Lecture 5. ARIMA Modeling and Prediction

Box-Jenkins Methodology

Zhonghai Lu

KTH Royal Institute of Technology

April 11, 2024

Outline

Overview

- Based on the Wold Decomposition theorem, a stationary process can be approximated by an ARMA model. In practice, finding that approximation may not be easy.
- Box and Jenkins popularized an approach that combines the moving average and the autoregressive approaches in the book ”Time Series Analysis: Forecasting and Control” (Box, Jenkins, and Reinsel, 1994).
- Building good ARIMA models generally requires more experience than commonly used statistical methods such as regression.

Stages in Box-Jenkins modeling

There are three primary stages in building a Box-Jenkins time series model and one stage for using it for prediction.

1. Model Identification
2. Parameter Estimation and Model Selection
3. Model Validation
4. Model Prediction

- Typically, effective fitting of Box-Jenkins models requires at least a moderately long series.
- Chatfield (1996) recommends at least 50 observations. Many others would recommend at least 100 observations.

Model identification

- The first step in developing a Box-Jenkins model is to determine if the series is stationary and if there is any significant seasonality that needs to be modeled.
- Stationarity can be assessed from a run sequence or line plot. The run sequence plot should show constant location and scale.
  It can also be detected from an autocorrelation plot. Specifically, non-stationarity is often indicated by an autocorrelation plot with very slow decay.
- Seasonality (or periodicity) can usually be assessed from an autocorrelation plot, a seasonal subseries plot, or a spectral plot.

Differencing operation

- Box and Jenkins recommend the differencing approach to achieve stationarity.
- However, fitting a curve and subtracting the fitted values from the original data can also be used in the context of Box-Jenkins models.

Transformation for stationarity

Three techniques often used to transform a non-stationary time series to stationarity.

- Difference the data. Given a series y_t, we create a new series y'_t: $y'_t = y_t - y_{t-1}$
  The differenced data will contain one less point than the original data. Although you can difference the data more than once, one difference is often sufficient.
- If the data contain a trend, we can fit some type of curve to the data and then model the residuals from that fit.
  Since the purpose of the fit is to simply remove long term trend, a simple fit, such as a straight line, is typically used.
- For non-constant variance, taking the logarithm or square root of the series may stabilize the variance.
  For negative data, you can add a suitable constant to make all the data positive before applying the transformation. This constant can then be subtracted from the model to obtain predicted (i.e., the fitted) values and forecasts for future points.

Variance stabilization

- If the data show different variation at different levels of the series, then a transformation can be useful.
  Assume the original series $\{y_1, y_2, \cdots, y_n\}$, generate a transformed series $\{z_1, z_2, \cdots, z_n\}$ where

$$\text{Square root transform: } z_t = \sqrt{y_t}$$
$$\text{Logarithm transform: } z_t = log(y_t)$$

- In particular, logarithms are useful because they are more interpretable: changes in a log value are relative (percent) changes on the original scale.

Box-Cox transformation

Each of these transformations is close to a member of the family of Box-Cox transformations.

$$z_t = \begin{cases} log(y_t), \lambda = 0 \\ (y_t^\lambda - 1)/\lambda, \lambda \neq 0 \end{cases}$$

- $\lambda = 1$, no substantive transformation.
- $\lambda = 1/2$, square root plus linear transformation.
- $\lambda = 0$, (natural) logarithm transformation.
- $\lambda = -1$, negative reverse plus 1.

- if some $y_t = 0$, then must have $\lambda \neq 0$.
- if some $y_t < 0$, to use logarithm by adding a constant to make all y_t positive.

Back transformation

- We must reverse the transformation (or back-transform) to obtain forecasts on the original scale.
- The reverse Box-Cox transformations are given by

$$y_t = \begin{cases} e^{z_t}, \lambda = 0 \\ (\lambda z_t + 1)^{\frac{1}{\lambda}}, \lambda \neq 0 \end{cases}$$

- For the difference operation ($y'_t = y_t - y_{t-1}$), the back transformation is called an **integration** step done by

$$y_t = y'_t + y_{t-1}$$

Seasonality

- Transformation techniques such as differecning, trend fitting, and Box-Cox transform can be used to achieve Stationarity.
- These transformations are intended to generate series with constant location and scale.
- Although seasonality also violates stationarity, this is usually explicitly incorporated into the time series model.

Seasonality

- Many time series display seasonality, which means periodic fluctuations.
  - For example, time series of retail sales will typically show increasing sales from September through December (peak for the Christmas season) and declining sales in January and February after the holidays.
- If seasonality or periodicity is present, it must be incorporated into the time series model. Two issues:
  - How to detect seasonality?
  - How to model seasonality?

Detect seasonality by visualization

- A line plot will often show seasonality.
- Multiple box plots can be an alternative. It assumes that the seasonal period is known.
- if the period is not known, ACF plot can help identify seasonality.
  - If there is significant seasonality, the autocorrelation plot should show spikes at lags equal to the period.
  - For example, for monthly data, if there is a seasonality effect, we would expect to see significant peaks at lag 12, 24, 36, and so on, although the intensity may decrease.

The CO_2 Example

- The CO_2 data set contains monthly mean CO_2 concentrations at the Mauna Loa Observatory as measured by the continuous infrared analyser of the Geophysical Monitoring for Climatic Change division of NOAA's Air Resources Laboratory.
  In total, it has 161 data.
- Is the series stationary?
- Is the series without trend stationary?

