Practical Machine Learning Project

Doug Evans

July 25, 2015

Project Overview

For this project, we are asked to analyze data provided by:

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. For more information about this data, it can be found at: http://groupware.les.inf.puc-rio.br/har#ixzz3gO4tuBJf (http://groupware.les.inf.puc-rio.br/har#ixzz3gO4tuBJf)

According the project directions, "The goal of your project is to predict the manner in which they did the exercise".

Overall Approach

From the data, paper, and data provided through the resources provide above, we know that 6 participants were fitted with 4 motion sensing devices (arm-band, glove, belt, and dumbbell) and asked to perform Unilateral Dumbbell biceps curls in five different fashions: exactly according to a 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). Class A is ostensibly the correct way to perform the curl, while classes B - E are intentional incorrect ways to perform the curl (this was instruction given to each participant).

We have been instructed to predict, from the sensor data provided, the manner in which they did the exercise (class A through E). This is a fairly broad requirement, and could result in months of work and potentially thousands of hours of research and computer time, which is not practical in this setting. Instead, my focus here is simple, and that is to try and answer is these two points: 1) To what degree (and accuracy) can simple machine learning methods be used to predict the manner in which exercises were performed?, and: 2) What restrictions must be applied when using the resulting model(s) to predict the manner in which the exercises are performed?

Data Exploration

I reviewed the data structure using both R and Microsoft Excel. The data has 159 features and the outcome (classe), and is not tidy in one regard. There are two types of observations, identified by the variable new_window (it is either yes or no). Many columns (features) have no data except for the cases with the feature new_window = 'yes' (approx 400 of a total of 19622 observations). The data is explained in the documents from the data curators above, but appears to be either motion sensor data measurements (new_window = 'no'), summary measurements (new_window = 'yes'), and data acquisition information (time

stamp). This mixture of data types raises a question as to whether both data types should be in the training modeled. I decided to do most all modeling without the window_new = 'yes' data. This is a small fraction of the observations, about 2%. Also, I noted that the project test data (20 observations) are all with the window_new feature = 'no', which reinforces removal of the 'yes' observations from the modeling.

Because the data has time stamp information, this potentially makes prediction within the dataset very effective (which I confirm), as each participant and activity is performed at unique times. It does not seem appropriate to use time stamps in the prediction algorithm, as it is not a measured parameter of the sensors. But I have an instance where I model it (model 1) just to show its effect. The majority of the modeling reported in this paper (models 2 through 5) us performed without the time stamp feature included as a predictive feature.

Finally A few comments are in order regarding predictions that might be possible from this dataset, and from the broadness of 'question' we are given to answer. We know that this is time series data, which means it is sequential, and within each sequence would be a curve representing the motion of a curl exercise. This would be very difficult to model and is not even contemplated in this paper or modeling. Also, it could have a desirable goal to predict 'classe' based on measurements for individuals that are not part of the data set. I'll explore this a little in one model, mostly to show that this would be vary difficult, and is far too sophisticated a question to try and answer in the context and scope of this paper.

NOTE: I've left most of the code out of this paper (echo = off the the Rmd file) so I have room for the 2000 word imposed limited (which I exceed, my apologies). Please review the Rmd file included in the repository if you would like to see the code (less than 100 lines of code, and its very simple code).

Data Subsetting

I create several data subsets which will be used for multiple purposes. The main subsetting is to subset out a training set, which will be 70% of the original data, along with validation and test subsets each being 15% of the original data (70% + 15% + 15% of 19622 observations).

I further subset the training set by subsetting out observations with feature window_new = 'no' and 'yes'. Of the 13,737 observations in the training set, 280 are subsetted out due to the window_new = 'yes' feature. I create one remaining experimental subset, removing one participant's data (Eurico). The training data (without Eurico) is a special case and will only be used to demonstrate the limited predictive power of the modeling techniques discussed in this paper (model 5).

