Brian Castro
Isay Garcia
Joshua Ramirez

Transportation Mode Detection

Introduction

Our project's goal was to create a model that could accurately predict the method of transportation based on given feature variables. Our data consisted of three separate data files. All three datasets included 5894 samples but varied in feature variables and sensors. The first dataset (Dfirst) had 12 features derived from 4 sensors. The second dataset (Dsecond) had 32 features derived from 8 sensors. Finally, the third dataset (Dthird) had 36 features derived from all 9 sensors. Using these three datasets, we set out to create a model that could accurately predict the method of transportation.

Preliminary Analysis

Before we could create any model, we had to first find a way to select feature variables that would best represent our dataset. So we began by using the smaller dataset, Dfirst. We first tried the information gain method in which we attempted to determine which feature/s provided the largest gain in information from including them in a model. Our initial analysis using this method concluded that feature variables that measured standard deviation provided the highest amount of information. Now that we had identified which features were important, we attempted to create our first model.

Decision Trees (Version 1)

Our first model was a Binary based Decision Tree. We began by first converting our target variables (Bus, Train, Car, Walking, and Still) into a binary classification of 1 and 0's depending on whether the target variable could be classified as vehicle or non-vehicle. We then created a testing and training dataset from Dfirst based on the classification adjustment. We then ran our training dataset through a binary decision tree that was grown to its entirety and plotted the results.

However, after looking at the decision tree we noticed that overlapping was present in our decision tree. Therefore, we had to adjust values in our model, such as CP values, in order to prune the tree to produce a decision tree where cross-validation error was minimized. Once this was done, we then ran our resulting model against the testing data set. This resulted in a classification accuracy of 85.41 percent. Since the accuracy was rather high we decided to move on to testing this model again using a non binary decision tree. The non binary decision tree ran using the unmodified target variables (meaning that we left the target variables in their original state: Bus, Train, Car, Walking, Still). However, this resulted in a lower accuracy that sat at 62.33 percent. This made us go back and review our model and review how we were defining and selecting our feature variables.

Brian Castro
Isay Garcia
Joshua Ramirez

Further Analysis

Our original method for obtaining the most important feature variables resulted in poor classification accuracy in our decision tree model. So we decided to try again to obtain the best feature variables using another method: Linear Regression. Starting with Dfirst we set the target variable (method of transportation) as the dependent variable and everything else as an independent variable. This way we could determine how the different variables in our dataset affect the target variable. Then we proceeded to run Forward, Backward, Stepwise and Subset regression in an attempt to determine which feature variables to include in our model.

Forward Selection is a multi step method in which a model is created by first selecting the most significant feature variable on step 1 and then evaluating the performance of the model. Every step selects the next most significant feature and re evaluates the model each time a new variable is added to it. This is done until there are no more significant variables to add to the model. The resulting summary from the Forward Selection method displays the feature variables that best represent the model. However, an issue with Forward Selection is that if it selects a variable as significant in step 1 for example, but then becomes insignificant in step 7, that variable is not removed from the model and is kept instead. To deal with this issue we chose to also run the Backward Selection method.

Backward selection is a multi step method in which a model is created by taking all the variables present in the dataset and then removing the most insignificant variable in step 1 and then evaluating the performance of the model. Every step selects the next most insignificant feature variable and re evaluates the model each time a new variable is removed from it. This is done until there are no more insignificant variables to remove from the model. The resulting summary from the Backward Selection method displays the feature variables that were removed from the model in order to produce the best model. However, an issue with Backward Selection is that if it selects a variable as insignificant in step 1 for example, but then becomes significant in step 7, that variable is not added to the model and is removed instead.

Stepwise Selection is a combination of both methods in simplest terms.

Subset Regression is a method that creates a multitude of models based on the amount of variables we have and then compares the models against each other. This method is used to determine the number of variables needed to create the best model to represent our dataset.

Using the methods described above, we came to the conclusion that out of the 12 feature variables present in Dfirst, only 9 features were significant to representing our data set.

