HomeWork 09

Ruzan Sasuri

Abstract

For this homework we have to implement the k-Nearest Neighbors algorithm to find and remove the outliers from a training data set, so that it can be used in training a classifier for improved accuracy.

1 Overview

The k-Nearest Neighbors algorithm is a classification algorithm that works by classifying a record as the mode of the classes of the k nearest points to it. Although it is a classification algorithm, k-NN can be used as a way to clean the data by finding outliers in the data. After using k-NN, the clean data can then be further used as a training set by other classification algorithms. In my case, I am using the clean data on a modified version of the decision tree algorithm I created for an earlier homework. I need to then find if the cleaned data provides a boost to the classifier's accuracy.

2 Approach

2.1 k-NN

We first need to find the best k. To find the best k, we run k-NN for all values of k from 1 to n-1 where n is the number of records. On finding the updated classes, we calculate the misclassification rate for that value of k. We then choose the first k at which we get the lowest possible misclassification rate.
For each value of k, we first find the k nearest neighbors within a certain threshold. We find the k nearest points for a point using the square of the euclidean distance, whose distance is less than the threshold. For this assignment the threshold has been set to 8. Now if a point does not have k neighbors, it means that there weren't k points within the threshold distance of the point. We call these points outliers and remove them for the current iteration of k. My algorithm found 17 outliers out of the 325 attributes. This means 308 records are retained. Once the k nearest neighbors are found, we then find the new class value for the point as the mode of the class values of its k nearest neighbors.
Once we find the best k values, which is 3 for my algorithm, we then run a k-NN as above for the best k, and then use the remaining points with their new class values as the training set for the decision tree algorithm shown in the next subsection.

2.2 Decision Tree Trainer

The trainer creates a decision tree using the data set that was cleaned by the k-NN. We first split the training data into two parts, 70% of the training data is used to train the data and 70% is used to test the created classifier. We then run the decision tree algorithm from a previous homework, with a few changes. On creating a decision stump on WEKA I found that the stump created had an accuracy of 63.5%. So I decided to try and create a decision stump as my classifier. To do that I had to change the homogeneity threshold to 55% of the class values after a split.

2.3 Decision Tree Classifier

I used the testing set I created as mentioned in the previous subsection to test the accuracy of the new classifier. This creates a new file that contains the newly classified class value for each record.

2.4 Accuracy Calculator

I use the testing set created in the subsection Decision Tree Trainer and the class values from the classified file created in the subsection to find the accuracy of the created classifier. In my case, the accuracy is 61.29%.

3 Challenges

It was challenging trying to find the best k value as it took 30 minutes each time I had to run k-NN for n-1 k values. It was also difficult deciding how to decide which was the best k value to use. Once those were done, the next challenge I faced was in changing the decision tree training algorithm to create a tree which was had a more suitable accuracy as shown in WEKA. Since I had to test for as many homogeneity values as possible, that took time. Once that was done, I had to retry the same for all possible k's and distance thresholds. So all together finishing the homework was not as challenging as it was time consuming.

4 Conclusion

In this homework we had to implement the k-Nearest Neighbors algorithm to find and remove the outliers from a training data set, so that it could be used in training a classifier for improved accuracy. All in all the homework was an interesting one as we had immense freedom to use k-NN as much as possible. I hope to use k-NN in the future, and find more ways to use k-NN as a pre-processing tool for other classifiers.