Activity Recognition Analysis

Mary Anne Plano
January 20, 2014

Introduction

The Human Activity Recognition database was built by recording measurements provided by waist-mounted smartphones with inertial sensors while subjects performed daily activities. [1,2] The task of this analysis is to build a function that predicts what activity a subject is performing based on the quantitative measurements from the Samsung smartphone.

Methods

Data Collection

For this analysis we used a sample of 7352 observations of 563 variables provided at the course website. [3]The data were slightly pre-processed to ease loading into R, the raw data can be found at an archival website provided in the references. [4] The data were downloaded from the provided website using the R programming language.[5] All analyses were performed using the R programming language and a script illustrating the commands utilized can be found online.[6]

Exploratory Analysis

An initial survey of the data revealed the need to clean up the column names of the data as they contained unsuitable characters. Also, the column names differed for different data subsets. The column names were regularized and made unique to facilitate analysis and interpretation. There was no missing data and all numeric features were found to have expected values of between -1.0 and 1.0. The data column representing which activity the subject was engaged in was a character class and was changed to factor class for ease of analysis. All other data were provided as numeric.

The provided data set was divided into two subsets – one for training which was based on 4 subjects and contained 1315 rows of data and one for testing which was based on 4 different subjects and contained 1485 rows of data. All exploration was done on the training set and the test set was only used at the end to evaluate miscalculation errors.

Initial exploration involved creating several scatter plots of various features and color-coding the data points based on activity to determine if there were any obvious features that could be used to distinguish activities. Initially plots were made of various features which, based on common sense might differentiate, such as angle-based features that could distinguish laying from sitting for example or looking at acceleration in the z direction to separate out the walking up or walking down activity. Also, the clustering techniques of K-means [7,8]was explored as a way of finding the key features to utilize for determining the associated activity. Eventually classification trees were found to be the best and simplest method for exploring features and modeling the Samsung data.[9,10]

Statistical Modeling

To create a prediction function for the Samsung data a classification tree model was utilized. Initially all features of the data were included and the algorithm selected the ten best features. One step of pruning was performed to create the best model with 6 leaves – the number of activities to be classified.

Activity Recognition Analysis

Mary Anne Plano
January 20, 2014

Analysis

A classification tree was built on the Samsung data allowing the algorithm to select from all 561 features provided. The ten features selected are listed in table 1 below.

Table 1. Ten features selected in the initial classification tree.

Feature #	R variable name	Feature name
1	"tBodyAcc.max...X"	Max body accel X
2	"tGravityAcc.mean...X"	Mean gravity accel X
3	"tGravityAcc.max...Y"	Max gravity accel Y
4	"tGravityAcc.mean...Y"	Mean gravity accel Y
5	"tGravityAcc.mean...Z"	Mean gravity accel Z
6	"tGravityAcc.min...Y"	Min gravity accel Y
7	"fBodyAccJerk.bandsEnergy...17.32"	Bands energy body accel jerk (FFT)
8	"fBodyAccMag.std.."	Body accel mag (FFT)
9	"fBodyAcc.bandsEnergy...17.24.1"	Bands energy body accel (FFT)
10	"tBodyAccJerk.max...X"	Max body accel jerk X

The initial classification tree had 12 terminal nodes and had a misclassification rate of 2.4% on the training data. Table 2 shows the relationship between the actual activities and the ones classified so it would be possible to diagnose which activities had the most misclassification issues. From Table 2 we can see that errors of misclassification are between sitting and standing and that there are issues discerning the difference amongst walking on a flat surface versus walking up or down. Also reported in the table are the precision and recall values for each activity. Precision is the percentage of the prediction for a particular activity that is true - e.g. for the “Walk” activity 265 were predicated to be “Walk”, but only 258 of those were correct giving a precision of 97.4%. Recall is the fraction of the activity that we predicted which is true – e.g. for “Sitting” there are actually 198 true cases of “sitting” and we predicted 188 of them or 94.9% recall.

Table 2 Confusion Matrix of actual activity (vertical) versus classified activity by initial tree based on training data.