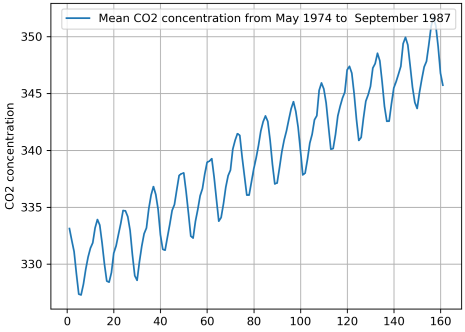


Figure: The CO_2 concentration data set

Example of visualizing seasonality

The CO_2 data set has 161 measurements of monthly mean CO_2 concentrations.
Both plots show evident periodic behavior.

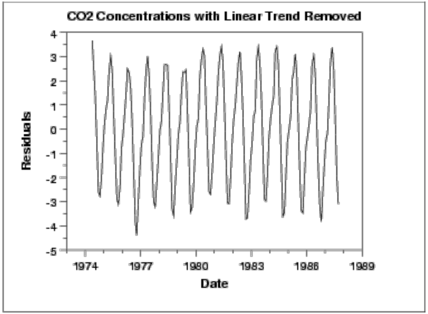


Figure: The CO_2 data without linear trend

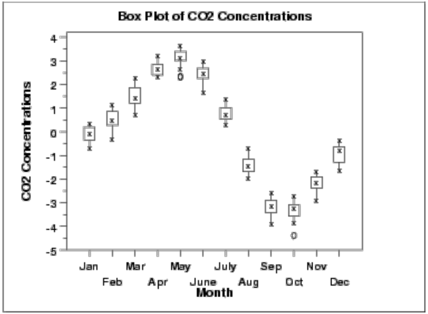


Figure: Box plots for monthly data

ACF of the CO_2 data

For this data which are trended and seasonal, we see a combination of the trend and season effects.

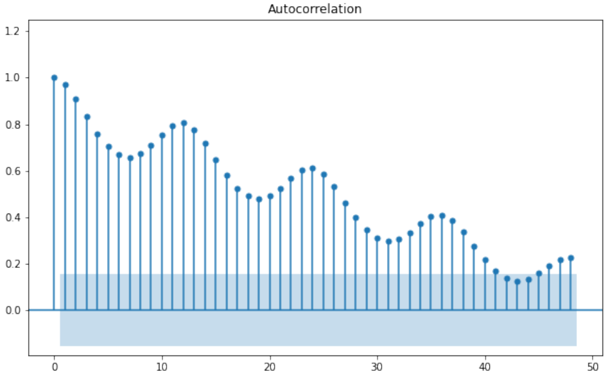
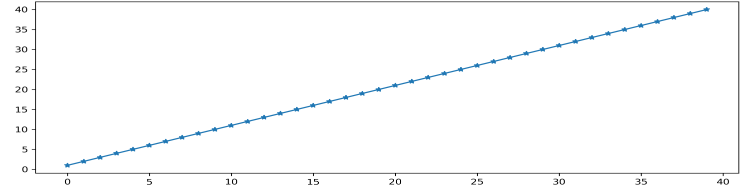


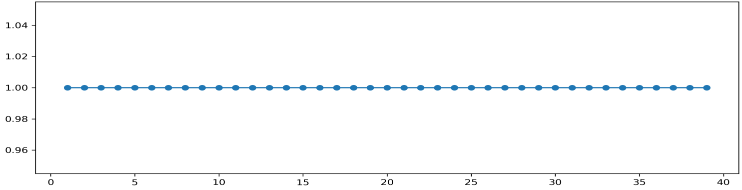
Figure: ACF of the CO_2 data

Differencing example

The differencing can remove trend. Can it remove seasonality?

- Differencing operation can remove trend.
- A simple example

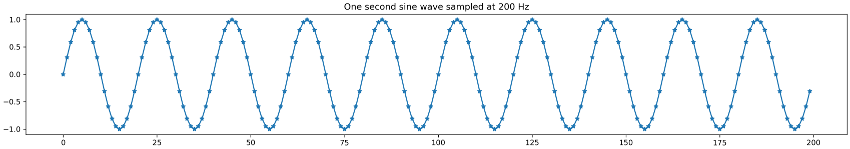




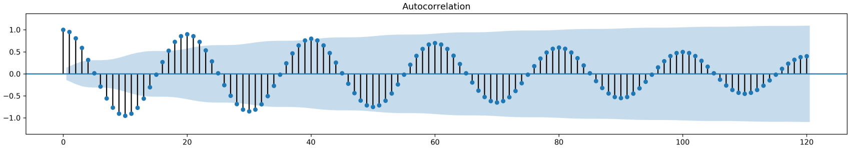
Differencing example

Let's look at a periodic signal.

- Sampling length $T = 1$ s with a sampling rate of 200 Hz, i.e. $N = 200$ data points.
- Signal frequency: $F = 10$ Hz, as 1 second signal shows 10 troughs.
  Period $P = 0.1s$ and one period spans 20 data points.

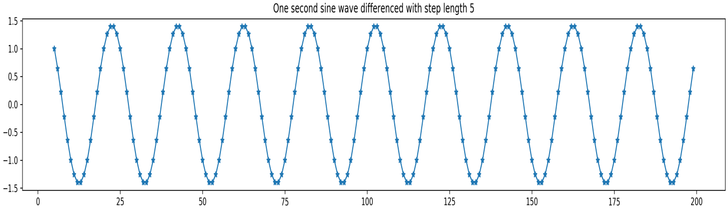


- Auto-correlations are peaked at lags 20, 40, 60, .. i.e., multiples of the period 20.

