HumanExerciseRecognitionAlgorithm

Shouvik Das

Descriptiom: Human Activity Recognition - HAR - has emerged as a key research area in the last years and is gaining increasing attention by the pervasive computing research community especially for the development of context-aware systems. There are many potential applications for HAR, like: elderly monitoring, life log systems for monitoring energy expenditure and for supporting weight-loss programs, and digital assistants for weight lifting exercises. This human activity recognition research has traditionally focused on discriminating between different activities, i.e. to predict “which” activity was performed at a specific point in time (like with the Daily Living Activities dataset above). The approach we propose for the Weight Lifting Exercises dataset is to investigate “how (well)” an activity was performed by the wearer. The “how (well)” investigation has only received little attention so far, even though it potentially provides useful information for a large variety of applications,such as sports training.In this work (see the paper) we first define quality of execution and investigate three aspects that pertain to qualitative activity recognition: the problem of specifying correct execution, the automatic and robust detection of execution mistakes, and how to provide feedback on the quality of execution to the user. We tried out an on-body sensing approach (dataset here), but also an “ambient sensing approach” (by using Microsoft Kinect - dataset still unavailable) Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). THe WLE dataset is licensed under Creative Commons license (CC BY-SA).Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Install Libraries

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.1
```

```
library(ggplot2)
```

Doanloading training and test datasets

```
trainingRaw <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
testingRaw <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))
trainingCleanUp <- trainingRaw[,colSums(is.na(trainingRaw))==0]
testingCleanUp <- testingRaw[,colSums(is.na(testingRaw))==0]
```

Remove Columns that are unnecessary in prediction

```
trainingCleanUp<-trainingCleanUp[,-c(1:7)]
testing<-testingCleanUp[,-c(1:7)]
```

Find how many variables have high correlation

```
set.seed(23114)
M <- abs(cor(trainingCleanUp[,-53]))
##diag(M)
nrow(which(M>0.95,arr.ind = T))
```

```
## [1] 62
```

```
#Create Traning and Validation Sample
set.seed(23114)
inTrain<-createDataPartition(trainingCleanUp$classe, p=3/4, list=FALSE)
training<-trainingCleanUp[inTrain,]
validation<-trainingCleanUp[-inTrain,]
```

create the preProc object, excluding the response (classe)

```
set.seed(23114)

preProc  <- preProcess(training[,-53], method = "pca", pcaComp = 12, thresh=0.8)
```

Apply the processed object to the train and test data, and add the response

to the dataframes

```
training_pca <- predict(preProc, training[,-53])
training_pca$classe <- training$classe
```

train_pca has only 12 principal components plus classe

```
validation_pca <- predict(preProc, validation[,-53])
validation_pca$classe <- validation$classe
```

Valid_pca has only 12 principal components plus classe

Preparing models to compare accuracy

```
mod_rf <- train(classe ~ ., data = training_pca, method = "rf", verbose  = FALSE)
mod_gbm <- train(classe  ~ ., data = training_pca, method = "gbm", verbose  = FALS
E)
mod_lda <- train(classe  ~ ., data = training_pca, method = "lda", verbose  = FALS
E)
##mod_treebag <- train(classe  ~ ., data = training_pca, method = "treebag", verbo
se  = FALSE)

pred_rf <- predict(mod_rf, validation_pca)
pred_gbm <- predict(mod_gbm, validation_pca)
pred_lda <- predict(mod_lda, validation_pca)
##pred_treebag <- predict(mod_treebag, validation_pca)
```

Compare accuracies in different models

```
confusionMatrix(pred_rf, validation_pca$classe)$overall[1]
```

```
##  Accuracy
## 0.9661501
```

```
confusionMatrix(pred_gbm, validation_pca$classe)$overall[1]
```

```
##  Accuracy
## 0.7589723
```

```
confusionMatrix(pred_lda, validation_pca$classe)$overall[1]
```

```
##  Accuracy
## 0.4806281
```

```
##confusionMatrix(pred_treebag, validation_pca$classe)$overall[1]
```

Ensemble stacked model accuracy

Compare Accoracy, select final model and run model on test data

```
confusionMatrix(combPred, predValDF$classe)$overall[1]
```

```
##  Accuracy
## 0.9661501
```

```
test_pca <- predict(preProc, testing[,-53])
test_pca$problem_id <- testingCleanUp$problem_idtable
finalPred <- predict(mod_rf,test_pca)
```

Prediction result on test dataset

```
finalPred
```

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