Processo Autoregressivo
UFRGS - FIS01082

Aluno: Henrique Alexandre Boneto - 288744

Maio 2020

Processos Autoregressivos

Um modelo autorregressivo de ordem p denotado por AR(p) é um modelo que tenta predizer o futuro baseado em dados passados. É definido como:

$$x_t = c + \sum_{i=1}^{p} \phi_i x_{t-i} + \epsilon_t$$

Sendo $\phi_1, ..., \phi_p$ os parâmetros de ajuste do modelo, e ϵ_t é um processo aleatório puro com média zero (ruído branco gaussiano).

Processos Estacionários

Um processo estacionário é aquele em que suas propriedades não dependem no tempo. Por exemplo, dado uma série de dados, a distruibuição de probabilidade dessa série não irá mudar em decorrer do tempo, e por consequência sua média, variância, etc. também não irão mudar.

Teste da Raíz Unitária

Para que um processo autoregressivo tenha um processo estacionário é preciso que a raiz unitária respeite a seguinte condição para que a variança da série convirja, como demonstrado em aula pelo professor:

$$|a| < 1$$

Dickey-Fuller

O teste de Dickey-Fuller testa se existe raiz unitária em um modelo autoregressivo. Como também demonstrado pelo professor, o valor de β deve ser menor do que 0 para que a série de dados seja estacionária, já que $\beta = (a - 1)$, obtido a partir de uma regressão linear tal que:

$$\Delta X_t = \beta X_{t-1} + \epsilon_t$$

Equações de Yule-Walker

As equações de Yule-Walker são obtidas a partir do modelo autoregressivo multiplicando as equaçãos pelos atrasos $X_t, X_{t-1}, ..., X_{t-p-1}$ e depois aplicando o valor esperado, ficando com:

$$r_p = \sum_{j=1}^{p} \phi_j r_{p-j}$$

Ou, na forma matricial:

$$\mathbf{R} = \mathbf{\Phi r}$$

Podendo ser resolvida:

$$\mathbf{\Phi} = \mathbf{R^{-1} r}$$

Onde **R** é a matriz de Toeplitz das autocorrelações e $\mathbf{\Phi}$ são os parâmetros a encontrar do modelo.

Autocorrelação Parcial

A autocorrelação parcial entre duas variáveis é a correlação que sobra caso todas as outras variáveis fossem removidas [**2**]. Ela será usada pois a função autocorrelação de um modelo autorregressivo se torna zero, mas nunca decaem abruptamente [**2**].
Por isso, as autocorrelações parciais dão a possibilidade de encontrarmos a ordem de nosso modelo autoregressivo, por exemplo, dado um AR de ordem **2**, as autocorrelações parciais para um t >2 geralmente serão 0 ou muito próximas.
Para um modelo autoregressivo de ordem p elas são calculadas através da regra de Cramer sobre as equações de Yule-Walker resolvendo para $\phi_{11}, \phi_{22}, ..., \phi_{kk}$.

Prática

Série de Dados

Os dados escolhidos para esse trabalho são de um dos índices do mercado financeiro mais famosos do mundo, SP 500 (Standard Poor's 500), entre o período de 2019-01-02 até 2020-01-30, de frequência diária. É composto por quinhetos ativos presentes na bolsa americana (por exemplo Microsoft, Intel, Google...) e é considerado um indicador importante da economia geral dos Estados Unidos. Os dados foram retirados do site Yahoo Finance (`https://finance.yahoo.com`).
Abaixo pode-se ver uma amostra dos dados (esses dados são chamados de OHLCV, pois possuem informações da abertura do preço, preço mais alto, baixo e de fechamento, para um período de tempo, assim como a informação de volume financeiro), com a coluna de retornos "returns" já calculada:

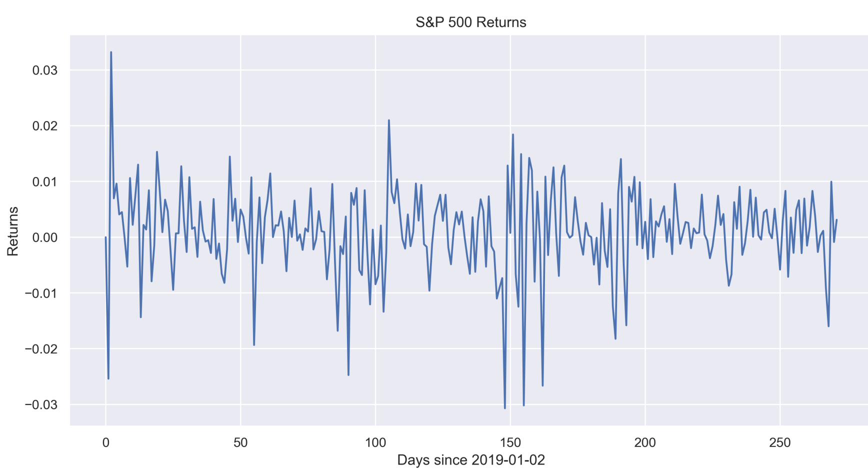
	date	open	high	low	close	adj_close	volume	returns
0	2019-01-02	2476.959961	2519.489990	2467.469971	2510.030029	2510.030029	3733160000	0.000000
1	2019-01-03	2491.919922	2493.139893	2443.959961	2447.889893	2447.889893	3822860000	-0.025385
2	2019-01-04	2474.330078	2538.070068	2474.330078	2531.939941	2531.939941	4213410000	0.033196
3	2019-01-07	2535.610107	2566.159912	2524.560059	2549.689941	2549.689941	4104710000	0.006962
4	2019-01-08	2568.110107	2579.820068	2547.560059	2574.409912	2574.409912	4083030000	0.009602
...	...	...	...	...	...	...	...	...
267	2020-01-24	3333.100098	3333.179932	3281.530029	3295.469971	3295.469971	3707130000	-0.009125
268	2020-01-27	3247.159912	3258.850098	3234.500000	3243.629883	3243.629883	3823100000	-0.015982
269	2020-01-28	3255.350098	3285.780029	3253.219971	3276.239990	3276.239990	3526720000	0.009954
270	2020-01-29	3289.459961	3293.469971	3271.889893	3273.399902	3273.399902	3584500000	-0.000868
271	2020-01-30	3256.449951	3285.909912	3242.800049	3283.659912	3283.659912	3787250000	0.003125

272 rows × 8 columns

*Os retornos financeiros foram calculados com base nos preços de fechamento (coluna ”close”), utilizando a seguinte fórmula:

$$r_t = \frac{pc_t - pc_{t-1}}{pc_t}$$

Onde r_t é o retorno no tempo t, pc_t é o preço de fechamento no tempo t e pc_{t-1} é o preço de fechamento no tempo (t-1).
Abaixo o gráfico de retornos gerado:



Abordagem

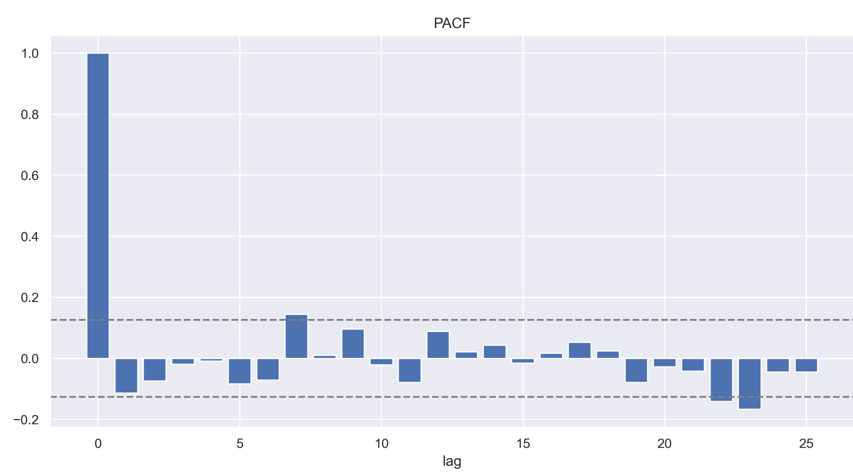
Para este trabalho, pretende-se fazer a predição dos retornos financeiros diários do índice ao decorrer do tempo usando um modelo autoregressivo utilizando a linguagem de programção Python. Para isso, primeiramente foram definidos dois subconjuntos de dados: um para "treinar" (df_train) o modelo (obter os coeficientes), constituído de um período de 2019-01-02 até 2020-01-28 e outro para "testar" (df_test) com os resultados obtidos, constituído pelos últimos 3 dias 2020-01-29 até 2020-01-31.