		Predicted values						
		Laying	Sitting	Standing`	Walk	Walk-down	Walk-up	**Recall**
Actual values	Laying	221	0	0	0	0	0	**100%**
	Sitting	0	188	10	0	0	0	**94.9%**
	Standing	0	0	227	0	0	0	**100%**
	Walk	0	0	0	258	4	4	**98.1%**
	Walk-down	0	0	0	5	186	2	**96.4%**
	Walk-up	0	0	0	2	4	204	**97.1%**
	Precision	**100%**	**100%**	**95.8%**	**97.4%**	**97.4%**	**97.1%**	**97.6%**

In order to improve the given model boxplots were created of all 10 features listed in table 1 versus the activity to discover possible transformations of these features that might improve the classification process in the sitting versus standing and in the walking activities. Several transformations were tested including multiplying feature values together and also squaring the values or cubing the values. However, none of transformations tested were seen to improve the initial model. So, the next step was to use a 10-fold cross-validation to determine the optimal number of leaves to prune the original tree. [11] A plot of both the misclassification error and deviance are shown in figure 1. As can be seen, both measures indicate that using a tree of size 6 would be optimal.

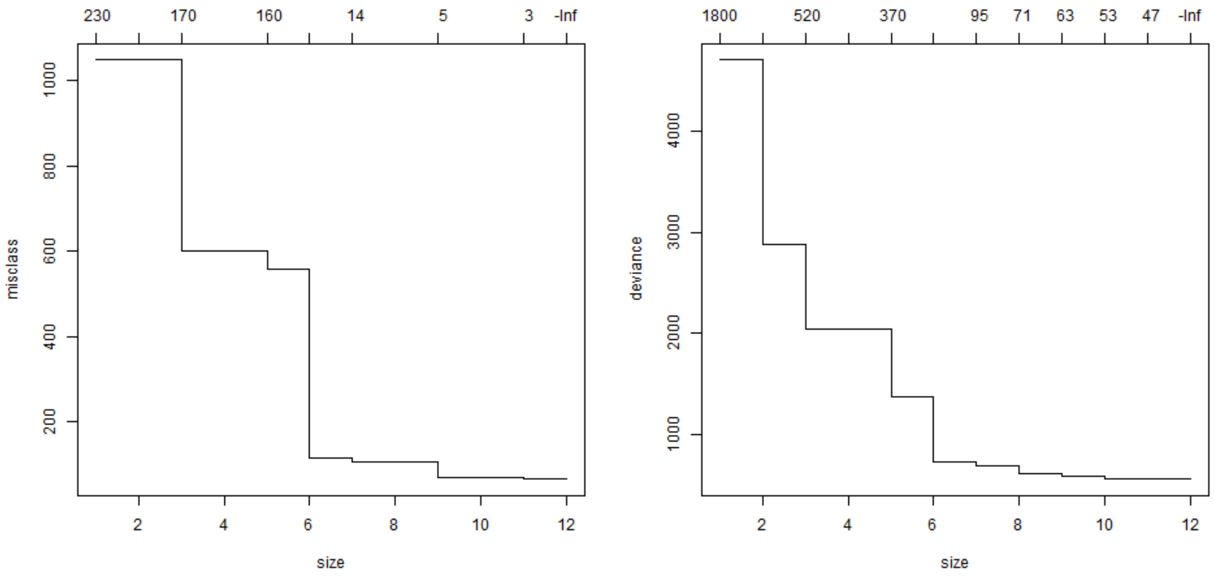

Figure 1 – Plots of the error versus tree size based on cross-validation of the original model. On the left is graphed the misclassification error versus size of the tree and on the right is the deviance or model impurity versus tree size. Both show a significant drop at size 6, so that was selected for pruning.

Next I pruned the model to the best 6 leaves and obtained the classification tree shown in Figure 2. The pruned model utilized only the 5 features labeled as Features 1,2,3,6 and 8 in Table 1. The pruned model was applied to the training set to obtain the re-substitution misclassification error and it was found to be 7.2%. An increase in the misclassification error of the simpler, pruned model is to be expected, but the simpler model should have a better error rate when applied to the test set than the original full tree model.

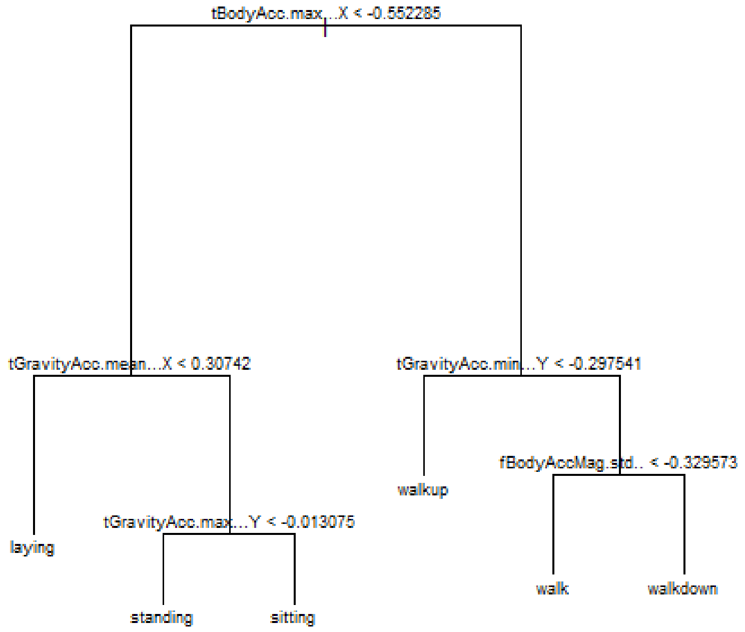
tBodyAcc.max...X < -0.552285
tGravityAcc.mean...X < 0.30742
tGravityAcc.min...Y < -0.297541
walkup
fBodyAccMag.std.. < -0.329573
laying
tGravityAcc.max...Y < -0.013075
walk
walkdown
standing
sitting

