Practical Machine Learning Project

Kea Duckenfield

April 18, 2018

# Predicting the Manner of Lifting a Dumbbell

## Executive Summary

The assignment was to build a model to predict the manner in which a dumbbell was lifted using the Weight Lifting Exercise Dataset.

## Introduction

The dataset contains 19,622 observations of six people (young, healthy men) lifting a dumbbell. Five ways to lift the dumbbell were identified: one correct (A), and four incorrect (B-E). Three types of sensors (accelerometer, gyroscope, and magnetometer) were attached to four location: belt, arm, forearm or glove, and dumbbell. These yielded measurements in three spatial dimensions at a frequency of 45 Hertz, and the raw measurements and derived quantities were compiled with subject identity, time of measurement, and outcome to yield 159 variables in the dataset.

## Data Cleaning and Exploratory Analysis

The data were provided as .csv files. I read these into R replacing missing values with "NA." I also loaded the caret library.

library(caret)  
## Read data into R, while filling empty observations with "NA."  
train <- read.csv(file = "C:/Users/kea/Documents/coursera/ml/wk4/project/pml-training.csv", na.strings = c("", "NA"), sep = ",")  
test <- read.csv(file = "C:/Users/kea/Documents/coursera/ml/wk4/project/pml-testing.csv", na.strings = c("", "NA"), sep = ",")

I began tidying the dataset by removing 43 variables with near-zero variability.

## Remove variables with near-zero variability  
nzv <- nearZeroVar(train, names = TRUE)  
all\_cols <- names(train)  
train\_var <- train[, setdiff(all\_cols, nzv)]

Roughly 90 per cent of about 60 variables were missing (either empty or NA). I removed those variables. This left 58 predictors.

## Remove variables with mostly missing observations  
train\_nona <- train\_var[, !sapply(train\_var, function(x) any(is.na(x)))]  
##

## Model Building and Cross-Validation

Since the variables I would have focused on, based on reviewing the literature and thinking about the problem, had turned out to be unpopulated, I was forced to work empirically (mechanistically blind). In the end I chose to use a random forest model since prior research indicated its suitability, and I wanted to include categorical variables as is.

I partitioned the data 70% training and 30% testing. I built the model using the training data.

Even though the mere act of using partitioned training and testing data subsets in modeling is described as de facto cross-validation (see e.g. Geisser 1993); more than one Stack Overflow commenter asserted that there is no need for cross-validation as a guard against overfitting in random forest modeling, because it is estimated internally; and Zach Deane-Mayer (coauthor of the caret package) says in a DataCamp tutorial that the default train() resampling method, bootstrapping, gives comparable results, I reckoned I'd get dinged on this project if I didn't take the hint and include it anyway, so I set the trainContol method to be cross-validation.

## Split data into training and test sets  
set.seed(11)  
inTrain <- createDataPartition(y = train\_nona$classe, p = 0.7, list = FALSE)  
training <- train\_nona[inTrain,]  
testing <- train\_nona[-inTrain,]  
## Use training subset to train a random forest model, using ten-fold cross-validation  
modelFit <- train(classe ~ ., data = training, method = "rf", trControl = trainControl(method = "cv", number = 10, verboseIter = TRUE))

## Model Evaluation

Compared with using bootstrapping, cross-validation did seem to increase model accuracy very slightly if not non-negligibly. The caret algorithm chose mtry = 41, i.e. 41 randomly selected predictors at each cut in the tree. This accuracy was 0.9997816 (compared with 0.9996986 using bootstrapping). However, the 'proof' of how cross-validation helped improve model performance is in the 'pudding' of evaluating the change in error rate.

I ran the model on the testing data and checked the rate of successful prediction.

## Use predict() to run the model on the testing subset:  
pred <- predict(modelFit, testing)  
## Determine the rate of correct predictions:  
testing$predRight <- pred == testing$classe  
sum(testing$predRight == TRUE)/sum(testing$predRight)  
## Generate a table comparing predictions and outcomes:  
table(pred, testing$classe)

Both models were completely successful in predicting values in the testing subset; the out-of-sample error rate for both was 0.

I also wanted to know which variables had turned out to be important in prediction. I used varImp() to look at this information.

varImp(modelFit)

In both cases, roll\_belt dominated (7.3122/100), contributing more than three times the influence of the next most important variable. For the cross-validation model, the second most important variable was raw\_timestamp\_part\_1 (1.9272/100). Almost as important was pitch\_forearm (1.8917/100), and then accel\_belt\_z (1.5183), followed by roll\_dumbbell (0.9263) and num\_window (0.9038). The rest of the variables contributed 0.66 or less. The main difference in the bootstrap model was that pitch\_forearm was more important (2.1993; second most important).

## Discussion

I was happy with the accuracy and predictive power of my model (both versions). I was pleased to see that roll\_belt was the most important variable, because this is consistent with Velloso et al.'s previous study of the data, which identified mean and sd belt roll as two of the 17 most important variables. I had planned to consider forearm pitch as a candidate key variable, and its appearing in the top three most important variables confirmed my idea. I hadn't identified dumbbell roll as a key variable, so that was a surprise.

The appearance of two variables not intended to measure component mechanics of weightlifting (raw\_timestamp\_part\_1 and num\_window) in the top five most important variables in the model worries me. This suggests to me that this model is too empirical to be used to understand weightlifting mechanics generally. While this model works for this dataset, it may not work for other datasets (because they may not include the same types of ancillary variables).

## Conclusion

I wonder whether this dataset should really have been treated as a time series in order to better capture the mechanisms driving the process being studied. Nevertheless, the fact that it is dominated by at least one variable related to the category of variables describing belt roll, identified in previous work as key variables, is in its favor as possibly capturing not only empirical behavior but maybe also some mechanistic information about weightlifting.