Part I: Technical Roadmap

This document describes the final tasks completing the technical part of your term project. Part II details the format and logical organization of the final project report. Building on your skills and experience from the past three group assignments, the tasks listed here expand on the problem scope for unsupervised intrusion detection using time series analysis and forecasting applied to stream data from a supervisory control system. There are several goals: enriching the *feature engineering phase*, allowing for models with arbitrary many states, and advancing the *anomaly detection phase* by analyzing several datasets with injected anomalies. The data analysis, the design and selection of models, and the presentation of the experimental results form the technical basis of five main aspects to be addressed in your project report (as will be detailed in Part II, coming soon).

All groups will work with the same datasets for model training, testing and anomaly detection. This way, the results of your experiments will be comparable to the results of other groups and naturally allow for ranking model performance across all groups.

Please use only the datasets listed under "Term Project**" on the course page.**

Complete the following tasks:

1. **Feature Engineering.** Choose a subset of the response variables for training of multivariate Hidden Markov models on normal electricity consumption data. For deciding on the subset of variables that are most suitable for training your models, you need to perform a Principal Component Analysis (PCA)[1], a superior alternative to using a correlation matrix. Provide a proper rational for your final choice of response variables based on your PCA results. This method is explained in more detail on the next page. Note that you need to scale the raw data using *standardization* prior to applying PCA.

2. **HMM Training and Testing.** Partition your scaled data into train and test. Choose a weekday or a weekend day and a time window between 2 to 6 hours on that day. For this time window, train various multivariate Hidden Markov Models on the train data with different numbers of states. For models with at least 4 and not more than 24 states evaluate and compare the results of log-likelihood and BIC to select the 'best performing' model(s) with an overall good fit on the train data. Note that you do not need to train a model for each and every number of states across the given range. Finally, calculate the log-likelihood of the **test data** for your selected models to decide on the best one among these candidates.

[1] Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called *principal components*.

Note that you need to compare normalized log-likelihood of the train data and the test data. *Hint:* For calculating the log-likelihood of the test data, look at the `fit`-section on Page 15 and the `forwardbackward`-section on Page 21.

https://cran.r-project.org/web/packages/depmixS4/depmixS4.pdf

3. **Anomaly Detection.** Using the above multivariate HMM, compute the log-likelihood for the respective observation sequences associated with the same time window in each of three datasets with injected anomalies that are provided on the course page under Term Project. That is, for each dataset compute the log-likelihood over all instances of the time window over one full year. Compare and interpret the three datasets regarding the degree of anomalies present in each of the datasets in some detail.

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis is a useful technique for analyzing datasets with many variables. It is basically a type of linear transformation which takes a dataset with many variables (i.e., number of responses and number of samples), and simplifies it by turning it into a smaller number of variables, called *principal components*.

This technique also allows you to visualize how the data is spread out in a dataset. The underlying mathematics is somewhat complex though, so we won't go into too much detail, but PCA gives us a number (percentage) for each variable which indicates how much variance there is in the data for that variable.

To have a better understanding, let's assume you have a year worth of multivariate data which has 7 responses. Further assume you choose a time window from <start time> to <end time> on a <weekday>. Therefore, you would have 52 samples for each of these 7 responses. After applying PCA on this data, you obtain 7 principal components. Each of these PCs is represented by a number which explains a percentage of the total variation in the dataset. If PC1 is 65%, it means it has 65% of the total variance; in other words, nearly two-thirds of the information in the dataset can be encapsulated by using this one principal component.

In this part, you should (I) compute the principal components of the original dataset; (II) plot the results (PCs); and then (III) interpret the results. In order to compute the principal components, we recommend to use the stats package (the important commands you may need are `prcomp()` and `summary()`). To plot the result we recommend to use the **ggbiplot** package (it is based on the **ggplot** package).

Please read about **Principal Component Analysis** to gain a better understanding of this concept and also refer to the documentation of the packages you use. There are also three short videos on YouTube explaining PCA in a generally understandable way. You can find these YouTube videos at the following URLs:

https://youtu.be/HMOI_lkzW08
https://youtu.be/FgakZw6K1QQ
https://youtu.be/0Jp4gsfOLMs

SUBMISSION

Please submit the R code of your solutions for all three tasks as well as a **PDF copy** of your **term project report** presenting, explaining and illustrating your experimental analysis and evaluation, and the results through the course page by 28 NOVEMBER 2022, 23:59 PST.