Written Report – 6.419x Module 4

Name: Sandipan Dey

2. The Mauna Loa CO_2 Concentration

Include your answers to the following questions in your written report.

The final model

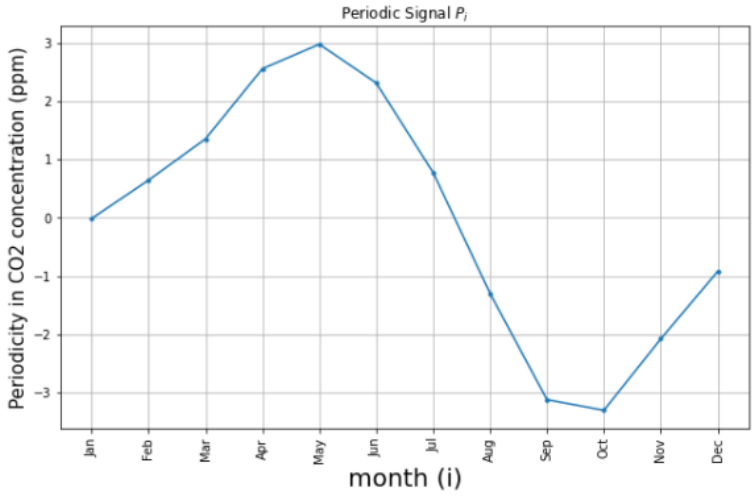
1. (3 points) Plot the periodic signal P_i. (Your plot should have 1 data point for each month, so 12 in total.) Clearly state the definition the P_i, and make sure your plot is clearly labelled.

Solution

The deterministic trend $F_n(t)$ is removed from the time series and the average residual $C_i - F_n(t_i)$ is computed for each month (to get one data point for each month). The collection of these points can be interpolated to form a periodic signal P_i, as shown in the next figure.

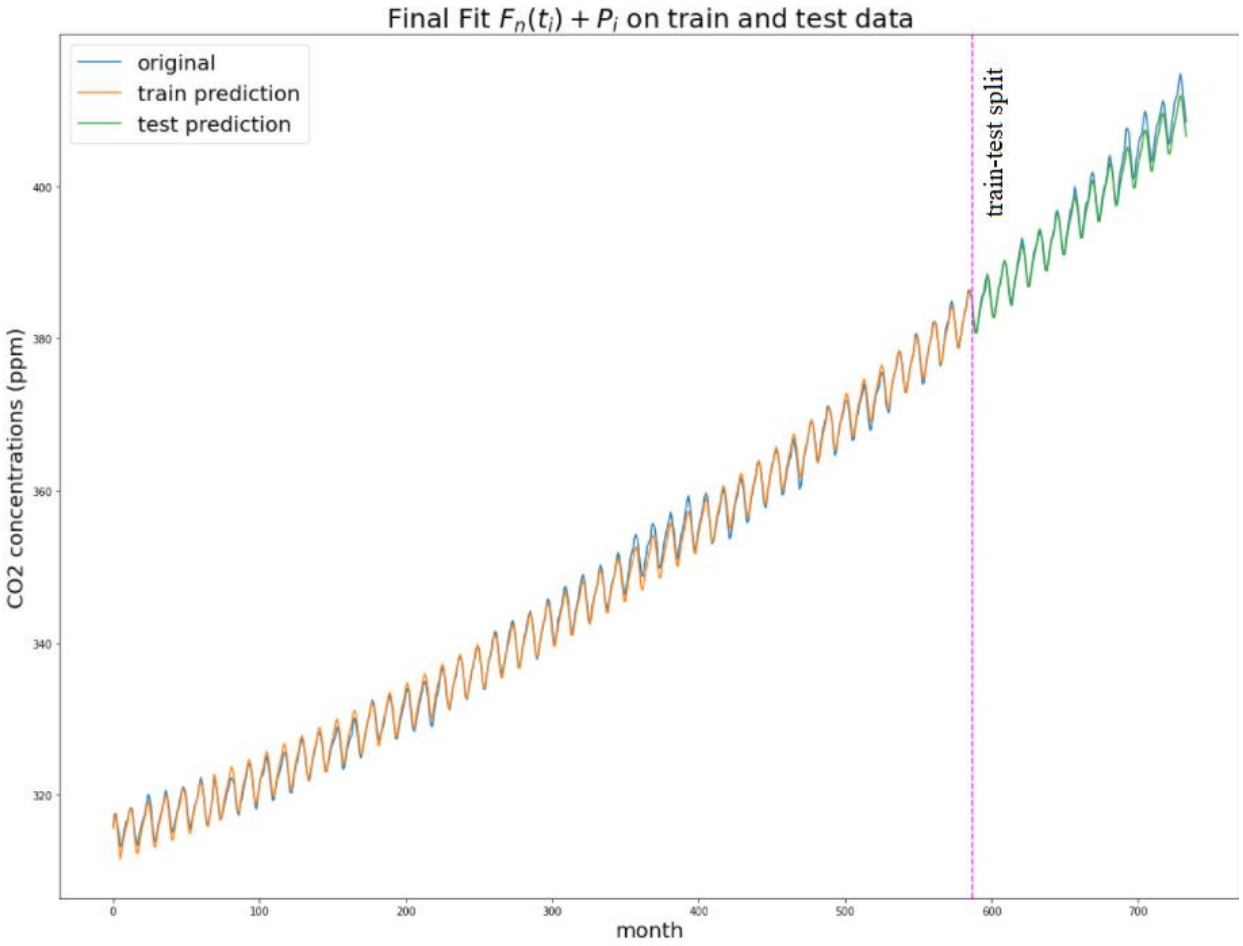
```
df_train['p'] = df_train.C - df_train.trend
p = df_train.groupby('month')['p'].mean()
p
```

```
month
1    -0.012919
2     0.646407
3     1.355569
4     2.561858
5     2.982891
6     2.316473
7     0.776297
8    -1.301213
9    -3.128074
10   -3.309520
11   -2.081487
12   -0.921507
Name: p, dtype: float64
```