Brian Castro
Isay Garcia
Joshua Ramirez

Decision Tree (Final Version)
After determining which feature variables best represented the dataset, we created a new dataset that included only the important feature variables. We then created a testing and training dataset again and ran them through a non binary decision tree just like we did in Version 1 of our decision tree. Like with our first model, we had some overlapping issues. So, we had to adjust values in our model, such as CP values, in order to prune the tree to produce a decision tree where cross-validation error was minimized. Once this was done, we then ran our resulting model against the testing data set. This resulted in a classification accuracy of 76.86 percent. This was a much better result when compared to our first attempt. So we proceeded to move forward with this model and attempted to run it again using the Dthird data set. The accuracy that resulted from this was 85.05 percent.

Support Vector Machine - All vs One Approach
We began building our support vector machine by first converting the target variables into binary values - Vehicle/Non-Vehicle, with vehicle being 1, and otherwise 0. Using the most important features selected from our linear regression model in the third dataset, we obtain a 91 percent accuracy in classifying vehicles and non-vehicles. Taking the correct predictions from the first classification, we split the data further between each group. Classifying the non-vehicles, “Still” and “Walking”, we obtained .97 accuracy. Classifying the vehicles group, we obtained .89, .88, and .86 accuracy for “Train”, “Bus”, and “Car” respectively. Overall, this method produced a 76% classification accuracy for the entire test dataset.

Support Vector Machine - One vs One Approach
Our next approach uses R’s built-in multiclass support vector machine to predict transportation mode based on all the feature variables associated with the third dataset and yields a 77% accuracy with the test dataset, this suggests that despite our best efforts to choose the most effective feature variables, SVM may not be the best choice of method to use for our purposes, therefore we refer back to classification trees and a different method to attain a method with more predictive power.

Random Forest
We attempted another method for predicting the transportation mode that was not taught in class. We used the Random Forest library in R for our 3rd and final model. Random Forest is a collocation of Decision Trees where each tree draws a sample from the training data set and one-third is the Test data set. Overfitting was a problem in our Decision Trees, so using Random Forest reduces the risk of overfitting. Another problem from our previous models was finding which feature variables were important. The Random Forest finds the most important features from the collection of decision Trees. We used the third data set because it had the most feature variables. If we had used the first or second data set, we would have obtained a lower prediction accuracy because the Random Forest would not find a pattern in the collection of decision trees.

We made a bar graph ranking the importance of the feature variables by Mean decrease Gini metric (In the R code). This lets us know which is the most important feature variable in each of the decision trees.

The Random Forest function by default makes five hundred decision trees and the number of splits we tried is six. Using the model to predict the Test dataset (1983 observations) and got a prediction accuracy of 94.7%. This is the best model out of the three we've used. Since this was our first time using Random Forest. For future reference, we can use the tuneRF() function from the Random Forest library to change the parameters.

Conclusion

When used appropriately, machine learning can be a powerful tool in detecting transportation mode using smartphone sensor data. Descriptive summary statistics taken from a combination of sensors in a 5 second window can be aggregated for each sample in a model and yield a high prediction accuracy when using relevant features and using the most appropriate machine learning algorithm for the target variable or the transportation mode. In our first attempt at building a model, the first dataset was used to train a classification tree recognizing its ability to classify multiple discrete classes using binary recursive partitioning. Using all the features in the first dataset yielded 76% accuracy. In our second and third iteration, we use forward, backward, and stepwise selection from our regression model to identify the most important features from the first, second, and third dataset to achieve an 85% accuracy at predicting transportation mode using classification trees. We then approach the problem through a different lens by implementing a support vector machine algorithm to classify the transportation mode via two categories: vehicle and non-vehicle, then further classifying the transportation mode within each of these classes, using only the most relevant features for each group. Doing so yielded 91% accuracy at classifying vehicles from non-vehicles and furthermore predicting the non-vehicles group, "walking" and "still", with 97% accuracy, and the vehicle group, "train", "bus", and "car" with .88,.87, and .86 percent accuracy respectively. Overall this method produced a 76% accuracy for predicting the test dataset and 77% when using multiclass svm to predict all 5 transportation modes in a single model. Finally, we use the Random Forest algorithm to produce multiple, random classification trees to perform classification based on the class that is predicted by the most classification trees and reducing the error. This yielded our model with the highest prediction accuracy at 94.7%. We also conclude this is the most appropriate method for detecting transportation mode using sensor data from smartphones.

Contributions

Brian Castro - SVM, Random Forest Models, and Final Report
Isay Garcia - Feature Selection Methods, Decision Tree Models, and Final Report
Joshua Ramirez - Decision Trees, Random Forest Models, and Final Report