

Detecting the Correct Form of Bicep Curl Exercise

Chengde Wu

July 20, 2017

Overview

The quality of workout is as important as the quantity of it. A research team used sensors to record five different forms of bicep curl from six participants. This project is to build a model to predict whether a person is performing bicep curl correctly using the sensor data.

Loading the Libraries and the Data Set

The training data and the test data are loaded separately. A quarter of the training data is reserved for validation and the rest of the three quarters are used in training.

```
library(lattice)
library(ggplot2)
library(caret)
library(randomForest)

setwd("C:\\Users\\Chengde\\Desktop\\Course8")
train_validation <- read.csv("pml-training.csv")
inTrain = createDataPartition(train_validation$classe, p = 3/4)[[1]]
training = train_validation[ inTrain,]
validation = train_validation[-inTrain,]
testing <- read.csv("pml-testing.csv")
```

Cleaning the Data sets

Selecting appropriate predictors is essential to build an accurate model. In the data sets, the first seven features are not appropriate to be used as predictors. There are also some fields does not contain useful information such as "div#0" or empty character. These fields are removed from the data set. Furthermore, "NA"s are replaced by 0, which would not change the properties of the data in the context of this data set. "read.csv" command read in some of the features as factor type. This features are converted into numeric type.

```

#columns that does not have any info in the training set
no_div0 <- function(x)
{
  result <- TRUE
  if(is.factor(x))
    if(length(levels(x))<5)
      result <- FALSE

  result
}

no_info_idx <- sapply(training, no_div0)

clean_data <- function(df)
{
  #subsetting data
  new_df <- df[,no_info_idx][,-seq(6)]

  #replace NA with 0
  new_df[is.na(new_df)] <- 0

  #convert factors to numeric
  fact_idx <- sapply(new_df, is.factor)
  n <- length(fact_idx)
  new_df[fact_idx[1:n-1]] <- sapply(new_df[fact_idx[1:n-1]], as.numeric)
  new_df
}

new_train <- clean_data(training)
new_validation <- clean_data(validation)
new_test <- clean_data(testing)

```

Performing PCA to select only the useful predictors.

PCA is performed on the training data set three times by reserving 90%, 95%, and 99% of the variance respectively. Then the PCA parameters are used on the validation data set to reduce the dimensions of the data. "classe" field is added to the post-processed data.

```
#perform PCA on the data set, 90, 95, 99 variance
X <- new_train[, -ncol(new_train)]
preProc90 <- preProcess(X, method="pca", thresh = 0.9)
preProc95 <- preProcess(X, method="pca", thresh = 0.95)
preProc99 <- preProcess(X, method="pca", thresh = 0.99)

train90 <- predict(preProc90, X)
train95 <- predict(preProc95, X)
train99 <- predict(preProc99, X)
train90$classe <- new_train$classe
train95$classe <- new_train$classe
train99$classe <- new_train$classe

validation90 <- predict(preProc90, new_validation[, -ncol(new_validation)])
validation95 <- predict(preProc95, new_validation[, -ncol(new_validation)])
validation99 <- predict(preProc99, new_validation[, -ncol(new_validation)])
validation90$classe <- new_validation$classe
validation95$classe <- new_validation$classe
validation99$classe <- new_validation$classe
```

Training the models/ Model Selection through Validation

Three models are trained using the data sets that are processed with PCA. Validation process showed that the model which reserved 99% variance of the original data showed highest accuracy of 93%, validation error rate of 7%. Note: In this report, the corresponding six lines of codes are commented because they take very long time.

```
#model90 <- train(classe~., data = train90, method="rf", prox=TRUE)
#model95 <- train(classe~., data = train95, method="rf", prox=TRUE)
#model99 <- train(classe~., data = train99, method="rf", prox=TRUE)

#confusionMatrix(validation90$classe, predict(model90, validation90))
#confusionMatrix(validation95$classe, predict(model95, validation95))
#confusionMatrix(validation99$classe, predict(model99, validation99))
```

Prediction of the Test Set

The model correctly predicted 18/20 on the test set.

```
#test99 <- predict(preProc99, new_test)
#predict(model99, test99)
```