2. (2 points) Plot the final fit $F_n(t_i) + P_i$. Your plot should clearly show the final model on top of the entire time series, while indicating the split between the training and testing data.

Solution



plot obtained from the following data-frame, with the *trend* ($F_n(t_i)$) and *seas* (periodicity P_i) columns computed as shown below.

	year	month	C	t	t2	trend	seas	pred
2	1958	3	315.70	0.208333	0.043403	314.268214	1.355569	315.623783
3	1958	4	317.45	0.291667	0.085069	314.335560	2.561858	316.897418
4	1958	5	317.51	0.375000	0.140625	314.403075	2.982891	317.385965
6	1958	7	315.86	0.541667	0.293403	314.538609	0.776297	315.314906
7	1958	8	314.93	0.625000	0.390625	314.606628	-1.301213	313.305416

3. (4 points) Report the root mean squared prediction error RMSE and the mean absolute percentage error MAPE with respect to the test set for this final model. Is

this an improvement over the previous model $F_n(t_i)$ without the periodic signal? (*Maximum 200 words*.)

Solution

RMSE and MAPE this time is 1.14936 and 0.0020859 as shown below

```
mse(df[ntrain:]['C'], df[ntrain:]['pred'], squared=False), mape(df[ntrain:]['C'], df[ntrain:]['pred'])
```

```
(1.1493602690794402, 0.0020859165947991)
```

which is an improvement over the previous values shown below

RMSE = 2.501332219489784

As a percentage, MAPE = 0.5320319129740852 %

4. (3 points) What is the ratio of the range of values of F to the amplitude of P_i and the ratio of the amplitude of P to the range of the residual R_i (from removing both the trend and the periodic signal)? Is this decomposition of the variation of the CO concentration meaningful? (*Maximum 200 words*.)

Solution

Range of values of F = 95.47089096434252
Amplitude of P_i = 3 (from the figure in Q_1)
Range of R_i = 4.716444703651803

Hence,

- The ratio of the range of values of F to the amplitude of P_i = 31.823630321447506
- The ratio of the amplitude of P to the range of the residual R_i = 0.636072335943468

It's meaningful since most of the variance is captured by the trend and small fraction of it by seasonality (periodicity).