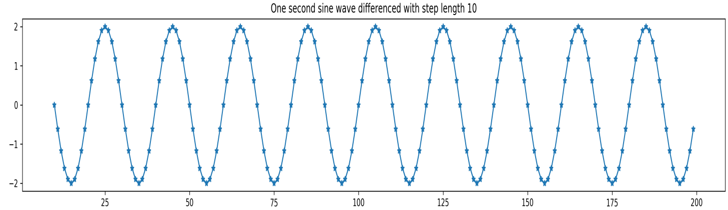


Differencing example

- differencing lag = 5

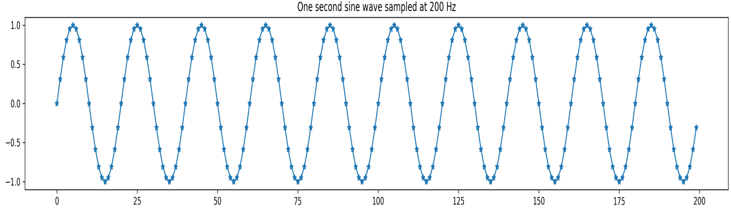


- differencing lag = 10

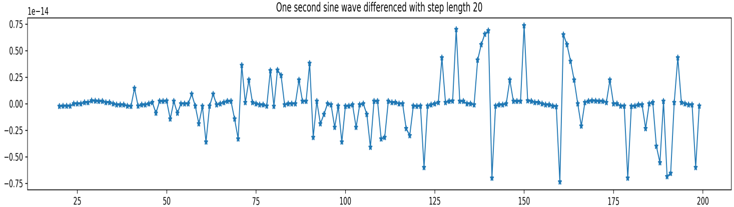


Differencing example

- differencing lag = 0 (original signal)



- differencing lag = 20



If we know the period, then we can use differencing to remove seasonality as well.

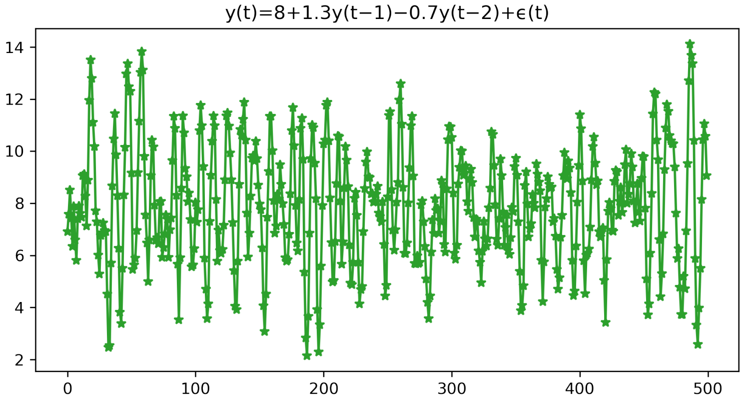
Order identification

- Once stationarity and seasonality are addressed, the next step is to identify the order (i.e., the p and q) of the autoregressive and moving average terms.
- The primary tools are the autocorrelation plot and the partial autocorrelation plot.
- The sample autocorrelation plot and the sample partial autocorrelation plot are compared to the theoretical behavior of these plots when the order is known.

Order p of AR process

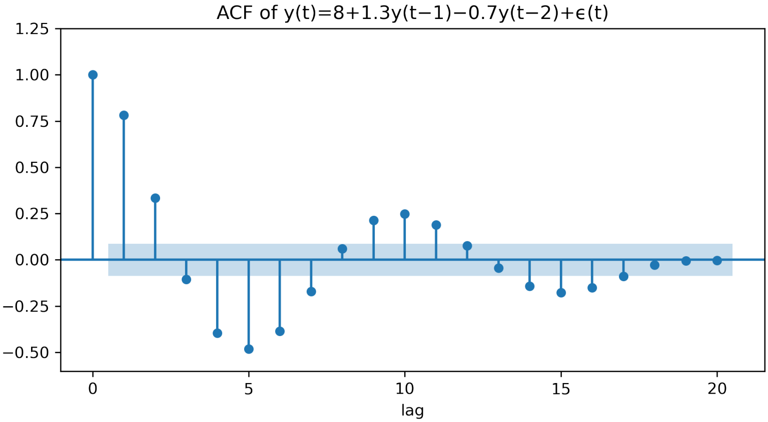
- Since the **partial autocorrelation (PACF)** of an AR(p) process becomes zero at lag p+1 and greater, we examine the sample PACF to see where it becomes zero.
- This can be done by placing a 95% confidence interval on the sample PACF plot.
  - Most software programs that generate sample autocorrelation plots will also plot this confidence interval.
  - If the software program does not generate the confidence band, it is approximately $\pm 2/\sqrt{T}$, with T denoting the sample size.
- For an AR(1) process, the sample autocorrelation function should have an exponentially decreasing appearance. However, higher-order AR processes are often a mixture of exponentially decreasing and damped sinusoidal components. Thus ACF is generally not helpful for identifying the order of the AR process.