Estacionariedade

Após isso, para verificar a estacionariedade dos retornos sobre os dados de treino, usou-se o teste da raiz unitária, utilizando uma regressão linear para obter o coeficiente angular da curva. O resultado obtido foi de aproximadamente -4.32×10^{-6}, concordando com o esperado para um processo estacionário.
Além disso, também é possível verificar que a média dos retornos financeiros foi de 0.0009 não mudam com o tempo (sempre muito próximas de zero), assim como sua distribuição de probabilidade tendo comportamento de calda longa.

PACF

Uma função foi implementada (pode ser vista na seção "Implementação" na função "acf_pacf"), através das equações de Yule-Walker, para obter as autocorrelações parciais da série de dados para estimar a ordem do modelo a ser utilizada. Os resultados obtidos sobre "df_train" está abaixo, com as linhas horizontais calculadas como $\pm 2/\sqrt{N}$ que demonstram um intervalo de confiança de 95% **[2]**(usado para julgar a importância estatística dos coeficientes):

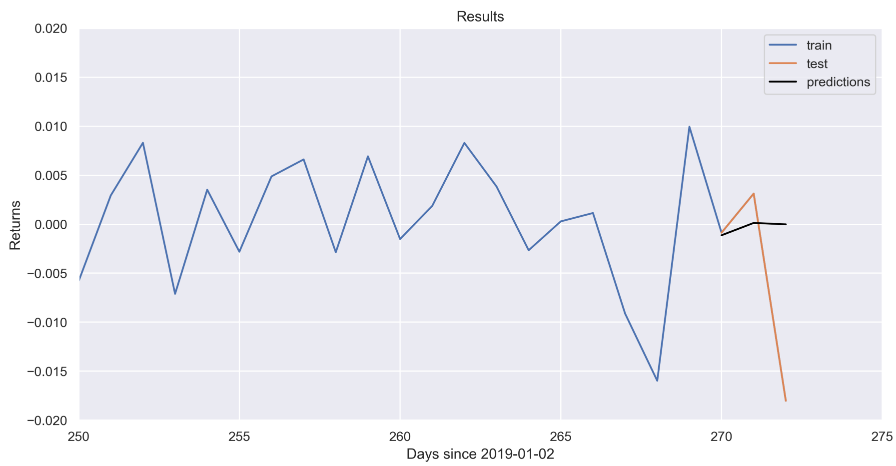


Através do gráfico é possível perceber que somente a primeira ordem de atraso chega mais perto do intervalo de confiança, e os próximos decaem rapidamente.

Desse modo foi escolhido um modelo autoregressivo de ordem 1 $AR(1)$, entretanto espera-se que os resultados não sejam condizentes com a realidade, visto que é um modelo simples tentando modelar algo complexo como o mercado financeiro. Vale também lembrar que muitas pessoas tomam atitudes dentro do mercado baseados nos acontecimentos do dia anterior,

Resultados

Abaixo é possível ver os resultados obtidos:



Implementação

Abaixo o código feito em Python:

```
#author: Henrique Boneto

#importing libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.linear_model import LinearRegression
from scipy.linalg import toeplitz
sns.set(style='darkgrid')

#configure data
df = pd.read_csv('data/S&P500.csv')
df.columns = ['date', 'open', 'high', 'low', 'close', 'adj_close', 'volume']
df['returns'] = ((df['close'] - df['close'].shift(1)) / df['close']).fillna(0)
```

```
#define train and test data
df_train = df[df['date'] < '2020-01-28']
df_test = df[df['date'] >= '2020-01-28']

#plotting returns
def plot_returns():
    fig, ax = plt.subplots(figsize=(12,6))
    plt.title('S&P 500 Returns')
    plt.xlabel('Days since 2019-01-02')
    plt.ylabel('Returns')
    ax.plot(df.index, df['returns'])
    plt.savefig('figs/S&P500.png', dpi=300)
plot_returns()

#stationarity
def check_stationarity(df):
    model = LinearRegression()
    x = np.array(df.index).reshape((-1,1))
    y = np.array(df['returns'])
    model.fit(x, y)
    #slope
    print('Slope:', model.coef_)

check_stationarity(df_train)

#implementation of ACF and PACF
def acf_pacf(df, lags=25):
    x = list(df['returns'])
    mean = np.mean(x)
    gama_0 = np.sum((x-mean)**2)
    auto_corr = []

    for i in range(lags+1):
        gama_t = 0
        for t in range(len(x) - i):
            gama_t += ((x[t] - mean) * (x[t+i] - mean)) / gama_0
        auto_corr.append(gama_t)

    R = toeplitz(auto_corr[:len(auto_corr)-1])
    alpha = auto_corr[1:]
    partial_auto_corr = np.linalg.solve(R, alpha)

    return(auto_corr, partial_auto_corr)

acf, pacf = acf_pacf(df_train)
```

```
#implementation of AR
def AR(df, order, n_pred, coefs):
    x = list(df['returns'])
    predictions = []

    for i in range(n_pred):
        pred = np.dot(x[-order:], coefs[:order])
        predictions.append(pred)
        x.append(pred)

    return(predictions)

#plotting PACF
def plot_pacf(df, acf, pacf):
    plt.figure(figsize=(12,6))
    plt.title('PACF')
    plt.xlabel('lag')

    #confidence interval
    y = 2/np.sqrt(df.shape[0])
    plt.axhline(y=y, color='gray', linestyle='--')
    plt.axhline(y=-y, color='gray', linestyle='--')

    #PACF
    plt.bar([0,1], [1,pacf[0]], color='b')
    plt.bar(np.arange(1, len(pacf)+1), pacf, color='b')
    plt.savefig('figs/pacf_S&P500.png', dpi=300)

plot_pacf(df_train, acf, pacf)

#model params choice
n_pred = df_test.shape[0]
order = 1

#make predictions
pred = AR(df_train, order, n_pred, pacf)

#plot results
plt.figure(figsize=(12,6))
plt.title('Results')
plt.xlabel('Days since 2019-01-02')
plt.ylabel('Returns')

plt.plot(df['returns'], label='train')
plt.plot(df_test['returns'], label='test')
```

```
plt.plot(df_test.index, pred, label='predictions', c='black')

plt.xlim([250,275])
plt.ylim([-0.02, 0.02])
plt.legend()
plt.savefig('figs/ar_results_S&P500.png', dpi=300)
```

Referências

[1] En.wikipedia.org. 2020. Autoregressive Model. [online] Available at: https://en.wikipedia.org/wiki/Autoregressive_model [Accessed 29 May 2020].

[2] Kirchgassner, G. and Wolters, J., 2010. Introduction To Modern Time Series Analysis. Berlin: Springer.

[3] En.wikipedia.org. 2020. S0026p 500 Index. [online] Available at: https://en.wikipedia.org/wiki/S%26P_500_Index [Accessed 30 May 2020].

[4] Otexts.com. 2020. 8.1 Stationarity And Differencing — Forecasting: Principles And Practice. [online] Available at: https://otexts.com/fpp2/stationarity.html [Accessed 29 May 2020].