Feature Selection

After playing around quite a bit, I decided on three sets of features to model. The decision of which features to select for modeling is based entirely on practical issues, specifically my time and computer modeling time. I show the code for these feature sets here.

```
# Set up the feature and label columns for three sets of covariates
modfeat1 <- c(3, 160) # 160 is the column containing the result classe
modfeat2 <- c(6:11, 160)
modfeat3 <- c(6, 8:11, 160)
```

The feature in modfeat1 is just the time stamp (column 3). The features in modfeat2 are related to the belt sensor, though column 7, num_window, appears to be derived somehow, and I strongly suspect is temporal nature, similar to the time stamp. Finally, in modfeat3, I have just the one subset of belt sensors, but I've removed the num_window feature. You may note, missing from this list are the vast majority of sensor measurements. I am omitting these because, when I did model them, they did not substantially change the outcome of the model, and particularly, they did not alter the points I will be making about the predictive quality of the data and the resulting models.

Model Technic and Selection

I selected 5 models to experiment with, and the intent is that each increment in model, from 1 through 5, I can demonstrate some predictive aspect of the model, modeling techniques, and data. The first model, mod1, uses a decision tree classification technique (rpart), which is fast to compute. In model I only model the time stamps, as shown in modfeat1 earlier in this paper. The second model (mod2), also uses rpart, but this time I switch to non-time stamp features (features 6 through 11). This gives a good side-by-side comparison of just the time stamp feature to actual measurement features. The next model (mod3) is like mod2, but this time I change the modeling method from rpart to random forest (allows another side-by-side comparison, this time to see the superior performance of random forest). The next model (mod4) is like mod3, also using random forest, but this time I remove feature 7 which is num_window (modfeat3), which I suspect is temporal in nature, like the time stamp feature (column 3 in modfeat1). The last model I train uses the same features as mod4, but this time the training data set is the subset with the participant Eurico removed. This will be used to demonstrate that predictions are not practical between participants using the techniques of this paper.

These choices should allow for a fairly nice side-by-side comparison, which I discuss in the test results and conclusion below. Here is the code used for generating these models.

```
mod1 <- train(classe ~ .,data=trtrn[,modfeat1], method="rpart")
mod2 <- train(classe ~ .,data=trtrn[,modfeat2], method="rpart")
mod3 <- train(classe ~ .,data=trtrn[,modfeat2], method="rf")
mod4 <- train(classe ~ .,data=trtrn[,modfeat3], method="rf")
mod5 <- train(classe ~ .,data=tryrynoEurico[,modfeat3], method="rf")
```

Test Results

There is a lot to cover here, and I'll just cover the highlights. All evaluations are done using prediction accuracy (fraction of correct predictions divided by total predictions).

The results below are organized into three groupings, showing the performance of the various models and datasets. Generally, the first group is the training data results, the second group is the validation results, and the last group is test data.

For the Training group (9 computations), I predict and compute the accuracy for all 5 models. Except for model 5, I predict and compute the accuracy of both the window_new = ‘yes’ and ‘no’ subsets, though the former is a very small dataset (280 observations). Note that the window_new = ‘yes’ dataset is not really a training set, but I wanted a place to demonstrate that the predictive results are reasonably good with it. Actually, this small subset could be considered a validation subset, and the way I am using it here is as a validation set, in this case justifying removing these observations from the training data model.

Note that, in the predictions for model 5, I use the same subsetted training set (without Eurico) against the a model created with that same training set (the one without Enrico), in this case I only have data with new_window = ‘no’, which is a minor point.

The Validation group (5 computations), follows the same logic and ordering as the training set predictions, except now there is only one dataset to predict against (I did not split the validation dataset on the window_new feature). In this case, Model 5 is used to predict against a data set that only contains Eurico. This is an attempt to use a model created with all participants but one to predict how the excluded participant is performing (not a good predictor).

For the Test group (5 computations), these follow the validation group ordering and logic, with the exception of model 5. Model 5 was trained from data without Eurico, but now used to predict test set which contains all participants, including Eurico.

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:Hmisc':
##
##     combine
```