3. Autocovariance Functions (Written Report)

Include your answer to this question in your written report.

1. (4 points) Consider the $\mathrm{MA}(1)$ model,

$$X_t = W_t + \theta W_{t-1},$$

where $\{W_t\} \sim W \sim \mathcal{N}(0, \sigma^2)$. Find the autocovariance function of $\{X_t\}$.

Include all important steps of your computations in your report.

Solution

$$\begin{aligned}
&cov(X_t, X_t) \\
&= cov(W_t + \theta W_{t-1}, W_t + \theta W_{t-1}) \\
&= var(W_t) + 2\theta cov(W_t, W_{t-1}) + \theta^2 var(W_{t-1}) \\
&= \sigma^2 + 0 + \theta^2\sigma^2 \\
&= (1 + \theta^2)\sigma^2
\end{aligned}$$

$$\begin{aligned}
&cov(X_t, X_{t-1}) \\
&= cov(W_t + \theta W_{t-1}, W_{t-1} + \theta W_{t-2}) \\
&= cov(W_t, W_{t-1}) + \theta cov(W_t, W_{t-2}) + \theta var(W_{t-1}) + \theta^2 cov(W_{t-1}, W_{t-2}) \\
&= 0 + 0 + \theta\sigma^2 + 0 \\
&= \theta\sigma^2
\end{aligned}$$

$$cov(X_t, X_{t-k}) = 0, \ \forall k \geq 2$$

Since white noise, we have the following property:

$$\mathrm{cov}(W_s, W_t) = \begin{cases} \sigma^2 & \text{if } s = t \\ 0 & \text{otherwise.} \end{cases}$$

2. (4 points) Consider the $\mathrm{AR}(1)$ model,

$$X_t = \phi X_{t-1} + W_t,$$

where $\{W_t\} \sim W \sim \mathcal{N}(0, \sigma^2)$. Suppose $|\phi| < 1$. Find the autocovariance function of $\{X_t\}$. (You may use, without proving, the fact that $\{X_t\}$ is stationary if $|\phi| < 1$.)

Include all important steps of your computations in your report.

Solution

By the stationarity assumption, we have $E[X_t] = E[X_{t-1}]$,

$\implies E[X_t]$

$= E[\phi X_{t-1} + W_t]$

$= \phi E[X_{t-1}] + E[W_t]$, by the linearity of expectation

$= \phi E[X_{t-1}]$, by the property of white noise, mean being 0

$= E[X_{t-1}]$

$\implies (1 - \phi)E[X_{t-1}] = 0$

$\implies E[X_t] = E[X_{t-1}] = 0$, since $|\phi| < 1$

Again, by the stationarity assumption, we have $Var(X_t) = Var(X_{t-1})$,

$\implies Var(X_t)$

$= Var(\phi X_{t-1} + W_t)$

$= \phi^2 Var(X_{t-1}) + Var(W_t)$, by the independence of white noise error with the signal

$= \phi^2 Var(X_t) + \sigma^2$, by the stationarity assumption

$\implies var(X_t) = \frac{\sigma^2}{1-\phi^2}$, since $|\phi| < 1$

Finally,

autocovariance with lag 1

$= \gamma_1 = cov(X_t, X_{t-1})$

$= cov(\phi X_{t-1} + W_t, X_{t-1})$

$= \phi var(X_{t-1}) + 0$, since white noise is independent with signal

$= \phi var(X_t)$, by stationarity

$= \phi\sigma^2$

autocovariance with lag 2

$= \gamma_2 = cov(X_t, X_{t-2})$

$= cov(\phi X_{t-1} + W_t, X_{t-2})$

$= cov(\phi^2 X_{t-2} + \phi W_{t-1} + W_t, X_{t-2})$

$= \phi^2 var(X_{t-2}) + 0$, since white noise is independent with signal

$= \phi^2 var(X_t)$, by stationarity

$= \phi^2\sigma^2$

and in general, autocovariance with lag h

$= \gamma_h = cov(X_t, X_{t-h})$

$= cov(\phi^h X_{t-h} + \phi W_{t-h+1} + \ldots + \phi W_{t-1} + W_t, X_{t-2})$

$= \phi^h var(X_{t-h}) + 0$, since white noise is independent with signal

$= \phi^h var(X_t)$, by stationarity

$\implies \gamma_h = \phi^h \sigma^2, \forall h \geq 1$

5. Converting to Inflation Rates

Include your answers to this question in your written report.

