Technical Roadmap - Part I

This document describes the final steps for completing the technical part of your term project and complements the project overview and report description. Building on skills and experience from the past three group assignments, the tasks listed here expands the problem scope for unsupervised intrusion detection in supervisory control systems based on time series analysis and forecasting, specifically by enriching the feature engineering phase, allowing for models with arbitrary many states, offering several datasets with injected anomalies, and advancing anomaly detection. The data analysis, the design and selection of models, and the experimental results form the technical basis of the five main aspects to be addressed in your project report (see the report description for details).

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

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

PART I

Complete the following tasks:

1. Train and Test HMMs. For training univariate and multivariate Hidden Markov Models on normal electricity consumption data choose most suitable dependent variables using *Principal Component Analysis (PCA)*[1] for f*eature engineering,* explained on the last page. Provide a proper rational for your final choice of variables and number of states for the univariate and the multivariate models you trained. Document the performance of these two models on the train and test data for a single time window of your choice.

2. 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 test datasets with injected anomalies, 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.

[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*.

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis is a useful technique for analysis of datasets with many variables. It basically is 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 chose a time window from <start time> to <end time> on <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 (7 variables) can be encapsulated by this one principal component.

Hint: In order compute the principal components you need to have a single value for each response in each sample. Considering the fact that we are dealing with time series, we recommend to simply calculate the average of each response values during the chosen time window (the average of values from <start time> to <end time> for each response for each instance of <weekday>).

In this part, you should (I) compute the principal components of the original dataset; (II) plot the results (PCs); (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 use the documentation of the packages you use.