Time series Analysis & Modeling

DATS 6450

LAB # 10- Levenberg Marquardt (LM) Algorithm

The main purpose of this LAB is to implement the maximum likelihood estimation algorithm called (Levenberg Marquardt) using Python program to estimate the coefficient of ARMA model. We are using simulated data for this lab with known parameters to verify accuracy of the developed code. You have already checked the accuracy of your GPAC code to estimate the order of ARMA model using simulated data in the previous LAB. You will do the same for this lab but to estimate the parameter of ARMA model.

You can verify the accuracy of the programmed LM algorithm, since you know what the true coefficients are.

1- Develop a python program that implement LM algorithm and estimate the coefficients of the following ARMA model. The pseudo code for the LM algorithm is given in lecture #10. The true coefficient is -0.5 and the estimated parameter should coverage to the true coefficient.

$$y(t) - 0.5y(t-1) = e(t)$$

2- Data generation: Your code should be written for the general case that asks a user the following information: Hint: mean(y) needs to be subtracted from the y(t) for non-zero mean white noise.
   a. Enter number of samples:
   b. Enter mean of white noise:
   c. Enter variance of white noise:
   d. Enter AR order:
   e. Enter MA order:
   f. Enter coefficients of AR:
   g. Enter coefficients of MA:
3- Split the data into train and test set (80%-20%). Use the train set for parameter estimations.
4- Perform one-step ahead prediction and plot the train test versus the predicted values.
5- Derive the residual errors and calculate Q for 20 lags. Are the residuals white? You need to perform a chi-square whiteness test.
6- Display the confidence interval for the estimated parameter(s). Are all the coefficients statistically important? Justify your answer.
7- Display the estimated covariance matrix of the estimated parameters.
8- Display the estimated variance of error.
9- Display poles and zeros (roots of numerator and roots of denominator). Are there any zero/pole cancellation(s)? Justify your answer.
10- Plot the sum square error of versus the # iterations.
11- Perform a multi-step ahead prediction and compare it versus the test set. Compare the variance of prediction error versus the variance of forecast error. Write down your observations about the accuracy of this prediction.
12- Repeat above steps for the following 7 examples

Example 2: ARMA (0,1): $y(t) = e(t) + 0.5e(t-1)$

Example 3: ARMA (1,1): $y(t) + 0.5y(t-1) = e(t) + 0.25e(t-1)$

Example 4: ARMA (2,0): $y(t) + 0.5y(t-1) + 0.2y(t-2) = e(t)$

Example 5: ARMA (2,1): $y(t) + 0.5y(t-1) + 0.2y(t-2) = e(t) - 0.5e(t-1)$

Example 6: ARMA (1,2): $y(t) + 0.5y(t-1) = e(t) + 0.5e(t-1) - 0.4e(t-2)$

Example 7: ARMA (0,2): $y(t) = e(t) + 0.5e(t-1) - 0.4e(t-2)$

Example 8: ARMA (2,2): $y(t)+0.5y(t-1) +0.2y(t-2) = e(t)+0.5e(t-1) - 0.4e(t-2)$

Upload the formal **solution report (as a single pdf**) plus **the .py file(s)** through BB by the due date.