1 Introduction

All online documents/notebooks are available. Please find here below a summary of available resources :

- Course notebooks
- Theory
- MVA time series course
- Dauphine course
- Fastai notebooks
- Fastai book
- Fourier analysis

2 Theoretical questions

2.1 [3 points] Estimating an auto correlation at step h

2.1.1 Question

Let X_t be a weakly stationary time series and let's assume that we have made three observations from this series i,e, n = 3. Let X_1= 5, X_2 = 3 and X_3 = 4. Give the theoretical definition of the auto correlation function $\gamma_X(h)$ and the statistical estimators associated. Calculate $\hat{\gamma}_X(1)$.

2.1.2 Correction

$$\hat{\gamma}_X(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_t - \bar{x})(x_{t+h} - \bar{x})$$

By replacing h by 1, it comes :

$$\hat{\gamma}_X(1) = \frac{1}{n}\sum_{t=1}^{n-1}(x_t - \bar{x})(x_{t+1} - \bar{x})$$

For n=3 and h=1, it comes :

$$\hat{\gamma}_X(1) = \frac{1}{3}\sum_{t=1}^{2}(x_t - \bar{x})(x_{t+1} - \bar{x})$$

Hence the auto-correlation value for a distance of 1.

$$\hat{\gamma}_X(1) = \frac{1}{3}((5-4)*(3-4)+(3-4)*(4-4)) = -\frac{1}{3}$$

2.2 [7 points] AR(1) process

Consider the time series model given by $X_t = 0.5 + 0.75 * X_{t-1} + w_t$ where w_t are i.i.d. $\mathcal{N}(0, \sigma^2)$. You may assume that this process is weakly stationary.

2.2.1 Find a one-step ahead forecast

Question:
If X_t=1.5, find a one step ahead forecast (clue : compute the conditional expectation $E\left[X_{t+1}|X_t\right]$).
Correction:

$$E\left[X_{t+1}|X_t\right] = E\left[0.5 + 0.75 * X_t + w_t|X_t\right]$$

Applying the conditional expectation properties, it comes :

$$E\left[X_{t+1}|X_t\right] = 0.5+0.75*X_t+E\left[w_t|X_t\right] = 0.5+0.75*X_t+E\left[w_t\right] = 0.5+0.75*X_t = 1,625$$

2.2.2 Find a l-step ahead forecast

Question:
By recursively developing $X_t = 0.5 + 0.75 * X_{t-1} + w_t$, find a l-step forecast closed formula for $E\left[X_{t+l}|X_t\right]$ (clue : you will have to use the summation formula for a geometric series). Justify why you can forget the product terms with w_t when taking the expectation.
Correction:
Let's note

$$E\left[X_{t+l}|X_t\right] = X_{t+l}^t$$

w_t being i.i.d and independent of any X_t, it comes :

$$E\left[w_{t+l}|X_t\right] = 0, \forall l \geq 1$$

We can therefore forget the w_t when taking the expectation.
By applying recursively the AR(1) equation it comes:

$$X_{t+2}^t = 0.5 + 0.75 * X_{t+1}^t = 0.5 + 0.75 * (0.5 + 0.75 * X_t)$$

$$X_{t+3}^t = 0.5 + 0.75 * X_{t+2}^t = 0.5 + 0.75 * (0.5 + 0.75 * (0.5 + 0.75 * X_t))$$

$$X_{t+3}^t = 0.5 + 0.5 * 0.75 + 0.5 * 75^2 + 0.75^3 * X_t$$

By recurrence, we can show that :

$$X_{t+l}^t = 0.5 + 0.5 * 0.75 + 0.5 * 75^2 + ... + 0.5 * 75^{l-1} + 0.75^l * X_t$$

By applying the geometric serie summation formula :

$$X_{t+l}^t = \frac{0.5 * (1 - 0.75^l)}{1 - 0.75} + 0.75^l * X_t = 2 * (1 - 0.75^l) + 0.75^l * X_t$$