1. Repeat the model fitting and evaluation procedure from the previous page for the monthly inflation rate computed from CPI.

   Your response should include:

- (1 point) Description of how you compute the monthly inflation rate from CPI and a plot of the monthly inflation rate. (You may choose to work with log of the CPI.)
- (2 points) Description of how the data has been detrended and a plot of the detrended data.
- (3 points) Statement of and justification for the chosen AR(p) model. Include plots and reasoning.
- (3 points) Description of the final model; computation and plots of the 1 month-ahead forecasts for the validation data. In your plot, overlay predictions on top of the data.

Solution

Computing monthly inflation rate

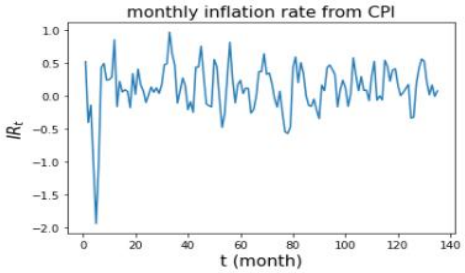
To compute the monthly inflation rate, let's use the following formula:

$$\mathrm{IR}_t = 100(\ln(\mathrm{CPI}_t) - \ln(\mathrm{CPI}_{t-1}))$$

, where t indexes the months and express IR_t in percentages (multiply by 100). The next plot shows how the time series looks like.

```
df['IR'] = np.nan
for i in range(1, len(df)):
    df.loc[i, 'IR'] = 100*(np.log(df.loc[i, 'CPI']) - np.log(df.loc[i-1, 'CPI']))
```

	year	month	date	CPI	t	IR
0	2008	7	2008-07-24	100.00000	0	NaN
1	2008	8	2008-08-01	100.52510	1	0.523726
2	2008	9	2008-09-01	100.12380	2	-0.400003
3	2008	10	2008-10-01	99.98538	3	-0.138344
4	2008	11	2008-11-01	98.97539	4	-1.015274



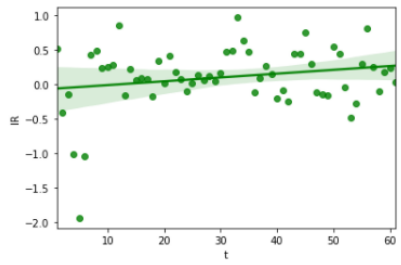
Detrending

As before, split the data into train and test (using the split date as 2013-09-01) and detrend the data into $IR_t = T_t + R_t$ by fitting a linear trend $T_t = \alpha_1 t + \alpha_0$ as before. Fit a linear regression model on the training data to estimate the linear trend, as shown below.

```
reg = LinearRegression().fit(df_train['t'].values.reshape(-1,1), df_train['IR'].values)
reg.intercept_, reg.coef_
```

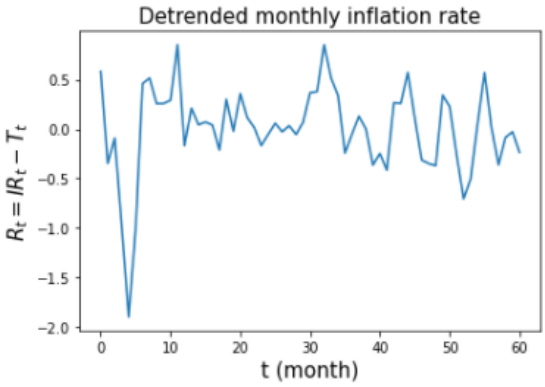
```
(-0.06427097234449808, array([0.00553001]))
```

we have α_0=-0.06427 and α_1=0.00553, as shown in the above result obtained. The following figure shows the linear trend found from the training data.



Subtract the linear trend from the data IR_t to get the residuals R_t. Let's visualize R_t.