```
##    [,1]
## pf "TRANING DATA RESULTS, MODELS 1 - 5, AND TWO SUBSETS"
##    "Training Data, Model 1, new_window = no, accuracy = 0.993"
##    "Training Data, Model 1, new_window = yes, accuracy = 1"
##    "Training Data, Model 2, new_window = no, accuracy = 0.409"
##    "Training Data, Model 2, new_window = yes, accuracy = 0.418"
##    "Training Data, Model 3, new_window = no, accuracy = 1"
##    "Training Data, Model 3, new_window = yes, accuracy = 1"
##    "Training Data, Model 4, new_window = no, accuracy = 0.992"
##    "Training Data, Model 4, new_window = yes, accuracy = 0.9"
##    "No Eurico Data, Model 5, new_window =  no, accuracy = 0.99"
##    " "
##    "VALIDATION DATA RESULTS, MODELS 1 - 5"
##    "Validation Data, Model 1, accuracy = 0.994"
##    "Validation Data, Model 2, accuracy = 0.406"
##    "Validation Data, Model 3, accuracy = 0.999"
##    "Validation Data, Model 4, accuracy = 0.89"
##    "Eurico only Data, Model 5, new_window =  no, accuracy = 0.172"
##    " "
##    "TEST DATA RESULTS, MODELS 1 - 5"
##    "Test Data, Model 1, accuracy = 0.992"
##    "Test Data, Model 2, accuracy = 0.414"
##    "Test Data, Model 3, accuracy = 0.999"
##    "Test Data, Model 4, accuracy = 0.898"
##    "No Eurico Data Model 5, accuracy = 0.785"
```

Conclusion

There are some interesting results. First, Model 1 gave excellent results for all cases (training, validation, and test), and the model is extremely simple and fast to create. This theoretically satisfies the project direction, but I think it is a poor predictor because the time stamps is, as stated earlier, not really part of measurement data that is in the focus of the study.

Model 2, which switches to other more practical predictive features (modfeat2), shows much poorer accuracy, but does show similarly consistent accuracy with all three data sets (training, validation, and test). Results drop from the 0.993 down to 0.41 (from model 1 to model 2). The fact that all three data sets show similar accuracy is a sign that the model was not over fit (due, I believe, to the simplicity of the rpart method).

Model 3 is very interesting. Switching methods from rpart to random forest gave major improvements in accuracy. The accuracy for all three datasets is actually better than with model 1 accuracies with just the time stamp using the rpart method. In this case, all three datasets perform well. We will see in the next paragraph that features 7 is likely to be a key to this good performance of model 3.

Model 4, which removes the one feature (column 7, num_window) from model 3, while still using random forest, demonstrates that the excluded column did in fact have temporal information in it , and by removing it performance went down. Even though the training set shows excellent accuracy, both the validation and test

sets both do much worse (compare accuracy reductions from .99+ down to .90). This difference between training, validation, and test sets shows the effects of over fitting, especially with the lost of the temporal features in columns 3 and 7 removed.

Finally, Model 5 does something quite different. Here, we use the same good features selection as in Model 4 (modfeat3) and method (random forest), but we build the model without one of the participants (Eurico). For the accuracy computation in the training group (above), the prediction is performed using same subsetted training set (against Model 5), qne looks as expected and performed very well (accuracy is .99), which is similar to model 4. But I then show model 5 in the validation results, making predictions against a data set that only contains Eurico. There you will see the results are extremely poor. A random pick of classe would give 20% accuracy (chance of 1/5), and we only are getting 17.3%, so we'd do better guessing than using the model.

In the test set area, I test model 5 using the test data set, which has Eurico data included. In it, you can see that we have deteriorated accuracy from model 4, and this is because we now know that almost all of the predictions performed for Eurico are wrong (from the validation set testing). This explains the drop from 90% to now only 79% (model 4 to model 5).

In summary, I feel that model 4 is the correct predictive model to use, but only for data collected from these participants. None of these models, as demonstrated by model 5, can be used to assess similarly instrumented participants not in the training data. Under these conditions, an out-of-sample accuracy of 90%, as demonstrated in both the validation and test sets, could be expected.

I finish by showing the significant features (gini) for both model 3 and model 4, primarily to demonstrate that removal of columns 7 explains the change in performance for these two models. Notice that model 3 has the feature num_window at 100% relative to the next feature roll_belt (at 30%), a much lower significance. This accounts for the lower accuracy of model 4.

```
varImp(mod3)
```

```
## rf variable importance
##
##                   Overall
## num_window       100.000
## roll_belt         28.950
## yaw_belt          24.905
## pitch_belt        21.060
## total_accel_belt   6.834
## new_windowyes      0.000
```

```
varImp(mod4)
```

```
## rf variable importance
##
##                   Overall
## roll_belt          100.00
## yaw_belt            89.78
## pitch_belt          89.33
## total_accel_belt    21.93
## new_windowyes        0.00
```