CSCI 567 Spring 2016 Mini-Project

David Brenn-Cogen **Congyue Wang**

Abstract

As part of the Santander Customer Satisfaction competition, we use two approach to train the data. The first one is neural network. The structure is consisted of 4 layers with 369, 50, 10 nodes on each layer. We train the data in hidden layers with sigmoid function. In order to improve the performance of the data, we add gradient descent and decrease learning rate as optional choices. But the result is not that good. Then we trained a classification tree splitting branches on Gini's diversity index. Through cleaning the data and limiting the number of branches on the tree we achieved an ROC score ~0.79.

1 Approach

1.1 Cleaning data

The training data was processed to remove any empty columns and constant columns with no variance. Then we normalize rest of the features.

1.2.1 Neural network

The structure is consisted of 4 layers with 369, 50, 10 nodes on each layer. We train the data in hidden layers with sigmoid function. In order to improve the performance of the data, we add gradient descent and decrease learning rate as optional choices.
The main advantage of the resilient back propagation training algorithm is that it can eliminate these harmful effects of the magnitudes of the partial derivatives. During the resilient backpropagation process, sign of the derivative is used to determine to which direction the weight will be updated. However, the magnitude of the derivative has no effect on the weight updating process. The size of the weight change is determined by a separate update value. If two successive interactions have the same sign, then the weight and bias in increased by the factor. If two successive interactions have different sign, then the weight and bias in increased by the factor. The update value remains the same if the derivative is zero. From here we can see that if the weights are oscillating then the weight change is reduced. The magnitude of the weight change is increased if the weight continues to change in the same direction for several iterations.

1.2.2 Classification Tree

Using a classification tree with each branch splitting on Gini impurity allows the model to select the most significant features for classification. Initially we tried splitting on information gain,

however splitting on Gini impurity provides better results. For each column attribute, the model finds all of the unique values, then picks a subset of points in between the unique values to use as splitting points.The splitting point which provides the maximum improvement in Gini impurity is chosen. Then the attribute with most significant splitting point is chosen to split the tree on that point. The alogithm stops once improvements to Gini impurity reach a certain threshold in order to prevent overtraining.

1.3.1 Running neural network

We first use all the data to do the training but we find that it is very slow and results are almost zeros. We try to make it run faster by getting the Eigen value of the features and sot them in a descending order. We picked up the top features to do the trying. We picked 10 features first, with learning rate to be 1,2,3,4,5. The result is not so good. We then enabled gradient descent and decreasing learning rate method to see if we can get better result. But it seems that the result is bad. Then we tried to add more features to the training. We tried 100 features, 150 features and so on. The best result we can ever get is 0.553.

1.3.2 Running classification tree

Run train.m to load the training data and train the classification tree.

2 Results

Table 1: Santander results

Model	
ClassificationTree	0.796917
Neural Network	0.5531656

2 Future

For the neural network, the result is not that good and we think the reason may due to small learning steps. We tried 10000 steps to train the data. The result may also due to relatively small hidden layers and hidden nodes. If we can increase hidden nodes and hidden layers the result may be better. Of course increasing hidden nodes and hidden layer will increase computation time.

The classification model could be improved in the future by using an ensemble model with Adaboost. Due to the way in which split values are chosen for each branch of the tree, each tree classifier is unique. Therefore training with multiple learners should improve classification. Another approach to improving our classification score should be a closer look at processing and cleaning the training data. Many of the column attributes are not significant to the classification problem and may be removed in order to speed up training time. Any decrease in training time makes it possible to run more learners as part of an ensemble approach.