Machine Learning: Final Programming Project[1]
Anomaly Detection Final Project
Due December 10, 2019

Overview

The objective of this assignment are:

- Implement the StrOUD algorithm using LOF (local outlier factor) as the *strangeness* function. LOF and StrOUD are outlined in detail in the lecture slides
- Handle signal data collected from a device that controls a centrifuge (see below), adapting the data to create features from the data
- Find anomalous and normal data in the test set and assign p-values to the predictions

You are working as a intelligence analyst and are supplied electromagnetic signal data captured by a sensor attached to a control unit for a centrifuge. This centrifuge has the capability to be used for enriching uranium. The centrifuge has four standard modes of operation (call them **A**, **B**, **C**, **D**). For each mode, $\approx$ 100 samples of the signal data (each sample has 20,000 attributes) are provided in 4 directories, one for mode **A**, ...). An additional mode, mode **M**, has been created by the infection of the Stuxnet virus. A folder with samples from the **M**, or maliciously infected mode, are also provided. A file with all this signal data is available from the Canvas assignment page and is called **SignalData.zip**.

Your mission, if you choose to accept it, is to develop an anomaly detection algorithm that identifies infected centrifuges. A general outline to develop this solution is:

- Create a baseline of signals, using data from all the normal modes (A, B, C, and D). Do **not** include data from Mode M in this baseline.
- Extract features for each signal, resulting in a vector of fixed dimensionality (you are free to decide which features you want: examples are coefficients of the Fast Fourier transform (FFT), or PCA components of the data; it's advisable that you keep the number of features at a reasonable level, for performance reasons). I am providing code to get the coefficients of the FFT.
- Use this baseline to create a sample distribution of strangeness using LOF.
- Apply StrOUD to the test data provided, using this sample distribution.
- You can reserve some of the data in directories Model A, Mode B, Mode C, and Mode D, and Mode M for your own, private testing. It is up to you to decide how much data from the first four modes you want to include in the baseline.
- You must use your labeled test data (data from the normal modes and the malicious mode) to tune the hyper-parameters in your model (that is, the hyper-parameters of the *strangeness* function). This is a crucial step. You should use the F1 statistic to select which hyper-parameter is best.
- Your code should then apply this tuned model (the one with the best hyper-parameters) to the test data and produce the pvalues in the requested output file name (a command line parameter named –pvalueFile).

A paper that describes StrOUD in more detailed than the class notes is available on canvas under this PA description. A video on how to implement StrOUD is also available on the class website (search StrOUD on the calendar page).

[1] Assignment originally developed by Daniel Barbara

Output and Validation

Your program must accept a single **testing** dataset, which will be a directory name that contains files similar to the other supplied earlier for directories (files with .txt extensions that each contain the same 20,000 features as the earlier data for Mode A).
Using your best model (with the optimum hyper-parameters), your program must calculate the **pvalue** of each sample and print it to a file. Each of these values should be printed on their own line and with 5 decimal point precision (e.g., 0.81472). For the provided test dataset, this file will have 499 lines.

You will submit this file to Autolab, which will use the area under the curve (AOC) of the ROC metric comparing predictions made with your pvalues to the ground truth.

Autolab will show results based on a randomly constructed a test set (the ground truth for the supplied test set is only available to Autolab). This is standard practice on sites like Kaggle to prevent teams from "tuning" their submissions based on the feedback they receive.

Collaboration Policy and Package Usage

For this project, you are only allow to use the following python packages: numpy, math, sys, pandas, keras, tensorflow, sklearn, matplotlib, seaborn.
If you need more than these packages, please **request** them from me (as maybe I omitted a few here).
For this assignment, you can work either **individually** or in **pairs**. As before, helping colleagues to find a bug or to help them understand some concepts is OK, sharing of code is prohibited (outside of your pair). You should be able to explain all of your code to me, as failure to do so will be an indicating that you over-collaborated. This is a machine learning class, so, do not be surprised that I am using text mining tools to detect similar code to the class and to Internet sites.

If you have questions about this policy, please see me.

Deliverables

This PA has the following deliverables. If you are working in a pair, please submit to Autolab and Canvas only once per pair.

Autolab Deliverable

Submit a single file as requested above that contains your calculated pvalues for the test data provided. Your calculated AUC value will be posted as your score.

Canvas Deliverable

The following deliverables must be placed in a single zip file named CS480Final.zip. This zip file should include the following:

1. Your python code (which should be a single file) named **centStroud.py**. This file should be based on the template file provided on the website (and therefore work with the command line parameters).
2. A small report (2 page + images, so, no more than 5 pages) in **PDF format**. Your report must:
   - Include the name of all team members
   - Describe the problem you are trying to solve.
   - Describe StrOUD at a very high level (what does it do)
   - Describe your approach for tuning hyper-parameters.
   - Produce a plot that illustrates the performance measure(s) you utilized to judge the quality of your model and its values across the hyper-parameter values your evaluated.
   - A plot of the ROC curve for test data you created. Recall that you can take a mixture of the normal modes (A, B, C, D) and M on your own to test your method. You can use sklearn to calculate the ROC. You must provide a 2 to 4 sentence description of the curve and what it means.

Rubrics

The following is the breakdown how this assignment will be graded:

Item	Description	Points
Readability/Style	Program is commented, following naming style for either Python/Java. Points will be deducted for poor variable names or unreadable code	10
centStroud.py	Submit your only python file as part of the zip.	5
	Method tunes hyper-parameters correctly	10
	method computes LOF correctly	15
	method computes baseline data (for StrOUD) correctly	10
	method formats pvalues correctly and receives a score above 85% on their test submission.	15
$plot_1$	ROC curve is contained in the report	10
$plot_1$ comments	Comments are made about the ROC curve per the specifications	5
problem description	quality problem described in the report	5
STRoud description	quality of StrOUD overview in the report	5
Hyper-tuning plot	present in the PDF and includes a brief discussion (3 to 6 sentences)	10