Human Activity Recognition By Machine Learning

Quansheng

Wednesday, July 16, 2014

Abstract Four sensors were attached to arm, belt, forearm of the human body and excising equipment dumbbell and their movement during a dumbbell excising routine, Unilateral Dumbbell Biceps Curl, were recorded by sensors. The routine was doned in five diffirent ways and reported by the "classe" variable. Using the random forest machine learning algorithm, the fashions of the routine were accurately predicted by the sensor movement data, with accuracy of 99.34%.

Data Processing

- download human activity data. The data were splited into training and test datasets

```
        if (!file.exists("./PML")) dir.create ("./PML")
        myTrainUrl <-
"http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
        download.file(myTrainUrl,"./PML/pml-training.csv")
        myTestUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv"
        download.file(myTestUrl,"./PML/pml-testing.csv")
```

- read in training and testing csv files by read.csv.

```
        data<-read.csv("./PML/pml-training.csv")
        dataTest<-read.csv("./PML/pml-testing.csv")
```

Build a model with "classe" as an outcome and other variables as predictors

- Use sensor movement data attached to arm, forearm, belt and dumbbell as predcitors of how the routines are done, reported in the "classe" column. Irrelevant id variables and empty columns were removed and only columns containing those sensor data are used in our modeling process.

```
        library(caret)

## Loading required package: lattice
## Loading required package: ggplot2

        sensordata<-data[,c(8:11,37:49,60:68,84:86,102,113:124,140,151:160)]
```

- Split sensor data sets into training set and cross validation set. Training data were used to training models and crossValidation data were used to assess the out of sample error.

```
        inTrain<-createDataPartition(sensordata$classe,p=0.8,list=FALSE)
        training<-sensordata[inTrain,]
        crossValidation<-sensordata[-inTrain,]
```

- Fit training set data with random forest "rf" method. The data were preprocessed with knnImpute method to replace NA with near neigbor average.

```
        modelFit<-train(classe~.,data=training,preProcess="knnImpute",
                        method="rf",trControl=trainControl(method="oob"))
```

```
## Loading required package: randomForest
## randomForest 4.6-7
## Type rfNews() to see new features/changes/bug fixes.
```

- Predict the classe outcome of the cross validation data with the model built with "rf" method. The out of sample error was assessed with 99.34%.

```
        predictions<-predict(modelFit,crossValidation)
        confusionMatrix(predictions,crossValidation$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    3    0    0    0
##          B    0  755    1    0    1
##          C    0    1  681    6    2
##          D    0    0    2  635    0
##          E    0    0    0    2  718
##
## Overall Statistics
##
##                Accuracy : 0.995
##                  95% CI : (0.993, 0.997)
##     No Information Rate : 0.284
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.994
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             1.000    0.995    0.996    0.988    0.996
## Specificity             0.999    0.999    0.997    0.999    0.999
## Pos Pred Value          0.997    0.997    0.987    0.997    0.997
## Neg Pred Value          1.000    0.999    0.999    0.998    0.999
## Prevalence              0.284    0.193    0.174    0.164    0.184
## Detection Rate          0.284    0.192    0.174    0.162    0.183
## Detection Prevalence    0.285    0.193    0.176    0.162    0.184
## Balanced Accuracy       0.999    0.997    0.996    0.993    0.998
```

- Predict the classe outcome of the test data set.

```
        sensordataTest<-
dataTest[,c(8:11,37:49,60:68,84:86,102,113:124,140,151:160)]
        predictTest<-predict(modelFit,sensordataTest)
        predictTest

##  [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```