Example: AR(2)



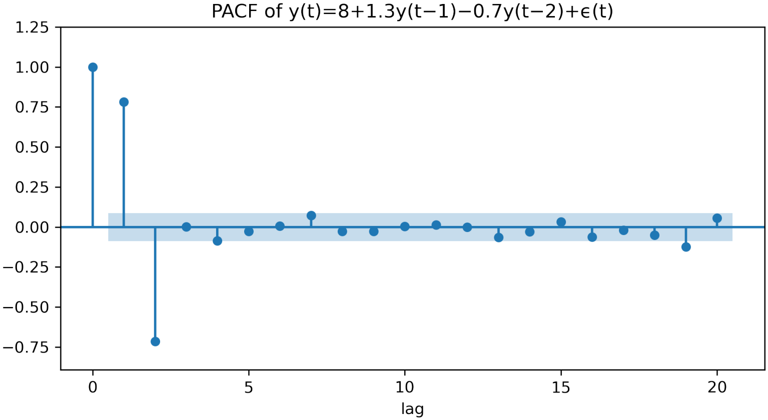
Example: ACF of AR(2)

The ACF of an AR(p) process decreases gradually.



Example: PACF of AR(2)

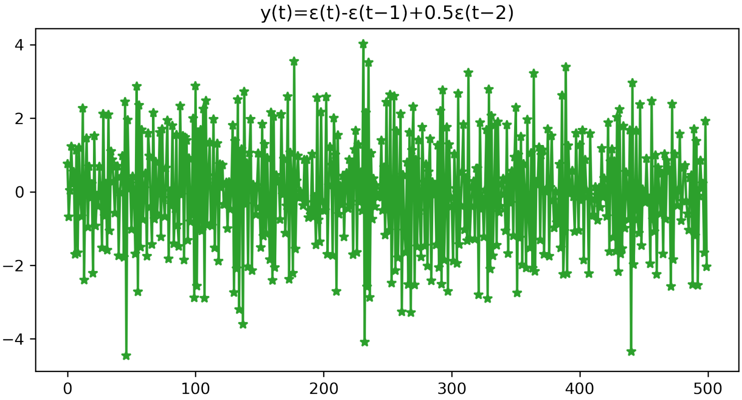
The PACF of an AR(p) process cuts off at lag p.
In the example, its PACF becomes 0 after lag $p = 2$.



Order q of MA process

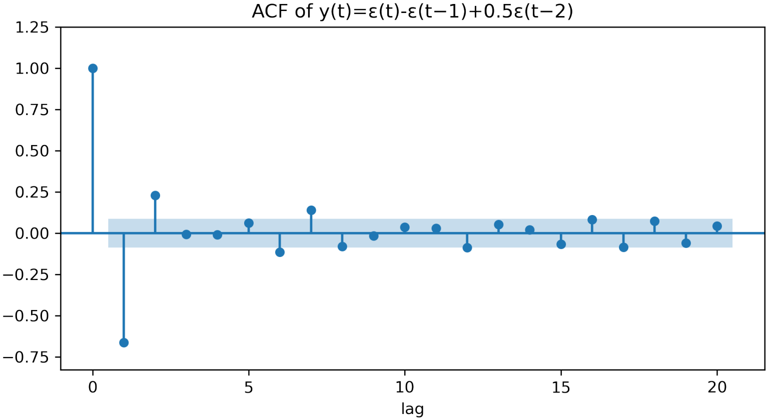
- Since the **autocorrelation function (ACF)** of a MA(q) process becomes zero at lag q+1 and greater, we examine the sample autocorrelation function to see where it essentially becomes zero.
- This can be done by placing the 95% confidence interval for the sample autocorrelation function on the sample autocorrelation plot.
- The sample partial autocorrelation function (PACF) is generally not helpful for identifying the order of the MA process.

Example: MA(2)



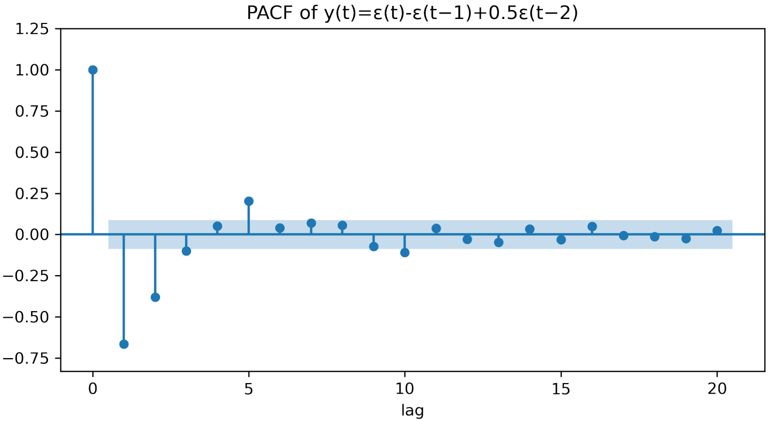
Example: ACF of MA(2)

The ACF of an MA(q) process cuts off sharply.
In the example, its ACF becomes 0 after lag $q = 2$.



Example: PACF of MA(2)

The PACF of an MA(q) process decreases gradually.



Model identification

- The theoretical ACF and PACF for the AR, MA, and ARMA models are known, and are different for each model.
- These differences among models are important to select models.

Model	ACF behavior	PACF behavior
AR(p)	Tails off gradually	Cuts off after p lags
MA(q)	Cuts off after q lags	Tails off gradually
ARMA(p,q)	Tails off gradually	Tails off gradually

Table: ARMA order identification using ACF, PACF

Parameter estimation

Once the model order has been identified (i.e., the values of p, d and q), we need to estimate the parameters $c, \phi_1, \cdots, \phi_p$ and $\theta_1, \theta_2, \cdots, \theta_q$, and σ^2 (variance of residuals).

