Technical Roadmap

This document describes the final steps for completing the technical part of your course 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.

Complete the following tasks:

1. **Data Exploration.** For the new dataset, representing the electricity consumption for households over a time period of ≈ 4 years, perform a correlation analysis for each disjoint pair of dependent variables. Represent the results in the form of a correlation matrix and visualize the matrix using color-coding to highlight the relevant information.

   Determine a single observation time window during a weekday and a weekend day that shows a clearly recognizable electricity consumption pattern over a time period of several hours. Visualize the observable pattern. Use the same time window for the weekday and the weekend day. You may choose any weekday and weekend day.

2. **Feature Engineering.** Choose a suitable dependent variable as well as a combination of several dependent variables for training univariate and multivariate Hidden Markov models on normal electricity consumption data. To decide on which dependent variables are most suitable to optimize the accuracy, while minimizing the complexity of the models, analyze the correlation information; in addition, perform a *Principal Component Analysis (PCA)*[1] as explained on the last page. Provide a proper rational for your final choice.

[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*.

3. **Training and Testing.** Train and test a number of univariate HMMs and multivariate HMMs that each have a different number of states across a range from not less than **4 states** to any number of states that you deem reasonable to show that any additional states do not result in further increased performance. To do this, you need to partition the original dataset into a train set and a test set. An 'optimal' model is one which represents the time window adequately and is neither overfitted nor underfitted on the data. Find the model with the best number of states and the best combination of dependent variables (multivariate case) by comparing bias and variance of models based on their log-likelihood and BIC values.

   Present the essential model characteristics—including the dependent variable(s), number of states, log-likelihood and BIC values on both the training and the test dataset—of your best univariate model and your best multivariate model in a single table, with the heading *Experimental Results*, summarizing the final results of your experimental analysis.

4. **Anomaly Detection.** For each of the datasets with injected anomalies, perform the following anomaly detection methods.

   Pick a single week and apply the Moving Average method for the chosen time window on a weekday and a weekend day of this week, using the same week for all five datasets. To detect complex anomalies, you need to define several (say three different) thresholds to differentiate anomalies from expected (normal) behaviour. The thresholds represent different margins for the acceptable range of noise.

   Additionally, use your best univariate model as well as your best multivariate model, respectively, as a basis for representing expected behaviour. Compute the log-likelihood for the respective observation sequences associated with these same time windows in each of the five datasets. That is, for each dataset compute the log-likelihood over all instances of the time window over one full year.

   Summarize the anomaly detection results for the Moving Average method and the two HMM models in separate tables.

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.