using a method called imputation by predictive mean matching.

This method utilizes the subset of complete entries to compute a set of regression coefficients for each variable. Then the algorithm runs the regression on the missing data and generates a predicted value. Finally, the algorithm finds the most similar observed value to the predicted value and imputes that observed value.

With my preprocessing complete, I standardized the data so each column would have a mean of 0 and standard deviation of 1. Then I sampled 70% of my data for a training set and the remaining 30% into a test set, then began constructing models using the caret package for R.

For each of the three methods, I constructed models from the training set, using 10-fold-cross-validation to prevent overfitting. The caret package alows for easy performance optimaization by automatically constructing models with different parameter values. For each of the three methods, I then chose the final model with regards to its area under the ROC curve.

The random forest model has just a single parameter – the number of randomly chosen predictor variables. The model was repeatedly trained with 2, 5, and 8 randomly chosen variables and achieved a maximum Area under the ROC curve of .824 with two varialbes.

The support vector machine model has two parameters: sigma and C, where sigma represent the slack variable and C is a regularization parameter: a small C will yield a larger margin, while large C will give rise to a more narrow narrow margin. Similarly, sigma controlls the smoothness and flexibility of the decision boundry. After testing several different values of sigma and C, the model achieved a maximal ROC of .815 with sigma = .1 and C = .15.

Although we didn't discuss artificial neural networks in class, they seemed like an interesting concept to me, so I included one in my analysis. Put simply, this model takes input in a series of nodes which make up the input layer. It then feeds this informations to a given number of hidden layers where nodes will weight the inputs, and then outputs the information to an output layer. This model has one parameter – the number of hidden layers. The area under the ROC curve for this model was maximized at .828 with one hidden layer.

With three finals models, I chose the best one on its classifcation accuracy of the test set. The artifical neural network model was the winner, achieving an accuracy of nearly 80%