- The main approaches are non-linear least squares and maximum likelihood estimation (MLE).
- MLE is generally the preferred technique. The likelihood equations for the full Box-Jenkins model are complicated. See (Brockwell and Davis, 1991) for the mathematical details.
- Estimating the parameters is a complicated non-linear estimation problem. For this reason, the parameter estimation should be left to a high quality software program. Different software may give slightly different answers as they use different methods of estimation, and different optimisation algorithms.

Model selection: Information criterion

Often there are many models which can fit the data. Which one to select?

- One common standard is Akaike's Information Criterion (AIC), which weights both model accuracy and model simplicity.

  $$AIC = -2\log(L) + 2(p + q + k + 1),$$

  where L is the likelihood of the data, $k = 1$ if intercept $c \neq 0$ and $k = 0$ if $c = 0$. The second term $(p + d + k + 1)$ is the number of parameters in the model (including σ^2, the variance of the residuals).
- The principle of selection is so called parsimony, which is in favor of a simpler model: A model with a smaller AIC value is better.

Model selection: Information criterion

- Akaike's Information Criterion (AIC):

$$AIC = -2\log(L) + 2(p+q+k+1),$$

where L is the likelihood of the data, $k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$.

- Corrected AIC:

$$AICc = AIC + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2}.$$

- Bayesian Information Criterion (BIC):

$$BIC = AIC + [\log(T) - 2](p+q+k-1).$$

Good models are obtained by minimizing either the AIC, AICc or BIC.

Model diagnostics

- Assumptions for a stable univariate Process: The error term ϵ_t is assumed to follow the assumptions for a stationary univariate process.
  The residuals should be white noise (or independent when their distributions are normal) drawings from a fixed distribution with a constant mean and variance.
- If the Box-Jenkins model is a good model for the data, the residuals should be white noise.
- Otherwise, we go back to the model identification step and try to develop a better model. Hopefully the analysis of the residuals can provide some clues as to a more appropriate model.

Portmanteau test

- In time series analysis, a portmanteau test checks whether any of a group of autocorrelations of the residual time series are different from zero.
- Instead of testing randomness at each distinct lag, it tests the ”overall” randomness based on a number of lags, and is therefore called a portmanteau test.

- Box, G. E. P.; Pierce, D. A. (1970). ”Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models”. Journal of the American Statistical Association. 65 (332): 1509–1526.
- Ljung, G. M.; Box, G. E. P. (1978). ”On a measure of lack of fit in time series models”. Biometrika. 65 (2): 297–303.

Portmanteau test

Consider a whole set of autocorrelation r_k values, check if the whole set is significantly different from a zero set.

- Box-Pierce test

$$Q = T \sum_{k=1}^{h} \gamma_k^2$$

where h is the max lag considered, and T number of observations.
If each γ_k is zero or small, Q is zero or small. If some γ_k values are large, Q is large.

- Ljung-Box test

$$Q = T(T+2) \sum_{k=1}^{h} (T-k)^{-1} \gamma_k^2$$

Preferences: h = 10 for non-seasonal data, h = 2m for seasonal data.
Better performance, especially in small samples.

Multi-step point forecasts

1. Rearrange ARIMA equation so y_t is on left hand side (LHS).
2. Rewrite equation by replacing t by $T + h$.
3. On RHS, replace future observations by their forecasts, future errors by zero, and past errors by corresponding residuals.

Start with $h = 1$. Repeat for $h = 2, 3, \cdots$.

ARIMA(3,1,1) forecasts: Step 1

$$(1 - \phi_1 B - \phi_2 B^2 - \phi_3 B^3)(1 - B) y_t = (1 + \theta_1 B)\epsilon_t$$

$$\begin{aligned}
[1 - (1 + \phi_1)B + (\phi_1 - \phi_2)B^2 + (\phi_2 - \phi_3)B^3 + \phi_3 B^4] y_t &= (1 + \phi_1 B)\epsilon_t, \\
y_t - (1 + \phi_1)y_{t-1} + (\phi_1 - \phi_2)y_{t-2} + (\phi_2 - \phi_3)y_{t-3} + \phi_3 y_{t-4} &= \epsilon_t + \theta_1 \epsilon_{t-1}, \\
y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} &+ \epsilon_t + \theta_1 \epsilon_{t-1}.
\end{aligned}$$

Point forecasts (h=1)

$$y_t = (1 + \phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

ARIMA(3,1,1) forecasts: Step 2

$$y_{T+1} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \epsilon_{T+1} + \theta_1 \epsilon_T.$$

ARIMA(3,1,1) forecasts: Step 3

$$\hat{y}_{T+1|T} = (1 + \phi_1)y_T - (\phi_1 - \phi_2)y_{T-1} - (\phi_2 - \phi_3)y_{T-2} - \phi_3 y_{T-3} + \theta_1 e_T.$$

(Future error = 0, past error = corresponding residual)

$$\epsilon_{T+1} = 0, \ \epsilon_T = e_T$$

Point forecasts (h=2)

$$y_t = (1+\phi_1)y_{t-1} - (\phi_1 - \phi_2)y_{t-2} - (\phi_2 - \phi_3)y_{t-3} - \phi_3 y_{t-4} + \epsilon_t + \theta_1 \epsilon_{t-1}$$

ARIMA(3,1,1) forecasts: Step 2

$$y_{T+2} = (1+\phi_1)y_{T+1} - (\phi_1 - \phi_2)y_T - (\phi_2 - \phi_3)y_{T-1} - \phi_3 y_{T-2} + \epsilon_{T+2} + \theta_1 \epsilon_{T+1}.$$

ARIMA(3,1,1) forecasts: Step 3

$$\hat{y}_{T+2|T} = (1+\phi_1)\hat{y}_{T+1|T} - (\phi_1 - \phi_2)y_T - (\phi_2 - \phi_3)y_{T-1} - \phi_3 y_{T-2}.$$

Prediction intervals

95% prediction interval

$\hat{y}_{T+h|T} \pm 1.96\sqrt{V_{T+h|T}}$
where $V_{T+h|T}$ is estimated forecast variance.

