HW4: Anomaly Detection

Prakhar Dogra

Email: pdogra@gmu.edu

G Number: G01009586

Team Name: pdogra

Introduction

Given Anomaly Detection problem has been solved using StrOUD Algorithm using Local Outlier Factor (LOF) as strangeness function. The solution includes applying Fast Fourier Transform (FFT) followed by Principal Component Analysis (PCA) on the given data set, to create smaller feature vectors as opposed to the given feature vector size of 20000. K Nearest Neighbor (KNN) algorithm has been used in the LOF score calculation. Using the calculated LOF scores of the normal examples, StrOUD algorithm was applied to calculate p-value of each test data point.

Score and Accuracy

As of 04/21/2017 3:00 a.m., public score is 1.0 and rank is 3.

Team Name

pdogra

Approach

Following Python files were created in the order of approach taken:

test_lof.py and *test_knn.py* (Testing StrOUD algorithm with given training data using LOF and KNN as strangeness functions)

Firstly, files (File#.txt) from Mode A, B, C and D (normal data) are loaded and stored as list of lists. Fast Fourier Transform is applied on it followed by Principal Components Analysis. Then LOF/KNN scores are calculated for each point and stored. This is called our baseline. Now we load the examples from Mode M (anomalous data) and stored as list of lists. Again Fast Fourier Transform is applied on it followed by Principal Components Analysis. Then LOF/KNN scores are calculated for each point. And finally using the StrOUD algorithm, p-value of each test point is calculated. To check the accuracy, we repeat the procedure for different confidence values (from 99.5 to 90).

Later, to get better results at the expense of more computation, we didn't apply Fast Fourier Transform and Principal Components Analysis. Even though the initial number of features were 20000 and Principal Component Analysis reduced the number of features to 400, the computation didn't take much time so it was safe to calculate LOF/KNN scores without applying PCA and FFT.

Following graphs show the accuracy increase against confidence decrease for both approaches (first using LOF as strangeness function and then using KNN as strangeness function):

