Time series Analysis & Modeling

DATS 6313

HW # 5

SARIMA

The main purpose of this LAB is to simulate the SARIMA model covered in the lecture using Python program and plot the ACF/PACF of the various SARIMA models. The dataset generation can be done by a Python package.

1- Develop a python code that asks a user to enter the following.
   a. *Enter the number of data samples [1000]: ______________*
   b. *Enter the mean of white noise [0]: _____________*
   c. *Enter the variance of the white noise [1]: _____________*
   d. *Enter AR order: ______________*
   e. *Enter MA order: _____________*
   f. *Enter the coefficients of AR (you need to include a hint how this should be entered):____*
   g. *Enter the coefficients of MA (you need to include a hint how this should be entered):____*
2- Convert the process in example 1-10 from the shift operator to the time difference equation. For example, the first example should be written as:

$$y(t) - 0.5y(t-3) = e(t)$$

3- Simulate the SARIMA model in 1 (bellow) using the scipy package and signal.dlsim with the white noise provided in the previous step.
4- Plot the ACF and PACF of the generated dataset. Use the statsmodels.graphics.tsaplots and subplot 2x1. You can use the following function for the ACF and PACF plots

```
from statsmodels.graphics.tsaplots import plot_acf , plot_pacf
def ACF_PACF_Plot(y,lags):
    acf = sm.tsa.stattools.acf(y, nlags=lags)
    pacf = sm.tsa.stattools.pacf(y, nlags=lags)
    fig = plt.figure()
    plt.subplot(211)
    plt.title('ACF/PACF of the raw data')
    plot_acf(y, ax=plt.gca(), lags=lags)
    plt.subplot(212)
    plot_pacf(y, ax=plt.gca(), lags=lags)
    fig.tight_layout(pad=3)
    plt.show()
```

5- Perform an ADF-test on the generated dataset and plot the rolling mean and variance versus time (hint: like LAB#1). Is the raw dataset stationary?
6- If the raw dataset is non-stationary, then apply the seasonal or non-seasonal differencing (in this case finding the correct seasonality is easy as the process is known) and plot the ACF/PACF of the seasonally/non-seasonally differenced dataset. Use the statsmodels.graphics.tsaplots and

subplot 2x1 (see the provided function in 3). What is the difference between the ACF/PACF of the original dataset and the ACF/PACF of the differenced dataset? Hint: If the generated dataset in step 2 is stationary then there is no need for differencing and this step needs to be skipped.

7- Perform an ADF-test on the differenced dataset and plot the rolling mean and variance versus time (hint: like LAB#1). Is the differenced dataset stationary? Hint: This step needs to be skipped if the raw data is stationary.
8- If the raw dataset is stationary, then plot the GPAC (the row and column should be picked based on the model order) of the raw dataset. If the differenced dataset is stationary, then plot then plot the GPAC of the differenced dataset. Highlight the pattern on the GPAC table. Do you see the correct order on the GPAC? Prove your answer.
9- Plot the first 500 samples of raw dataset and differenced dataset(if exists) in one graph. Add appropriate title, legend, x-axis, and y-axis label.
10- Write down your observation about the ACF/PACF plot of the SARIMA model and the seasonally/non-seasonally differenced dataset.
11- Apply step 2- 10 on the following examples 2 through 10.

Question 12-15 only needs to be answered for Example 2 only

12- Without use of the Python software, manually develop the forecast function for one-step, two step ,…, 50 step prediction. You should have multiple different equations for each prediction.
13- Using python plot the train test versus the 1-step ahead prediction for the first 100 samples.
14- Using python Perform the residual analysis and show that the residual errors [train – one step prediction] is white. Consider alpha = 0.01 and number of lags = 20.
15- Simulate the forecast function for the 50-step prediction. Print the variance of the test set versus the variance of the predicted values. Run the code multiple times until this ratio become close to 1. Then plot the 50 step predicted values versus the test set in one graph. Add an appropriate title, legend , x and y axis label. Hint : The time is fixed at t = 950 and 50-step prediction is made from t = 950.

1: $ARIMA(1,0,0)_3$:$(1-0.5q^{-3})y(t)=\epsilon_t$

2: $ARIMA(2,0,0)_3$:$(1-0.5q^{-3}+0.6q^{-6})y(t)=\epsilon_t$

3: $ARIMA(1,1,0)_3$:$(1-0.5q^{-3})(1-q^{-3})y(t)=\epsilon_t$

4: $ARIMA(2,1,0)_3$:$(1-0.5q^{-3}+0.6q^{-6})(1-q^{-3})y(t)=\epsilon_t$

5: $ARIMA(0,0,1)_4$:$y(t)=(1-0.9q^{-4})\epsilon_t$

6: $ARIMA(0,0,2)_4$:$y(t)=(1-0.2q^{-4}-0.8q^{-8})\epsilon_t$

7: $ARIMA(0,1,1)_4$:$(1-q^{-4})y(t)=(1-0.9q^{-4})\epsilon_t$

8: $ARIMA(0,1,2)_4$:$(1-q^{-4})y(t)=(1-0.5q^{-4}+0.6q^{-8})\epsilon_t$

9: $ARIMA(1,1,1)_4$:$(1-0.5q^{-4})(1-q^{-4})y(t)=(1-0.2q^{-4})\epsilon_t$

10: $ARIMA(0,2,1)\times ARIMA(0,1,1)_3$:$(1-q^{-1})^2(1-q^{-3})y(t)=(1-0.2q^{-1})(1+0.35q^{-3})\epsilon_t$

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