- $V_{T+h|T} = \hat{\sigma}^2$ for all ARIMA models regardless of parameters and orders.
- Multi-step prediction intervals for ARIMA$(0, 0, q)$:

$$y_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}, V_{T+h|T} = \hat{\sigma}^2[1 + \sum_{i=1}^{h-1} \theta_i^2], \text{for} \quad h = 2, 3, \cdots$$

- AR(1): Rewrite as $MA(\inf)$ and use above result.
- Other models beyond scope of this subject.

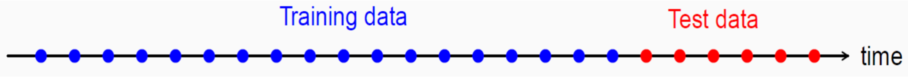
Prediction intervals

- Prediction intervals increase in size with forecast horizon.
- Prediction intervals can be difficult to calculate by hand.
- Calculations assume residuals are uncorrelated and normally distributed.
- Prediction intervals tend to be too narrow.
  - the uncertainty in the parameter estimates has not been accounted for.
  - the ARIMA model assumes historical patterns will not change during the forecast period.
  - the ARIMA model assumes uncorrelated future errors

Training and test sets

- A model which fits the training data well will not necessarily forecast well.
- A perfect training can always be obtained by using a model with enough parameters.
- Overfitting a model to data is just as bad as failing to identify a systematic pattern in the data.
- The test set must not be used for any aspect of model development or calculation of forecasts.
- Forecast accuracy is based only on the test set.

Forecast errors



- Forecast error is the difference between an observed value and its predicted value.

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$$

  where the training data is $\{y_1, y_2, \cdots, y_T\}$
- Unlike residuals, forecast errors on the test set involve multi-step forecasts.
- These are true forecast errors as the test data is not used in computing $\hat{y}_{T+h}|T$

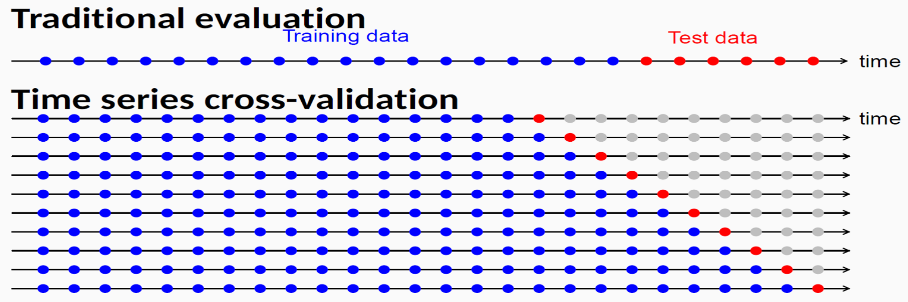
Metrics of forecast accuracy

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T}$$

where y_{T+h} is the $(T+h)th$ observation, $\hat{y}_{T+h|T}$ is the forecast based on data up to time T.

- MAE: Mean Absolute Error.
  MAE = mean($|e_{T+h}|$)
- MSE: Mean Squared Error.
  MSE = mean(e^2_{T+h})
- RMSE: Root Mean Squared Error.
  RMSE = $\sqrt{mean(e^2_{T+h})}$
- MAPE: Mean Absolute Percentage Error.
  MAPE = 100 mean($|e_{T+h}|/|y_{T+h}|$)

Time series cross validation



- Forecast accuracy averaged over test sets. Known as ”evaluation on a rolling forecasting origin”.
- A good way to choose the best forecasting model is to find the model with the smallest RMSE computed using time series cross validation.

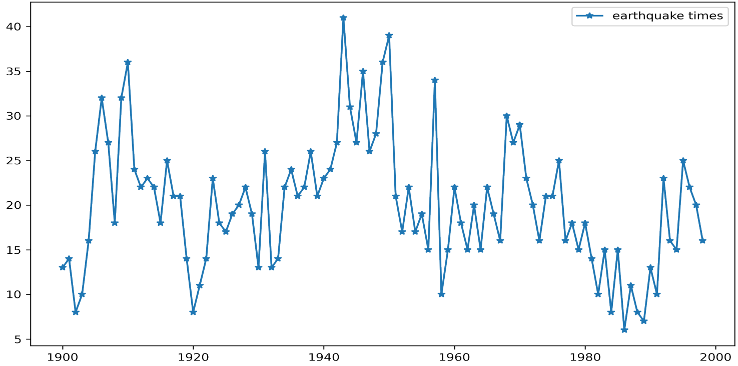
Demo examples[1]

- AR modeling: Model identification `http://localhost:8888/notebooks/IL2233VT22/Lec5_box_jenkins/AR(1)-earthquake.ipynb`
- ARIMA modeling and prediction `http://localhost:8888/notebooks/IL2233VT22/Lec5_box_jenkins/demo_arima_temperature_change_1880-1985.ipynb`

[1] The demo code is only for teacher, not available for students.