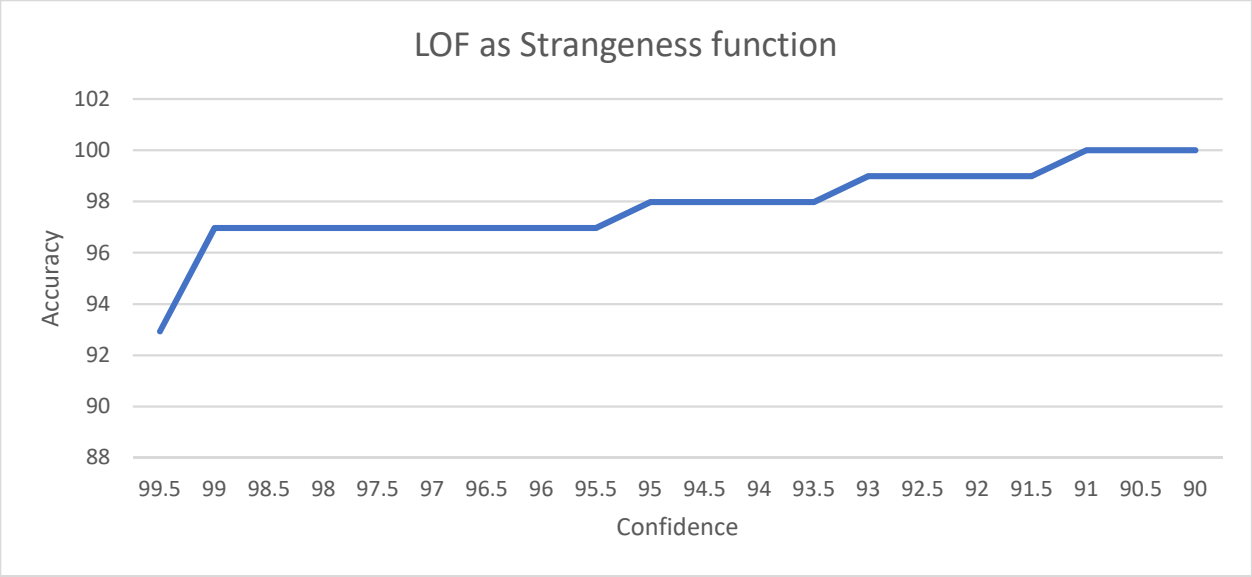

Figure 1 : Accuracy vs Confidence graph when using LOF as strangeness function

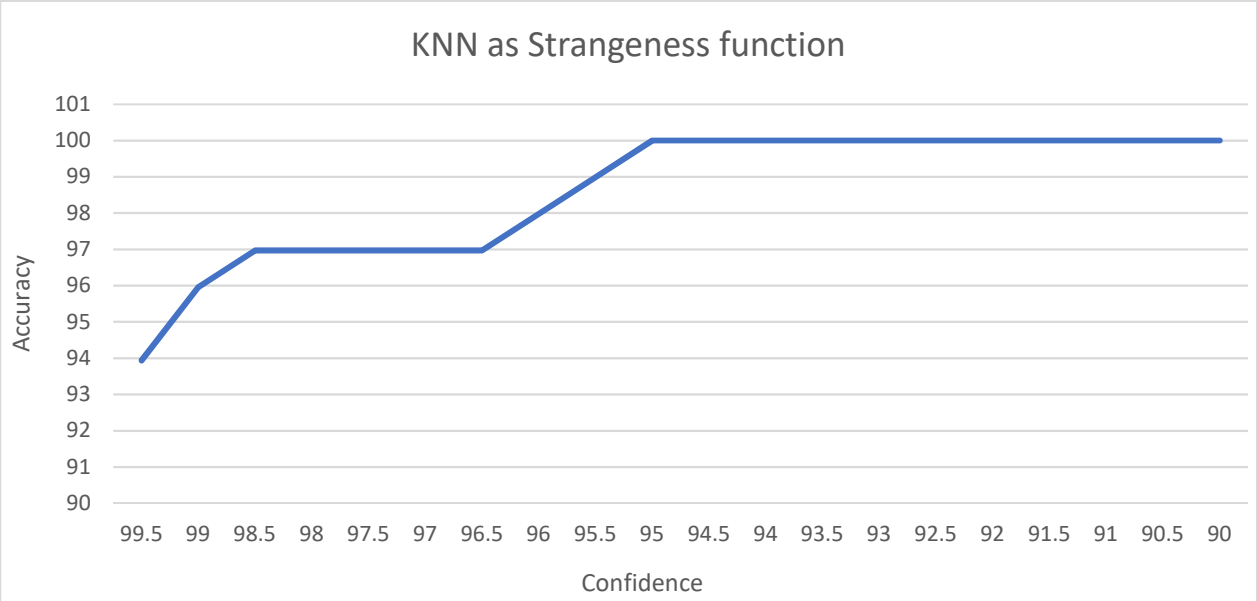

Figure 2 : Accuracy vs Confidence graph when using KNN as strangeness function

lof.py and *knn.py* (Calculates p-values using the StrOUD algorithm using LOF and KNN as strangeness functions)

Just like during testing, files (File#.txt) from Mode A, B, C and D (normal data) are loaded and stored as list of lists. Fast Fourier Transform is applied on it followed by Principal Components Analysis. Then LOF/KNN scores are calculated for each point and stored. This is called our baseline. Now we load the examples from Test Folder and store them as list of lists. Again Fast Fourier Transform is applied on it followed by Principal Components Analysis. Then LOF/KNN scores are calculated for each point. And finally using the StrOUD algorithm, p-value of each test point is calculated. These p-values are written to file named “results.txt”.

Methodology

FAST FOURIER TRANSFORM

Fourier analysis is used to express a function as a sum of periodic components, and for recovering the function from those components. Although applying Fast Fourier Transform gives us an imaginary number, we only need the coefficient (real part of the complex number). It helps reduce any noise in the data set.

PRINCIPAL COMPONENT ANALYSIS

Principal component analysis (PCA) is a statistical method to reduce the number of features in the data set.

In our data set, each data point has 20000 features so PCA was applied during testing from training set as well as when generating results from test data. Although to get better results, PCA code was commented out. At the expense of little increase in computation run time, we got slightly better results.

LOCAL OUTLIER FACTR VS K NEAREST NEIGHBOUR COMPARISON

When I used LOF as strangeness function, I got decent accuracy and then I submitted my solution on Miner website and got a score of 0.88. So I thought of using KNN as strangeness function and got a higher accuracy for the same set of parameters. Moreover, I got a score of 1.0. So I thought of comparing the two strangeness functions.

Later I found that removing PCA code for LOF, improved the accuracy and the leader board score. As instructed by the professor, my current submission on the Miner Website is of LOF.

STROUD ALGORITHM

When using LOF strangeness function we compare the LOF score of the respective point with the baseline. Whereas when using KNN as strangeness function we compare the sums of K Nearest Neighbor distances. Moreover, both methods require the use of KNN algorithm to find K nearest neighbors. For our purposes, we chose K = 3. Since for KNN we already had an accuracy of 100 upon removal of PCA code (not FFT), increasing the value of K didn't change anything. The same was observed when LOF was used as the strangeness function.