Version 1.0 20 March 2023

EH2745 Computer Applications in Power Systems

Assignment II

Overview

The purpose of Assignment II is to let you combine the machine learning techniques and the power system modeling techniques we have introduced throughout the course. You actually are quite free how to design the assignment, using techniques as you choose freely, but the overall structure of the assignment should follow the steps outlined below.

It involves defining a model of a powergrid using PandaPower and then run a time series simulation in this grid to create a dataset of measurements (voltage, power etc) from the your modeled grid. This dataset in turn will serve as the base for a machine learning experiment. The network structure and the parameters of the model are found in appendix 1.

This project assignment shall be solved individually, you can collaborate with friends and you can re-use code that you find online. But the code you hand in must be written by yourself, plagiarism is not OK. If you use code from others, you need to explain what the code can do in the screencast you hand in with your code.

Assignment

The assignment involves developing a Python (or Java) application that fulfills at least the following requirements:

To pass this assignment you should:

1) Define the network in pandapower
2) Implement a k-means clustering algorithm
3) Implement a KNN classifier
4) Use the two implemented algorithms on the data set
5) Report the results

The data you will perform your machine learning experiment on will be the bus voltage magnitudes and angles. You enable computation of voltage angles by adding the parameter calculate_voltage_angles=True to pandapower's run_simulation function.

In the first step, you will generate so called training data for a number of different operating states. A recommendation is to generate 30-70 samples (timesteps) for each operating state.

In the second, you will cluster this data with the k-means algorithm in order to investigate if these operating states are statistically distinguishable, i.e. does the optimal amount of clusters coincide with the number of simulated operating states?

In the last step, you will generate some more so called testing (or validation) data. You will use the first set of data to train a k-nearest neighbor classifier, and then validate the model using your newly generated data set. The class should represent the operating state. You should generate 10-30 samples for each operating state.

Generating training data

Note that the values from Appendix 1 are set when you create the model, they are then overwritten by the parameters used in pandapower time series simulations if you supply a series of values to simulate. ***Important note:*** The training data should contain about 3-5 times more samples than the testing/validation data.

Suggested operating states to investigate and create data for:

High Load
Set the P and Q for each load to a value higher than the default, and add some noise with a standard deviation of about 5-10% of the nominal values.

Low Load
Set the P and Q for each load to a value smaller than the default, and add some noise with a standard deviation of about 5-10% of the nominal values.

Generator Disconnected
Disconnect the generator at bus 3.

Line Disconnected
Disconnect the line between bus 5 and 6.

Additional features
Since this is a statistical assignment at its core, you may try your own definitions of operating states. Choose other lines to disconnect, change parameters of generators and loads, add other equipment. The above stated have been confirmed to show some interesting results, but you are encouraged to experiment.

Use your algorithms on the full 18-dimensional data set. If you try to do a grid average in order to reshape the data to a 2-dimensional one, you are very likely to lose information that may hurt the performance of your machine learning model.

Submission of solutions

The source code and screencast in which you present your solution shall be handed in no later than **the date published in Canvas**. You need to hand in two things:

1. A ZIP file of the GitHub repository containing your software. This code must be runnable, i.e. when the repository is unzipped and installed on another machine, no additional software installation shall be necessary. The GitHub repository shall contain a README file describing the included files.

2. A 10 minute long screencast/video where you explain how your code is structured, which functions you have used and provide a step by step guide through the program code.

Grading of assignment

The assignment is graded in three steps.

The grade Pass (**5 course points**) is awarded to a group that submits a Python application that fulfills the above requirements above. The equipment from the EQ and SSH files that must be at least included for the grade Pass are loads, generators, lines, transformers, buses.

For higher grades, **up to 7 bonus points** are awarded to a group based on the, the quality of the solution above the basic requirements. Below follows a number of examples that describe a high quality project.

Finally, **up to 3 grade points** can be awarded to students who during an individual oral presentation discussing the code and able to answer questions on implementation details. These individual slots are booked separately at course end, and are available to students who score >5 bonuspoints.

Examples of a high quality project follows below:

Good code style
Use the object oriented programming paradigm, separate code into modules and classes.

Consistent naming conventions.

Comments to supplement code.

Try to think logically and put code into function blocks.

Conforms to PEP8.

Implement a GUI
Add a GUI to your program that can be used for instance to:

Additional features of analysis
Add features to identify borderline cases, providing estimates of the correctness of your algorithms.

Use of alternate methods of analysis

In addition to the proposed methods kNN and k-means, explore the use of other methods to predict states.

Late or missing hand-ins
Submissions after the deadline, or re-submissions after receiving comments on an initial submission before the deadline **can only be awarded 5 course points** in total.

References and plagiarism

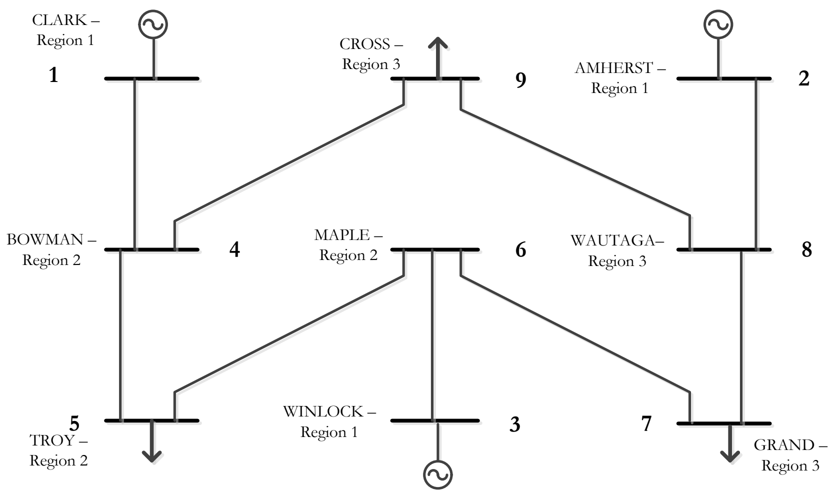
Please note that when solving the assignments co-operation between students is allowed and even encouraged. However, you as an individual are responsible for the content of your own program and plagiarism will result in an immediate failing of the assignment in addition to a written report to KTH's central disciplinary committee. This means that all students should **write their own programs.** You are not allowed to use source code from other groups and you are not allowed to copy source code from the internet.

You are however allowed to use code repositories, but your ability to explain this code as shown in your screencasts and possible individual presentations will be part of the assessment. This means that if you use a code library and cannot explain what the code does, you will not be graded as having passed the assignment, alternately you will not be awarded course points.

Appendix 1: Power System information

The single line diagram of the 9 bus system is shown below along with the active and reactive power of the load and generation busses. Below is all the information you need regarding the power system, note that line length is set to 10 km for all lines

.



Load Bus number	Active Power (Pd) (MW)	Reactive Power (Qd) (MVar)
5	90	30
7	100	35
9	125	50

Generation Bus Number	Active Power (Pg) (MW)	Reactive Power (Qg) (MVar)
1	0	0
2	163	0
3	85	0

Nominal voltage level is 110kV and the linetype for all lines can be set to: “149-AL1/24-ST1A 110.0” Bus 1 is the slack bus along with generation capable of supplying the system should another generator fall out of operation