It comes for l=10 and $X_t = 1.5$:

$$X_{t+10}^t = 2*(1-0.75^10)+0.75^10*X_t = 2*(1-0.75^10)+0.75^10*1.5 = 1.97184$$

2.2.3 Process expectation

Question:
Calculate $E(X_t)$ for this process (clue : the process is stationary)
Does $E[X_{t+l}|X_t]$ converge to $E[X_t]$
Correction:
The process is stationary, hence:

$$E[X_t] = \mu, \forall t$$

By taking the expectation of $X_t = 0.5 + 0.75 * X_{t-1} + w_t$, it comes :

$$E[X_t] = 0.5 + 0.75 * E[X_{t-1}] + E[w_t]$$

which translates to as the process is stationary of expectation μ:

$$\mu = 0.5 + 0.75 * \mu + 0$$

It comes

$$\mu = 0.5/(1 - 0.75) = 2$$

2.3 3) [10 points] Digression around stationarity

Question:
Give the definition of weak and strong stationarity. Digress around the concept of extrapolation and how the need for stationarity for proper univariate time series modeling can be derived from it when applying machine learning techniques. Use the appropriate course notebooks to highlight your digressions about extrapolation and the need for differencing before applying a ML model.
Correction:
Strong stationarity : The joint law of any $(X_{t1}, .., X_{tn})$ is the same as any of its translated tuples $(X_{t1+T}, .., X_{tn+T})$
Weak stationarity : $E[X_t] = \mu, \forall t$ and $V[X_t] = \sigma, \forall t$
(mean and variance are constant over time)

Basically, the student must refind and comment briefly each cell of the following the notebooks:

- Extrapolation
- ML without differentiating
- ML with differentiating

3 Coding questions

3.1 4) [10 points] Cointegration simulation and estimation

3.1.1 Question

a) Simulate a white noise X0 of size N using numpy.random.randn(). Plot the time series b) Use numpy.cumsum() to create X1 from X0. Prove that X1 is a process integrated of order 1 (first differentiation stationary) by using the adf() from the course notebook Stationarity c) Recreate Y0 and Y1 by following the same methodology d) Simulate a new variable $Z1_t = 1.5 * X1_t + \epsilon_t$ where ϵ_t are i.i.d and follow a gaussian normal distribution of mean 0 and variance 1. e) Apply the 2 steps Engle Granger methodology (by coding yourself the regression part) to prove that Z1 and X1 are correlated, but not Y1 and X1.

3.1.2 Correction

The notebook with the written correction can be found here: Correction

3.2 5) [10 points] Machine learning advanced framework

3.2.1 a) Creating a prepocessing function for modular predictor creation

In the notebook Machine learning with differencing, devise a function to preprocess the dataset and separate it in by predictors X and the dependent output Y to predict.
The function will look like:
column_name='Passengers'
Nval = 12
T=10
ml_model = LinearRegression()
validation_mape_loss =validate_features(df,column_name=column_name, Nval = Nval, T=T, ml_model=ml_model)
This function will plot the training and test set prediction and give back the loss on the forward validation set.
This exercice is just a refactoring one : everything is already available here Machine learning with differencing.

3.2.2 b) Features engineering

Modify now that function to incorporate 'engineered features' (we will add for instance the past average and the max and min of the lagged predictors up to lag l.

3.2.3 c) Assessment of the new model

By using your newly created function, discuss the performance of your model with added features compared to the ones without the added features (the one already assessed in the notebook.
Your code will look like:
column_name='Passengers'
Nval = 12
T=10
ml_model = LinearRegression()
for val_features in [True, False]:
print(f'validating features val_features')
test_mape_loss =validate_features(df,column_name=column_name, Nval = Nval, T=T,

ml_model=ml_model, add_features=val_features)

3.2.4 Correction

The notebook with the written correction can be found here: Correction