Human Activity Recognition

Practical Machine Learning Project

Introduction

Using small sensors attached to the body, it is now possible to collect, relatively cheaply, large amounts of data on athletes performing a particular activity. Using the *Human Activity Recognition* data set (see http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har)), which contains data that has been collected while subjects were performing weightlifting with a dumbbell, the goal of this porject is to apply machine learning in order to assess the correctness of each repetition, or detect a mistake in the execution of the activity.

The data comes from accelerometers placed on the belt, forearm, arm, and dumbbell of 6 participants. The participants were asked to perform dumbbell lifts correctly and incorrectly in 5 different ways, under the supervision of an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate. In this experiment, the six allowed possible ways of lifting the dumbbell are:

- 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)
- throwing the hips to the front (Class E).

The data was recorded using four 9 degrees of freedom Razor inertial measurement units (IMU), which provide three-axes acceleration, gyroscope and magnetometer data at a joint sampling rate of 45 Hz. For the Euler angles of each of the four sensors the following eight features were calculated: mean, variance, standard deviation, max, min, amplitude, kurtosis and skewness, generating in total 96 derived feature sets.

Using machine learning and pattern recognition techniques, the goal is to detect the correctness or mistakes in the execution of lifting the dumbbell. The application in the real world of such a model would allow to give real-time feedback to the athletes (qualitative activity recognition) and to detect possible mistakes in order to avoid injuries.

Cleaning the data set

The data set contains 19622 observations of 160 variables. The data is first splited between a training and a cross-validation set (with respectively 60% and 40% of the observations). Many variables contain only non numbers (“NA”), and therefore the data set needs to be tidied first. Variables containing more than 95% of “NA’s” are not included in the tidy data set. Furthermore, variables that are irrelevant for the machine learning algorithm are left out as well:

```r
data <- read.csv("pml-training.csv", na.strings="")
library("caret")

# create training and cross-validation sets:
inTrain <- createDataPartition(y=data$classe, p=0.6, list=FALSE)
training <- data[inTrain,]
cross_validation <- data[-inTrain,]

### Data Cleaning ###
training_tidy <- training
for(i in 1:length(training_tidy[1,])){
  column_i <- training_tidy[,i]
  column_i[(as.character(column_i)=="NA")] <-NA  # some NAs have wrong format
  training_tidy[,i] <- column_i
}
training_tidy[training_tidy=="#DIV/0!"] <- NA # DIV/0 is also NAs

inTidy <- c()
for(i in 1:length(training_tidy[1,])){
    if(  sum(is.na(training_tidy[,i])) > (0.95*length(training_tidy[,1])) ){ # remove columns that have 95% NAs
          inTidy[i]=0;
    }else{inTidy[i]=1;}
}
training_tidy <- training_tidy[,inTidy==1] # tidy data, contains now 60 variables

# remove variables that are not useful for the model, leaving 53 variables (including the outcome: "classe")
training_tidy <- subset(training_tidy, select = -X)
training_tidy <- subset(training_tidy, select = -user_name)
training_tidy <- subset(training_tidy, select = -raw_timestamp_part_1)
training_tidy <- subset(training_tidy, select = -raw_timestamp_part_2)
training_tidy <- subset(training_tidy, select = -cvtd_timestamp)
training_tidy <- subset(training_tidy, select = -new_window)
training_tidy <- subset(training_tidy, select = -num_window)
```

Machine Learning

The training data set is now tidy and contains 53 variables, including the outcome “classe” (corresponding to the 6 possible classes of execution of the weight lifting as described in the introduction). We can now use a machine learning algorithm to build a model that take the values of the 52 variables and link them with one of the 6 possible classes. Because of the characteristic noise in the sensor data, a Random Forest approach is used (like in the original paper, see Section Reference):

```r
# Here we use the randomForest function from the randomForest package:
set.seed(32333)
library("randomForest")
model.rf <- randomForest(classe ~ ., data=training_tidy)
```

Cross-validation

The random forest model can now be applied to the cross-validation part of the data set, in order to evaluate the accuracy of the model. **The accuracy of the model is 99.38%. The confusion matrix indicates that most of the classes are correctly predicted and that the expected out of sample error is 0.611%** (49 wrongly classified cases out of 7846).

```
prediction <- predict(model.rf,cross_validation)
confusionMatrix(prediction,cross_validation$classe)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    A    B    C    D    E
##          A 2228    3    0    0    0
##          B    3 1511    8    0    0
##          C    0    4 1360   17    3
##          D    0    0    0 1269   10
##          E    1    0    0    0 1429
## 
## Overall Statistics
## 
##                Accuracy : 0.9938
##                  95% CI : (0.9918, 0.9954)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.9921
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9954   0.9942   0.9868   0.9910
## Specificity            0.9995   0.9983   0.9963   0.9985   0.9998
## Pos Pred Value         0.9987   0.9928   0.9827   0.9922   0.9993
## Neg Pred Value         0.9993   0.9989   0.9988   0.9974   0.9980
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2840   0.1926   0.1733   0.1617   0.1821
## Detection Prevalence   0.2843   0.1940   0.1764   0.1630   0.1823
## Balanced Accuracy      0.9988   0.9968   0.9952   0.9926   0.9954
```

Conclusion

The *Human Activity Recognition* data set has been used to build a machine learning algorithm that allow to predict the quality of the execution of lifting a weight. The data set has been split between a training and a cross-validation set. The training set has been tidied, and the relevant variables have been selected to build a random forest algorithm. **The predictive accuracy on the cross-validation set is 99.38% and correctly predicts all of the 20 test cases** (which is higher than the recognition performance of 98.03% reported in the original study).

Such an algorithm could be used in real life to give real time feedbacks on athletes, and detect mistakes by classification. However the original study (see section reference) underlines that this approach would hardly be scalable. Indeed, it would be infeasible to record all possible mistakes (all possible classification of mistakes) for each exercise. In the original study, an second approach by building a model is proposed (i.e. not using machine learning).

Reference

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.