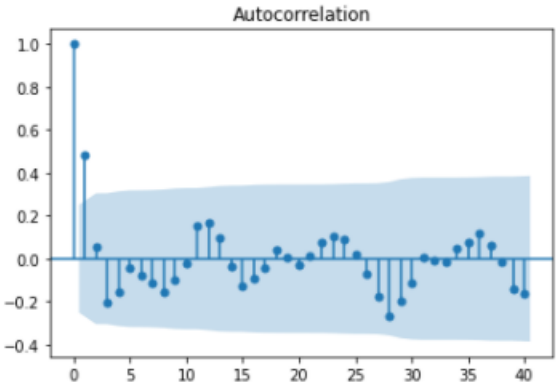
```
pred = reg.predict(df_train['t'].values.reshape(-1,1))
detrended = df_train['IR'].values - pred
```



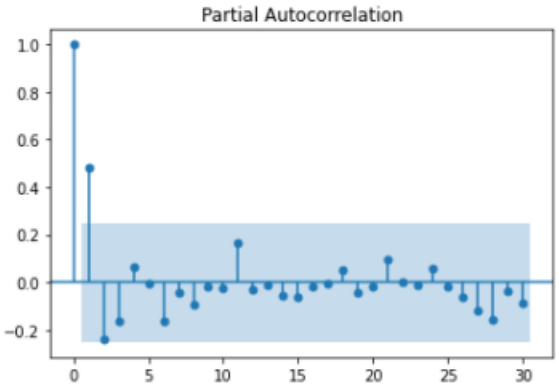
As can be seen from the above plot, there is no obvious further trend or seasonality in the detrended data. For all the analysis done, NA values are dropped whenever required.

Choosing the order p of the AR(p) model

```
import statsmodels.api as sm
sm.graphics.tsa.plot_acf(detrended, lags=40)
```



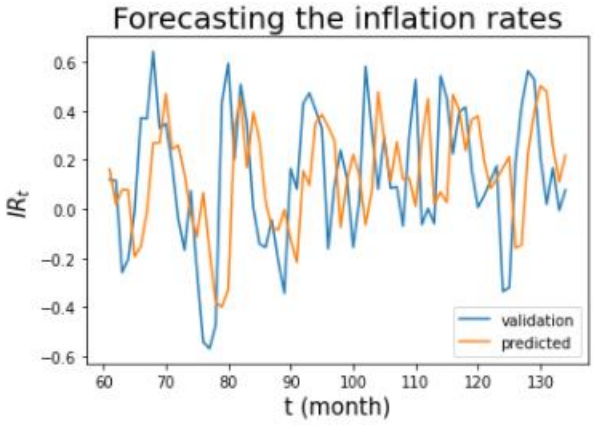

```
sm.graphics.tsa.plot_pacf(detrended, lags=30, method="ywm")
```



As can be seen, from the above PACF plot, the PACF drops sharply and becomes insignificant after p=1. Hence, the order of the AR model can be chosen to be 1.

Final model and forecasts on the validation (test) data

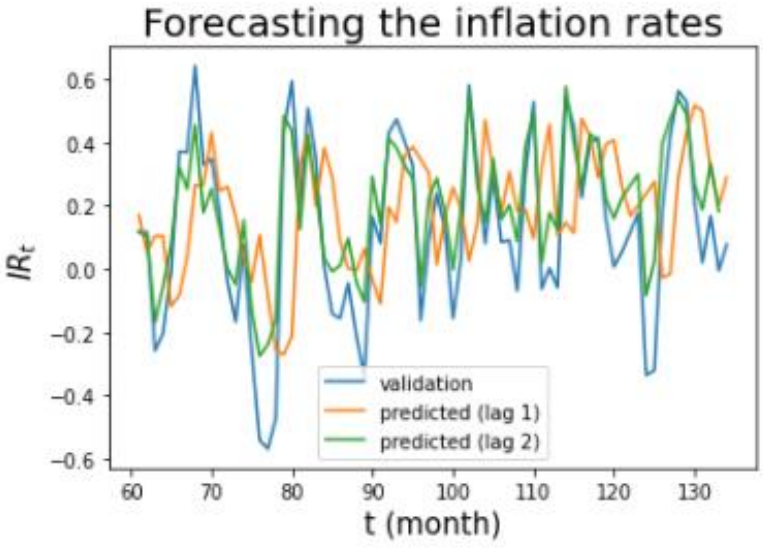
Hence, the final model is IR_t = -0.06427 + 0.00553t + AR(1). The forecast is shown in the below figure, overlayed on top of the original validation data.