Example 1

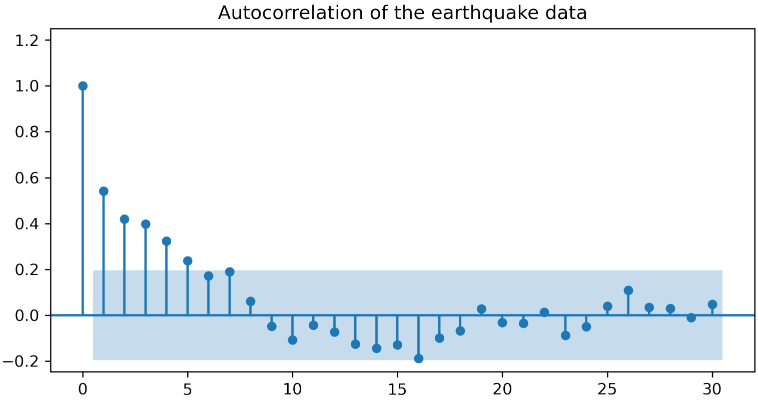
A data set of global earth quake times (7 scale above) from 1900 to 1998.
Source: Applied time-series analysis by Yi and Wang.



We first verify that the series is stationary using the ADF test.

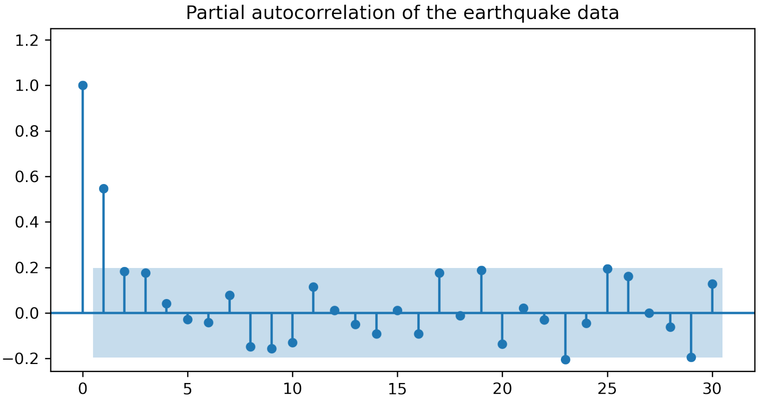
Example 1: Model identification

The ACF of the earth quake times shows a gradual decrease.



Example 1: Model identification

- The ACF of the earth quake times shows a gradual decrease.
- The PACF of the earth quake times shows cut off at lag 1 .
- Thus, we can consider using an AR(1) model to fit the data.



Example 2

Example: Build a model for a temperate change data set (1880 to 1985), 106 data points, 1 per year.

1. Visualization (line plot, acf and pacf graphs) and stationarity test.
   If not stationary, differencing and test stationary again to make sure.
2. Model identification using acf, pacf graphs
3. Model construction and parameter estimation
4. Model diagnostics and validation.
   The model is valid if the residual series is white noise.
5. Model prediction and accuracy measurement

We use *statsmodels* to implement the task.
import statsmodels.api as sm

Data visualization

First draw a line plot to visualize the data.

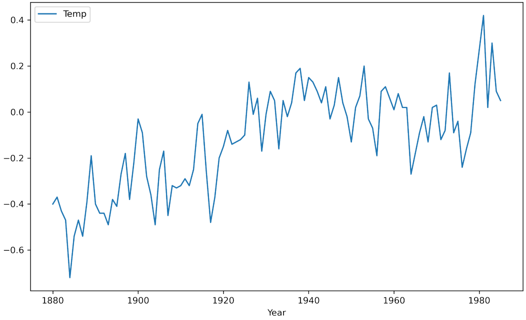


Figure: Temperature change series (1880-1995)

A visual inspection shows that the series is not stationary.
It has an upward trend. Do first-order differencing.

Data differencing

Make a first-order difference of the series.

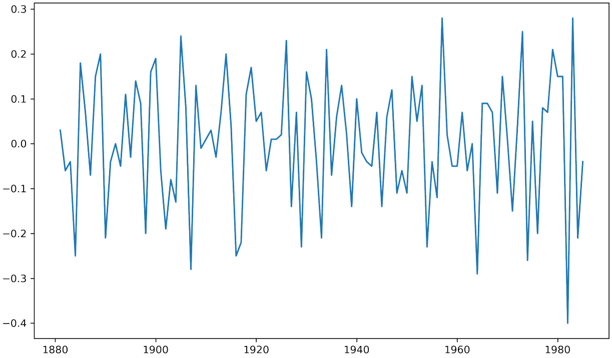


Figure: Differenced temperature change series (1880-1995)

A visual inspection shows that the series seems stationary. Do an ADF test to confirm.

Stationarity test

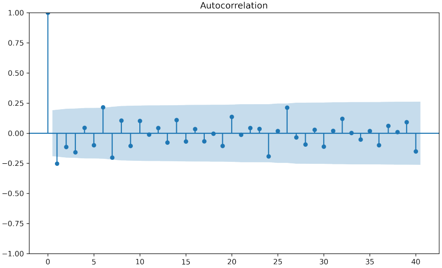
We can do an Augmented Dickey-Fuller (ADF) test to check the stationary of the series.

- If the p-value is less than or equal to 0.05, reject the null hypothesis – the series is stationary.
- If the p-value is larger than 0.05, accept the null hypothesis – the series is not stationary.

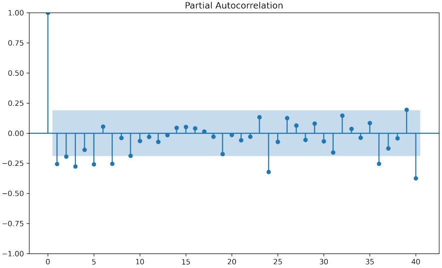
command: adfuller(series)
Result:
Test statistic: -6; p-value: 1.64e-08
Thus we have the confidence that the differenced series is stationary.

