Machine Learning Class Project

Randolph Beck

July 9, 2017

Executive Summary

The purpose of this project is for the student to work through a machine learning exercise from a set of raw data through a final proof test on a small set of data.

The training data provided is from a real set of biometric measurements on test subjects doing barbell exercises. Accelerometers attached to the subjects' bodies and the barbells record the motions which are linked to a human-generated subjective rating of the quality of the motion. The student's goal is to create a machine learning system to issue a rating that closely matches the rating given by a human. Nearly 20,000 sets (rows) of human rated measurements are provided, so there is abundant data for training and testing.

The data was preprocessed by eliminating over half the columns which were sparsely populated or apparently irrelevant to the rating. Then 20% of the rows were set aside for cross-validation of the model, while 80% was used for initial training. Algorithm selection was done by trial and error: 11 models representing different types were trained on the training data and then tested on the validation data to determine which performed best.

For the final test–the 20 sets on the Quiz–the predictions of all algorithms were calculated and put in descending order by their expected out-of-sample error. The final answers on each test point were determined by consensus of the top 5 models. The result was 100% correct answers on the 20 test measurements.

Problem Solving Strategy

This course has given little guidance on the practical issues of selecting among the many possible models, so I googled "R machine learning model selection" and found an excellent website with a discussion of this issue here: http://machinelearningmastery.com/evaluate-machine-learning-algorithms-with-r/

This site asserts that the only way to do it, in the absence of past experience, is by trial and error and recommends a strategy of picking a few models of each type from the groups of 1) Linear Methods 2) Non-linear methods 3) Trees and Rules and 4) Ensembles of Trees. ## Data Selection An initial scan of the data showed that there were 100 columns that were sparsely populated, and had empty values in the 20 test cases, so these columns were eliminated from the training set. At this point, I carelessly left the first 6 columns of data in the training set. These columns included information like the test subjects names and timestamps, which could not be relevant to categorization.

I ran training and cross validation on the all the models with the full set of columns and got surprisingly good accuracies on the cross-validation data, leading me to believe that the models were quite good. When I ran the best models on the Quiz data and submitted the answers, I scored only 35%.

Next I eliminated the first 6 columns and re-ran the training and validation. Accuracies on the cross-validation data was actually lower for about half the models, but the best of the models all concurred on the Quiz test data. When the consensus of the best models was submitted on the Quiz, the result was 100% accuracy.

Algorithm Selection

Based on the recommendations of the article cited above, I selected the following algorithms for trials: Random Forests, Linear Discriminant Analysis, Support Vector Machines radial, k Nearest Neighbors, CART,

Stochastic Gradient Boosting, Generalized Linear Model, GLM net, Naive Bayes, C5.0, and Bagged CART–in all, 11 models.

As mentioned, I split the training data into 80% for training the model (15,699 observations) and 20% for cross validation (3,923 observations), and attempted to run all 11 models on the training data using the train function in the caret tool. All ran except GLM. Since several of the others had quite good results, I abandoned GLM rather than invest time in trying to diagnose it.

Testing and Cross-Validation of the Model

After training, I ran the predict function of caret on each model to produce a vector of 3,923 classifications for each model. By comparing them to the classe field I computed an Accuracy which gives me an estimate of the Out of Sample Error Rate. Initially, the the accuracies looked astonishingly good. Two of the models had 100% accuracy! On nearly 4,000 trials that's impressive.

However, when I ran the best models against the test data for the Quiz, I got all A classifications. When I submitted that answer on the Quiz, only 35% were correct. Back to the drawing board.

Next, I dropped the first 6 rows of the training data and re-ran all the models and got the accuracies in the table below:

Model	Accuracy	Prediction on Test Data
C5.0	.99873	BABAAEDBAABCBAEEABBB
Random Forests	.99669	BABAAEDBAABCBAEEABBB
Bagged CART	.99465	BABAAEDBAABCBAEEABBB
Stochastic Gradient Boosting	.98547	BABAAEDBAABCBAEEABBB
k Nearest Neighbors	.97476	BAAAAEDBAABCBAEEABBB
Support Vector Machines radial	.94723	BAAAAEDBAABCBAEEABBB
Naive Bayes	.76676	AAAAAEDCAAAABAEAABAB
Linear Discriminant Analysis	.71323	BABACEDDAADAEABAABBB
CART	.49605	CACAACCAAACCCACAAAAC

We're now down to 9 models from 11, as another one produced error messages. Notably, the top 4 performers on Accuracy concur completely on the test data for the Quiz.

Final Result and Key Learnings

The final model is a vote between the top 5 performing models above. This produced a 100% response on the Quiz test data.

These are my conclusions from this exercise:

- it is possible to build a machine learning system to produce remarkably accurate predictions on complex measurements with current off-the-shelf algorithms in R
- there is not a systematic way to pick the best algorithm to use–you should try some of each type, measure results and select some to refine
- try to pick the minimum number of data fields necessary to get good results; including irrelevant fields can have dramatic negative effects