Machine Learning

Adam F Clark

Executive Summary

In this paper we discuss one method of designing an algorithm to predict the manner in which a person wearing specific types of accelerometers is performing barbell lifts. The model-training data includes the accelerometer readings from six persons who performed the barbell lifts; each with the proper form and four improper forms. This paper shows how the prediction model was built, cross-validated and provides estimates of the errors.

1.0 Exploratory Analysis and Basic Data Summary

This data set contains 19,622 observations of 159 independent variables, though not all observations occur at the same temporal frequency. An examination of the data shows that there are many variables with very sparse data. For this simple homework project, these extremely sparse variables are ignored. However, it is understood that in practice, much more care should be exerted on discovering the meaning of these variables and the reason for their sparsity. There are likely more elegant methods for handling them than shown here.

Furthermore, it causes some consternation to not take the time variables into better account. Perhaps the accelerometer reading at any specific time is not as important as the full motion of the move over time (indicated by many rows in the data set). It seems worthwhile to create variables that try to capture these time-based trends. Since this algorithm will be tested on single-row records, this extension cannot be done here and is left as an exercise to the reader.

Furthermore, the fields that show when the actions took place were also removed. Their predictive capabilities may be very good on the original dataset, but would most certainly prove inadequate on future data sets.

```r
nonSparseList <- c(6:11,37:49,60:68,84:86,102,113:124,140,151:160)
pmlRawData <- read.csv("pml-training.csv")
pmlData <- pmlRawData[,nonSparseList]
pmlData[,2:54] <- pmlData[,2:54] * 1.0

pmlRawTest <- read.csv("pml-testing.csv")
pmlTest <- pmlRawTest[,nonSparseList]
pmlTest[,2:54] <- pmlTest[,2:54] * 1.0

set.seed(12)
inTest <- createDataPartition(y=pmlData$classe,p=0.3,list=FALSE)
myTest <- pmlData[inTest,]
myTrain <- pmlData[-inTest,]
```

Since the data set is sufficiently large, 30% of the data is randomly selected and set aside for a testing set, and thus 70% of the records are used for training.

2.0 Simple Random Forest Approach

The response variable is discrete (Class A, B, C, D or E) so it seems simpler to use a classification tree-style approach as opposed to generalized linear models. So to start, we will use a simple Random Forest approach. While the final model's worth will be predicted using the testing set (described above), a 3-fold cross-validation is done by the Random Forest algorithm to help train the model. More "folds" may provide a better model, however the smaller number was chosen for computation-time purposes.

```
myFit <- randomForest(myTrain[,-55],
                myTrain$classe,
                cv.fold=3,
                na.action = na.omit,
                imporance=TRUE,
                proximity=TRUE)

myFit
```

```
##
## Call:
##  randomForest(x = myTrain[, -55], y = myTrain$classe, proximity = TRUE,      cv.fold = 3, na.action =
##               Type of random forest: classification
##                     Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.27%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3905    1    0    0    0 0.0002560164
## B    3 2651    3    0    0 0.0022581859
## C    0   12 2382    1    0 0.0054279749
## D    0    0   13 2237    1 0.0062194580
## E    0    0    0    3 2521 0.0011885895
```

The resulting Random Forest model has an estimated out of bag error rate of less than 1% (0.27%). Even though we used the K-folds cross validation technique during model creation, this error rate is optimistic as it was done to help train the model. The confusion matrix shows slightly better prediction of a properly-performed barbell lift than it does for the common improperly-performed lifts.

```
myVarImp <- varImp(myFit, scale = FALSE)
myTop <- order(myVarImp,decreasing=TRUE)

myVarImp$varNames <- rownames(myVarImp)
myVarImp[myTop[1:5],]
```

```
##                   Overall         varNames
## num_window       934.9592       num_window
## roll_belt        777.6796        roll_belt
## yaw_belt         561.8313         yaw_belt
## pitch_forearm    519.7464    pitch_forearm
## magnet_dumbbell_z 481.5013 magnet_dumbbell_z
```

The most "important" variable, as calculated by the model, is "num window". I searched for some explanation of this variable on the GroupWare website, but never found a good description. Thus I can't provide any hypothesis as to why this variable may hold so much predictive power (and ultimately lowers my personal confidence in the model). The next most important variables are the "roll belt", "yaw belt", and "pitch forearm". These variables partially explain the gross movements of the belt and forearm. If this model proves to be useful, then one might assume the arm accelerometers are not needed to predict future barbell quality.

3.0 Cross Validation

```
myPred <- predict(myFit,myTest)
myTest$predRight <- myPred == myTest$classe
table(myPred,myTest$classe)
```

```
##
## myPred    A    B    C    D    E
##      A 1674    2    0    0    0
##      B    0 1134    4    0    0
##      C    0    4 1023    5    0
##      D    0    0    0  960    2
##      E    0    0    0    0 1081
```

The confusion matrix (above) looks favorable to this model being a good predictor. Of the 5,889 observations in the test set, 5,872 were classified correctly leaving 17 classified incorrectly. That is an error rate of 0.28% - which is on par with the previously estimated error rate.

Conclusion

A random forest model can be created (using 3-fold cross validation) that performs well at predicting the quality of performing a barbell lift. In a real prediction environment, I posit that the error could be further improved by considering the entire movement as a whole event, not just one accelerometer reading.