2. (3 points) Which AR(p) model gives the best predictions? Include a plot of the RMSE against different lags p for the model.

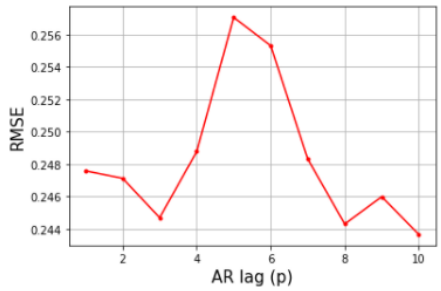
Solution

The next plot shows forecasting with 2 different lags on the validation (test) dataset (September 2013 onward).



The following shows the RMSE values for different values of the order (p) of the AR model fit on the residual data. As can be seen, p=10 gives the best predictions (with the lowest RMSE), then p=8 and p=3 gives the 2nd and 3rd best predictions.

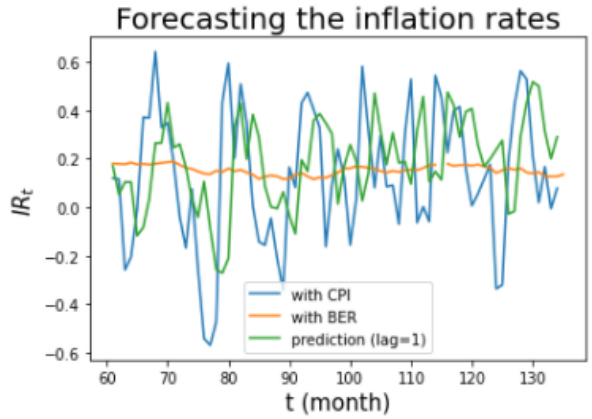
```
lags = range(1,11)
rmses = []
for lag in lags:
    res = AutoReg(detrended, lags = lag).fit()
    preds = []
    for t in range(len(df_train)+lag, len(df1)-2):
        p = reg.predict(np.array([t]).reshape(-1,1))[0]
        preds.append(p + res.params[0] + np.sum(res.params[1:]*detrended[t-lag:t][::-1]))
    rmses.append(mse(df_test['IR'].values[lag:-1], preds, squared=False))
```



Include your answers to this question in your written report.

(3 points) Overlay your estimates of monthly inflation rates and plot them on the same graph to compare. (There should be 3 lines, one for each datasets, plus the prediction, over time from September 2013 onward.)

```
plt.plot(range(len(df_train),len(df_train) + len(df_test)), df_test['IR'].values, label='with CPI')
plt.plot(range(len(df_train),len(df_train) + len(df2_test)), df2_test['T10YIE'].values, label='with BER')
plt.plot(range(len(df_train),len(df_train) + len(preds)), preds, label='prediction (lag=1)')
plt.legend()
plt.xlabel('t (month)', size=15)
plt.ylabel(r'$IR_t$', size=15)
plt.title('Forecasting the inflation rates', size=20)
plt.show()
```



5.External Regressors and Model Improvements

Include your answers to this question in your written report.

External Regressors

Next, we will include monthly BER data as an external regressor to try to improve the predictions of inflation rate. Here we only consider to add one BER term in the AR(p) model of CPI inflation rate. In specific, we model the CPI inflation rate X_t by

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \psi Y_{t-r} + W_t,$$

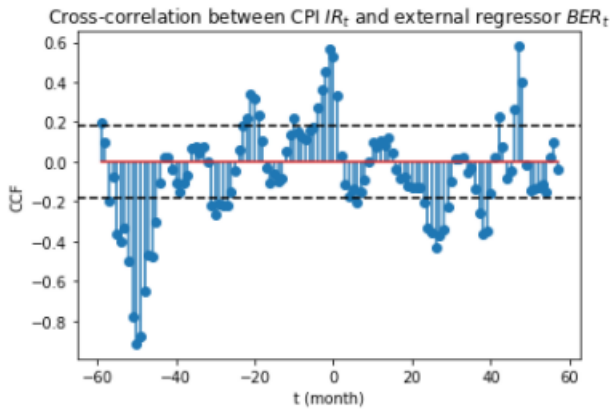
where Y_t is the inflation rate at time t, r >= 0 is the lag of BER rate w.r.t. CPI rate, and W_t is white noise.

