CS434 Final Project Report (3-page limit)

Peter Dorich, James Walter May, Garrett Bauer

6/13/2018

1 Feature formulation and preprocessing

1.1 Features

What are the features you feed to your learning algorithm? Did you simply flatten the 7 rows into a vector for features? Did you transform or aggregate the given data to engineer your own features?

For all of our attempts, we took the data from each 30 minute chunk and reduced it to just 9 features by taking the average of each column. This allowed us to keep the features smaller and (hopefully) more general, and it also allowed us to use chunks that didn't split up nicely into 30 minute intervals.

1.2 Preprocessing

Did you pre-process your data in any way? This can be for the purpose of reducing dimension, or reducing noise, or balancing the class distribution. Be clear about what you exactly did. The criterion is to allow others to replicate your works.

The preprocessing for the Decision Tree algorithm was a simple normalization function. No preprocessing was done on the data used for the neural network. For KNN, a simple was normalization was used. No preprocessing was used for the neural network.

2 Learning algorithms

2.1 Algorithms explored

Provide a list of learning algorithms that you explored for this project. For each algorithm, briefly justify your rationale for choosing this algorithm.

For this project we explored k-mean, K-nearest neighbor, decision trees, and a simple neural network. k-mean seemed like it would only work with large numbers of positive data points, so k-nearest neighbor was used instead since only a few positive data points are needed to get a working model.

We chose to use a neural network because it is one of the most powerful supervised learning classifiers that can find relationships in the data that might be hidden to a logistic regression classifier.

2.2 Final models

What are the final models that produced your submitted test predictions? For the k-nearest neighbor model, the accuracy for different k's were plotted in a range of 3-15, where k=3 showed the highest accuracy. For the general population, a neural network with 3 hidden layers of sizes 2, 4, and 6 respectively was used. For the first individual, a network with 3 hidden layers of sizes 9, 3, and 3 was used, and for the second individual, a network of sizes 9, 3, and 6 was used.

3 Parameter Tuning and Model Selection

3.1 Parameter Tuning

What parameters did you tune for your models? How do you perform the parameter tuning? For the k-nearest neighbor model, the accuracy for different k's were plotted in a range of 3-15, where k=3 showed the highest accuracy. For the neural network, we used a triple nested loop to check different combinations of hidden layer sizes. Networks with 2 and 4 hidden layers were checked too, but on average the F1 scores for 3-layer networks were higher.

3.2 Model selection

How did you decide which models to use to produce the final predictions? Do you use cross-validation or hold-out for model selection? When you split the data for validation, is it fully random or special consideration went into forming the folds? What criterion is used to select the models? For k-nearest neighbor, the general population method used hold-out, where one of the participants data was removed to ensure the highest accuracy rating as well as be used for validation. For the individual method, cross validation and hold-out would be more detrimental than useful since there were so few data points. For the neural network, we held out a section of each individual's data to compare against. Then, the data was run through a variety of different network structures, the best of which was determined by the highest F1 score. For the general population, a similar method was used, but instead of holding out a percentage of the training data files, one of the four files was held out for validation. Again, the network structure with the highest F1 score was chosen for the final predictions.

4 Results

Do you have any internal evaluation results you want to report? Even though we did not end up using the k-mean algorithm, it was interesting that for the individual and group, both appeared to have 5 clusters. If there was much more data, the k-mean method might be useful.