Data Mining Lab 1 – Anomaly Detection

INTRODUCTION

Often data is multivariate, sequential, and unlabeled. Mining sequential data is difficult because past data (rows) provide information for future data (rows). The data points are thus not i.i.d (independently and identically distributed). Learning from sequential data and time series is a large domain covering many problems, solutions, and algorithms. Multivariate simply means that there is data from more than one feature/signal. Specialized algorithms exist for dealing with multivariate sequential data. Here we study them separately.

In this exercise, you will apply the techniques taught in class to the problem of anomaly detection in SCADA systems. Anomaly detecting is typically harder than classification because the data are unlabeled. We must rely on statistics such as occurrence counts or value ranges to find anomalies, rendering many machine learning methods inapplicable. Securing SCADA system is considered one of the most important problems in cyber security.

Together with a fellow student, you will implement two approaches for anomaly detection. The first uses a distance function based on sequence alignment. Essentially, a data-point is an anomaly when its distance to its nearest training data point is above a threshold. The second ignores all sequential information and aims to find unexpected co-occurrences of feature values using matrix factorization. For this method, a data point is an anomaly when combinations of feature values occur that did not occur at train time.

LEARNING OUTCOMES

After completing this assignment, you will be able to:

1. Implement the Dynamic Time Warping sequential distance.
2. Implement the Principal Component Analysis matrix factorization.
3. Detect anomalies in multivariate sequential data.
4. Build an anomaly detection pipeline and evaluate its performance.

INSTRUCTIONS

Dynamic Time Warping – (25 points)

Implement the **dynamic_time_warping()** subroutine in the provided code base. This function takes as input two vectors of floating-point numbers and returns a floating-point number – the computed distance – as output. Test your implementation in WebLab. The number of points you get is equal to the number of test cases it computes correctly. Look in WebLab for further instructions.

Principal Component Analysis – (25 points)

Implement the **principal_component_analysis()** and **apply_principal_component_analysis()** subroutines in the provided code base. The principal_component_analysis() function takes as input a data frame and an integer that denotes the number of required eigenvalues. It should return the **eigenvectors** it computed from the dataframe. The apply_principal_component_analysis() function takes as input a data frame and a list of **eigenvectors**.

Familiarization (10 points)

Load the sensor data (train with training data, test with test data) into a Jupyter Notebook and understand the data using visualizations. Answer the following questions:

1. What types of signals are there?
2. Are the signals correlated? Do they show cyclic behavior?

Visualize these types and the presence or absence of correlation.

DTW-based anomaly detection – (15 points)

Choose one signal that displays interesting temporal behavior. Build a set of representative sliding windows from the training data. This can be all data points, but to lower the run-time you can consider taking a subset. Plot the distances (sometimes called residual) of the train and test data points of this signal to their closest representative window (excluding overlapping ones). What kind of anomalies do you expect to detect using DTW distances in this manner?

Experiment with using different sliding window sizes and jumps (sometimes called stride). Set these in such a way that you expect to detect anomalies with a low number of false alarms and such that the run-time is not too large. Show your analysis and explain your expectation.

PCA-based anomaly detection – (15 points)

Perform PCA-based anomaly detection on the signal multivariate data points (do not take sequential context into account). You compute PCA to the train data, and apply it to both the train and test data. Plot the distance (residual, your choice of distance function) between the original and reconstructed data points. Do you see large abnormalities in the training data? Can you explain why these occur? It is best to remove such abnormalities from the training data since you only want to model normal behavior. Describe the kind of anomalies you expect to detect using PCA.

Plot the PCA residuals for different number of components on the training data as one signal. Choose the number of components based on the residuals and detected anomalies. Aim to set it such that you expect a small number of false alarms while still able to detect anomalies. Show your analysis and explain your expectation.

Bonus – (10 points)

Try to outperform our baseline on the Kaggle competition! Feel free to use a distance-based method, a factorization-based method, or a combination. You are not allowed to use scikit-learn (or any other machine learning) libraries for your submission. Your solution has to run your own code for detection anomalies.

RESOURCES

See Brightspace for details.

PRODUCTS

A Jupyter Python notebook for all parts of the assignment. The word count should not exceed **1600 words**. You are not allowed to include libraries other than numpy, scipy, and pandas. **Do not include the data with your notebook!** The data will be available on the evaluation machine.

The notebooks will be assessed using the below criteria.

ASSESSMENT CRITERIA

The assignment will be reviewed by your peers, and you are expected to individually review 2 reports. The estimated time you should spend on a review (including code review) is 1 hour. The login details will be provided in the week of the deadline.

Knockout criteria (will not be evaluated if unsatisfied):

Your code needs to execute successfully on computers/laptops of your fellow students (who will assess your work). You may assume the availability of 4GB RAM. Please test your code before submitting. In addition, the flow from data to prediction must be highlighted, e.g., using inline comments.

Submissions submitted after the deadline will not be graded, **deadlines are strict!**

The report/code will be assessed using these criteria:

Criteria	*Description*	*Evaluation*
DTW	*Test suite score on WebLab.*	*0-25 points*
PCA	*Test suite score on WebLab.*	*0-25 points*
Familiarization	*Shows the behavior of one-two signals from the SCADA system. Provides useful input for further tasks.*	*0-10 points*
DTW pipeline	*DTW is used correctly, with explanations for the sliding window size and stride. The kinds of anomalies detected are identified correctly.*	*0-15 points*
PCA pipeline	*PCA is used correctly, with explanations for the number of used principal components. The kinds of anomalies detected are identified correctly.*	*0-15 points*
Report and code	*The data-detection flow is clearly described, including preprocessing and post-processing steps.*	*0-10 points*
Bonus	*Performance on Kaggle.*	*0-10 points*

Your total score will be determined by summing up the points assigned to the individual criteria. Your report and code will be graded by the teacher and assistants, and the peer reviews are used as guidance.

110 points (including bonus) can be obtained in each lab assignment. 330 points (including bonus) can be obtained in the 3 lab assignments. The total number of obtained points will be divided by 30 to determine the final lab grade.

The lab grade counts for 30% of the total graded for the course.

You will receive a penalty of 10 points for each peer review not performed. Significantly different reviews will be subject to investigation. If deemed badly done by the teacher or TA, you will also receive 10 penalty points.

SUPERVISION AND HELP

We use Mattermost for this assignment. Under channel Questions Lab1, you may ask questions to the teacher, TAs, and fellow students. It is wise to ask for help when encountering start-up problems related to loading the data or getting the expected output from Numpy. Experience teaches that students typically answer within an hour, TAs within a day, and the teacher the next working day. Important questions and issues may lead to discussions in class.

Lab sessions are Friday's 13:45-17:45 physically in different locations at campus (check mytimetable for locations). Please see Brightspace for details.

SUBMISSION AND FEEDBACK

Submit your work in Brightspace, under assignments. Also submit it on peer.tudelft.nl. Within a day after the deadline, you will receive several (typically two) reports to grade for peer review as well as access to the online peer review form. You have 5 days to complete these reviews. You will then receive the anonymous review forms for your groups report and code.

There is the possibility to question the review of your work, up to 3 days after receiving the completed forms. You should do so via the response function on peer.tudelft.nl.

In case of a failing grade for a lab assignment, you have the opportunity to resubmit your work on Brightspace until one week after grade notification.