02427 Advanced Time Series Analysis

Computer exercise 1

This exercise starts with some modelling with non-linear models and afterwards methods for non-parametric estimation are used to identify functional dependencies.

Part 1

Simulate different non-linear models. Three good models to start out with are: SETAR(2;1;1), IGAR(2;1), and MMAR(2;1). Try different parameters. For reporting, write the models and discuss the most essential features of each model together with some informative plots.

Hints
R and Matlab: Check out the script `3dPlotting.R`. It starts out showing a simple way to implement a simulation of a process with a for-loop.

Part 2

Compute the theoretical conditional mean, $M(x) = E\{X_{t+1}|X_t = x\}$, for a SETAR(2,1,1) of your own choice.

Simulate 1000 values from the chosen SETAR model. Use them and a local regression model to estimate the $\widehat{M}(x) = E\{X_{t+1}|X_t = x\}$. Try different bandwidths and comment on your findings.

Hints
R: You can get inspired by the script in `3dPlotting.R` and explore local regression with `loess()` or `lm()`. See `?loess` for which type of kernel etc. is used. Using `lm()` and doing the weights with your own kernel function enables more models to be fitted, which is especially useful for later computer exercises.
Matlab: The function in `regsmooth1D.m` does one-dimensional local polynomial regression.

Part 3

Use the cumulative conditional means technique in connection with the chosen SETAR model in part 2, and compare with the theoretical cumulative conditional mean. Optionally, explore the asymptotic behavior of the test described in the book on page 70.

Hints
R and Matlab: The scripts in `cumulativeMeans.R` and `cumulativeMeans.m` are implementations of the cumulative conditional means technique.

Part 4

Choose a model that can be written as:

$$y_t = \mu + g(x_{t-1})y_{t-1} + \epsilon_t$$

where x_t is the input, y_t is the output, ϵ_t is white noise and $g(\cdot)$ is a continuously differentiable function.

Simulate 3000 values of the process. Use these simulated values to estimate the dependence of y_t on x_{t-1} and y_{t-1} using the method of local regression.

Try also using conditional parametric models to estimate the same dependence.

Hints
R: In `3dPlotting.R` local polynomial regression is carried out, both using `loess()` and `lm()`. Conditional parametric models can be fitted with either: using `loess()` the parameter `parametric` needs to be altered and using `lm()` the way the weights are calculated needs to be altered.
Matlab: The function in `regsmooth2D.m` fits either a local polynomial regression model or a conditional parametric model. Look into it to learn what is done differently for the two modelling techniques.

Part 5

Choose at least one of the models used in the previous parts and compare the auto correlation function (ACF) with the lag dependence function (LDF). It is recommended to choose a model where essentially no linear correlation is found, for example a SETAR(2;1;1) where the two slopes are of opposite signs.

Hints
R: The script `ldf.R` is a way to estimate lagged dependent functions.
Matlab: The script in `ldf.m` does an estimation of lagged dependent functions. The script in `ldfone.m` can be used to see the estimated dependence function for some lag.