Third Group Assignment

The value of this assignment is 7%. You will find on the course page a further extended version of the electricity consumption dataset (covering a time period of ≈ 4 years) to be analyzed using the R language and environment.

This assignment builds on the previous two assignments and addresses advanced training and testing of Hidden Markov models for the purpose of **unsupervised intrusion detection** in supervisory control systems. In addition, basic methods for anomaly detection in time-series data from the continuous operation of a supervisory control system will be explored.

Please complete the tasks described below and submit an electronic copy of your solution through CourSys by November 9, 2020.

For the dataset assigned to your group, complete the following tasks:

1. **Data Exploration.** Choose a good combination of observed response variables for the provided electricity consumption data to achieve a reasonable model with regard to overfitting vs. underfitting and model complexity. Determine a time window on weekdays and on weekend days (identical window) that shows a clearly recognizable electricity consumption pattern over a time period of several hours. Graphically visualize the pattern observable for each of the chosen time windows.

2. **Model Training.** Train and test a number of multivariate HMMs that each have a different number of states across a range from not less than **4 states** to not more than **20 states**. To do this, you need to partition the dataset into a train set and a test set. We recommend to separate the last year of the data and consider it your test dataset; use the rest of it as the training dataset. In order to train a good HMM, one which represents the dataset adequately, make sure the optimal model is neither overfitted nor underfitted on the dataset. Find the best model by comparing the characteristic model values using log-likelihood and the BIC.

   Take into account that training a multivariate HMM, when compared to a univariate HMM, takes extra time. You may not need to train an HMM for each and every number of states within the given range.

3. **Model Testing.** After training the model and finding the best number of states for the chosen responses (based on log-likelihood and BIC), evaluate your model with the test data. This means to calculate the **normalized** log-likelihood of the the test data and compare it against the normalized log-likelihood of the trained model. In order to do this in depmixS4, you should create an instance of your trained model (using the command "**getpars**" and "**setpars**"), then feed the test data to it and calculate the log-likelihood.

Please note that you should use "**ntimes**" command in the training process, because you are training an HMM for a specific time window.

Hint: If your model is a fit model (not overfitted nor underfitted), the log-likelihood of the train dataset and test dataset should be close.

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

4. **Anomaly Detection.** For the second dataset—the one which includes anomalies—, apply both the Moving Average method as well as the Log-Likelihood method to the two chosen time windows for detecting anomalies. Decide on a range of threshold values (3 different ones may be enough) and provide some rational for your choices regarding the overall objectives of intrusion detection.

Please submit a short report for your solution and the R code through CourSys by November 9.

Thank you!