1. (4 points) Plot the cross correlation function between the CPI and BER inflation rate, by which find r, i.e., the lag between two inflation rates. (As only one external regressor term is involved in the model, we only consider the peak in the CCF plot.)
   Note: In general, multiple external terms $\sum_{i=1}^{m} \psi_i Y_{t-r_i}$ can be incorporated in the model if there are multiple peaks in CCF plots.

Solution

Let's use the training dataset again to compute the cross-correlations between the two time series datasets, as shown below.

```
import statsmodels.tsa.stattools as smt
forwards = smt.ccf(cpi, ber)
backwards = smt.ccf(ber, cpi)[::-1]
ccf_output = np.r_[backwards[:-1], forwards]
plt.stem(range(-len(ccf_output) // 2, len(ccf_output) // 2), ccf_output)
plt.show()
```



As can be seen, the peak (significant CCF) occurs at lag=48, hence choose r=48.

2. (3 points) Fit a new AR model to the CPI inflation rate with these external regressors and the most appropriate lag. Report the coefficients. *Python Tip:* You

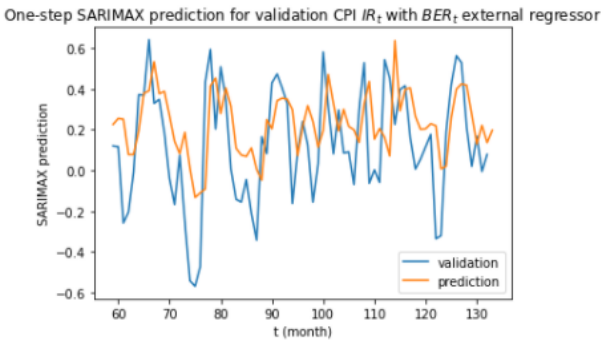
may use `sm.tsa.statespace.SARIMAX`.

Solution

The next code snippet shows how a new AR model can be fitted on the training CPI IR_t data with training BER_t data as external Regressor. Next the fitted SARIMAX model is used for one-step prediction on the validation data. The next plot shows the original test dataset overlayed with the predictions from SARIMAX.

```
# one-step forecasting with SARIMAX
kwargs = {'order': (1, 0, 0), 'trend':'t'}
m = statsmodels.tsa.statespace.sarimax.SARIMAX(cpi_train, ber_train, order=(1,0,0), trend='t').fit()
params = m.params
preds = []
i = 0
while i < len(df_test):
    x_last_obs = m.data.endog[-1]
    x_new_obs= cpi_test['IR'].values[i]
    x = [x_last_obs, x_new_obs]
    y_last_obs = m.data.exog[-1][0]
    y_new_obs= ber_test['T10YIE'].values[i]
    y = [y_last_obs, y_new_obs]
    m = statsmodels.tsa.statespace.sarimax.SARIMAX(x, exog=y, **kwargs)
    res = m.filter(params)
    pred = res.predict()[-1]
    preds.append(pred)
    params = res.params
    i += 1
```

```
plt.plot(range(len(cpi_train), len(cpi_train) + len(preds)), cpi_test['IR'].values)
plt.plot(range(len(cpi_train), len(cpi_train) + len(preds)), preds)
plt.xlabel('t (month)')
plt.ylabel('SARIMAX prediction')
plt.title(r'One-step SARIMAX prediction for validation CPI $IR_t$ with $BER_t$ external regressor')
plt.show()
```



3. (3 points) Report the mean squared prediction error for 1 month ahead forecasts.

Include your answers to the following questions in your written report.

Solution

The RMSE of the SARIMAX model predictions found is 0.26206

```
mse(cpi_test['IR'].values, preds, squared=False)
```

```
0.26206187212253684
```