PML assignment on Human Activity Recognition

BK

Saturday, November 14, 2015

Objective:

To determine if a person is exercising in the correct manner/ with correct movements, by assessing readings from the accelerometers used. Being part of a course assignment, we are further required to,
* demonstrate our ability to extract meaningful information from raw data inputs,
* show how we've attempted to eliminate undesired noise in the data,
* attempt to reduce out-of-sample errors by using appropriate data slicing techniques (like Cross Validation) and
* describe how we've picked a modeling approach that minimizes out-of-sample errors

Extracting meaningful information from raw data:

Exploratory analysis shows that this is a good sized data set with 160 columns. Some of which are not predictors for the following reasons,
* user name or time stamps (7 such columns)
* columns with NA (67 such columns)
* columns with no or little variance (34 such columns)

After removing these, we're left with 53 columns of data, including the "classe" predicted variable that we will use to train/ test the model.

Eliminating undesired noise:

90% of the variance in the data set is explained by just 18 principal components. This reduction from 52 to 18 predictors shows substantial corelation between predictors that would have caused undesirable "noise". This transformation to a reduced number of weighted predictors was done using Principal Component Analysis.

Data slicing:

We know that we need to guard against overfitted models that perform well only on the training data because such a model would be useless for real world usage. To prevent this, two different approaches were tried. One was Cross Validation, with 10 folds and 10 repeats. But from a review of subject literature, we see that while Cross Validation shows less bias, it has more variance. So, a natural counter choice was bootstrapping with a 100 samples. We see from the same subject literature that bootstrapping has the opposite challenge of low variance but more bias.

Training a model:

First, we know that this is a classification problem where we're trying to identify categorical class labels. Hence, we need to build a classification model. In contrast, a prediction model that predicts a continuously valued function is not a good fit. Next, after reviewing course material, we pick two different classification options that show promise for further evaluation - Boosting and Random Forest.
Note: The evaluations below cover 2 data slicing schemes (Cross validation & Bootstrap) and two model types (Boosting and Random Forest). While we are aware that we should test all of these as 2x2 = 4 combinations, these have only been tested in two combinations Cross validation + Boosting and Bootstrap + Random Forest. This is so because the out-of-sample error rate of the second trial was a low 3% and its output was verified to be 100% correct for the results submission portion of this assignment.

Cross Validation + Boosting:

Model details, including the confusion matrix, are provided in R's output below and are not repeated here (like the Gradient Boosted Model package was chosen, it picked the best of 150 different iterations in which variables like the number of trees (150), interaction depth (3), shrinkage (0.1) , observations per node (10) were optimized). The data used here was put together using the repeatedcv function with 10 folds and 10 repeats.

```
## Aggregating results
## Selecting tuning parameters
## Fitting n.trees = 150, interaction.depth = 3, shrinkage = 0.1, n.minobsinnode = 10 on full training set
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 18 predictors of which 18 had non-zero influence.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1239  106   72   31   28
##          B   33  682   57   25   82
##          C   40  110  672   99   60
##          D   71   25   24  640   38
##          E   12   26   30    9  693
##
## Overall Statistics
##
##                Accuracy : 0.8006
##                  95% CI : (0.7891, 0.8117)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7473
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.8882   0.7187   0.7860   0.7960   0.7691
## Specificity            0.9325   0.9502   0.9237   0.9615   0.9808
## Pos Pred Value         0.8394   0.7759   0.6850   0.8020   0.9000
## Neg Pred Value         0.9545   0.9337   0.9534   0.9601   0.9497
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2527   0.1391   0.1370   0.1305   0.1413
## Detection Prevalence   0.3010   0.1792   0.2000   0.1627   0.1570
## Balanced Accuracy      0.9103   0.8344   0.8548   0.8787   0.8750
```

Bootstrapping + Random Forest

Again, all the relevant model details, like the number of trees (500) and the confusion matrix, are provided in R's output below. The data used here was the output of the boot data slicing function with 100 samples.

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, verbose = FALSE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 2.49%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4139   19   15   10    2  0.01099164
## B   45 2759   41    0    3  0.03125000
## C    8   43 2489   24    3  0.03038566
## D   10    2   90 2305    5  0.04436153
## E    2    8   18   18 2660  0.01699926
```

Picking a model:

The assignment requires us to make a model choice that minimizes out of sample error. Here is the comparison of the two models:
Out of sample error rate for Gradient Boosted model with Cross Validated samples is:19.94%
Out of sample error rate for Random Forest model with Bootstrapped samples is:2.98%

Clearly, the Random Forest model with Bootstrapped samples is the better approach. The output it produces for the test file of 20 observations is given below. These are 100% accurate, as verified during the project submission/ scoring.
B, A, B, A, A, E, D, B, A, A, B, C, B, A, E, E, A, B, B, B