Figure 2. Pruned tree showing the values of the 5 features used to sort the data into the 6 activities of laying, standing, sitting, walking – up, walking and walking – down.

Table 3 Confusion Matrix of actual activity (vertical) versus classified activity by pruned tree based on training data.

				Predicted values				
		Laying	Sitting	Standing`	Walk	Walkdown	walkup	**Recall**
Actual values	Laying	221	0	0	0	0	0	**100%**
	Sitting	0	159	39	0	0	0	**80.3%**
	Standing	0	0	227	0	0	0	**100%**
	Walk	0	0	0	256	7	3	**96.2%**
	Walkdown	0	0	0	7	183	3	**94.8%**
	walkup	0	0	0	8	28	174	**82.9%**
	Precision	**100%**	**100%**	**85.3%**	**94.5%**	**83.9%**	**96.7%**	**92.8%**

Table 4 Confusion Matrix of actual activity (vertical) versus classified activity by pruned tree based on test data.

		Predicted values						
		Laying	Sitting	Standing`	Walk	Walkdown	walkup	**Recall**
Actual values	Laying	293	0	0	0	0	0	**100%**
	Sitting	0	176	88	0	0	0	**66.7%**
	Standing	0	31	252	0	0	0	**89.0%**
	Walk	0	0	0	222	7	0	**96.9%**
	Walkdown	0	0	0	9	191	0	**95.5%**
	walkup	0	0	0	95	71	50	**23.1%**
	Precision	**100%**	**85.0%**	**74.1%**	**68.1%**	**71.0%**	**100%**	**80.1%**

Next the simpler, 5 feature model was applied to the test set and the misclassification error rate was found to be 19.9% indicating that 80.1% of the activities in the test set were classified correctly. The most important next step in the analysis would be to find a few transformations of the given features that could be used to distinguish better between sitting and standing and amongst the walking activities.

Conclusion

An analysis of human activity data based on measurements recorded on a Samsung smartphone was conducted in order to create a function that could utilize the measurements to predict a subject's activity. Using the tree classification method and pruning technique a model based on 5 features demonstrated an 80% accuracy rate of classifying the activities on a test set of data. Further refinements of the model could be made by pursuing mathematical transformations of the measurements provided to help more accurately distinguish between activities. Another avenue to pursue would be a multiclass Support Vector Machine as mentioned in reference [2].

References

[1] A useful article about human activity research and this data can be found at https://sites.google.com/site/smartlabunige/software-datasets/har-dataset

[2]A Public Domain Dataset for Human Activity Recognition Using Smartphones https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2013-84.pdf

[3] Samsung data – preprocessed for R:
https://spark-public.s3.amazonaws.com/dataanalysis/samsungData.rda
[4] Samsung data – raw:
http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones
[5] R Core Team (2012). "R: A language and environment for statistical computing." URL:
http://www.R-project.org
[6] R script use for this analysis -
https://github.com/maplano/projects/blob/master/script%20for%20Activity%20Recognition%20Project.txt
[7]k-means article in Wikipedia: http://en.wikipedia.org/wiki/K-means_clustering
[8] K-means clustering – R documentation

http://stat.ethz.ch/R-manual/R-devel/library/stats/html/kmeans.html
[9]"Tree-based models" Quick-R tutorial: http://www.statmethods.net/advstats/cart.html

[10]Package 'tree' documentationhttp://cran.r-project.org/web/packages/tree/tree.pdf

[11] The built-in cv.tree() function was used. A good explanation for how the cross-validation is done can be found at this Montana State University site. "Lab 6 --- Classification Tree Models" http://ecology.msu.montana.edu/labdsv/R/labs/lab6/lab6.html