Model identification using correlation graphs

Now we examine the ACF/PACF to identify a proper ARMA(p, q) model to fit the data.



(a) ACF



(b) PACF

Figure: ACF and PACF plots

From ACF, we can see MA order q=1 suitable.
From PACF, an AR order p = 7 may be meaningful.

Model construction

```
import statsmodels.api as sm
model = sm.tsa.statespace.SARIMAX(series, order=(7,1,1),trend=[1,1,0,0])  #('ct')
model = model.fit(disp=-1)
print(model.summary())
```

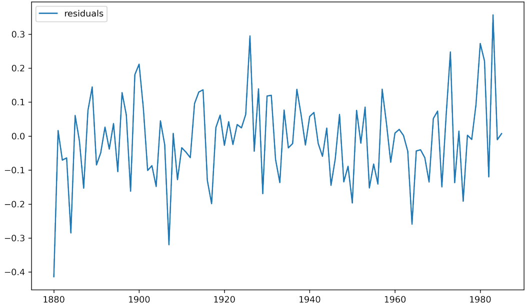


Figure: The residual series

Model diagnostics

Check the residual series: (1) plot the series; (2) plot its acf and pacf; (3) Ljung-Box test for white noise.

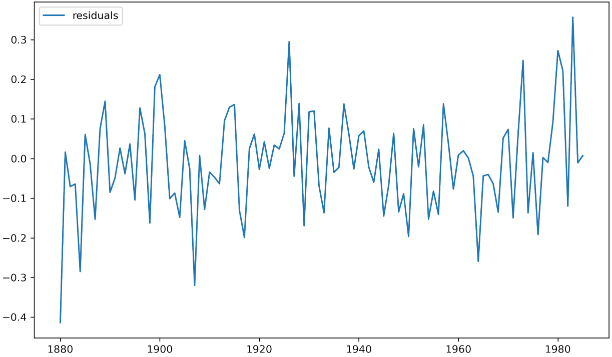
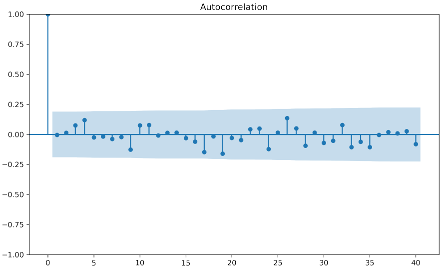


Figure: The residual series

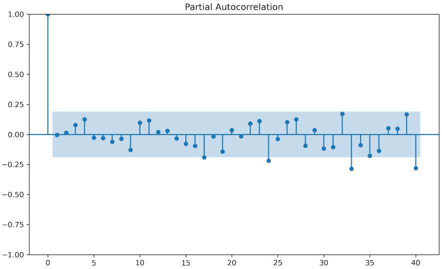
Model diagnostics: visual inspection

Check the residual series: (1) plot the series; (2) plot its acf and pacf; (3) Ljung-Box test for white noise.

plot(model.resid) plot_acf(model.resid, lags=40) plot_pacf(model.resid, lags=40) show()



(a) ACF



(b) PACF

Figure: ACF and PACF plots of the residual series

Model diagnostics: Ljung-Box test

```
sm.stats.acorr_ljungbox(model.resid, lags=[40], return_df=True)
```

	lb_statistic	lb_pvalue
40	27.612341	0.931141

Table: Ljung-Box test for white noise

Since the pvalue $0.931141 > 0.05$, we Accept the Null hypothesis. Thus the residual series is confirmed to be a white noise series. Thus the model is validated.

In-sample and out-of-sample prediction

- In-sample prediction

```
predictions = model.get_prediction()
```

- Out-of-sample prediction: H-step forecasts

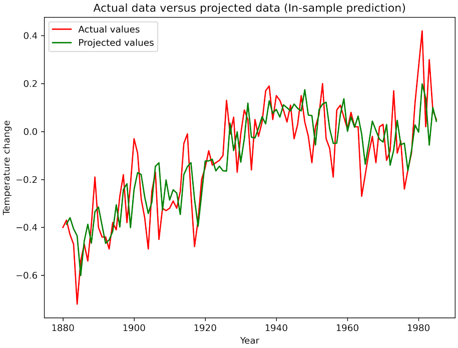
```
forecasts = model.get_forecast(steps=5)
```

- Load specific evaluation tools

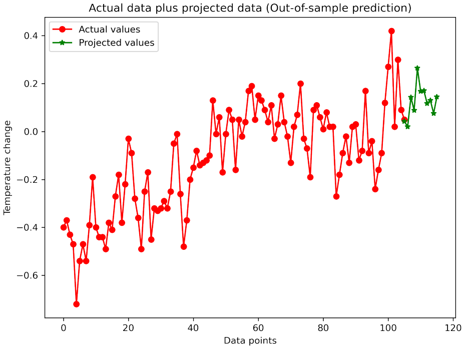
```
from sklearn.metrics import mean_squared_error
from statsmodels.tools.eval_measures import rmse
print('mse=', mean_squared_error(test, predictions))
print('rmse=', rmse(test, predictions))
```

Model forecasting

Now we show how the in-sample and out-of-sample predictions look like.



(a) In-sample prediction



(b) Out-of-sample prediction

Figure: Model prediction results

Summary

- Box-Jenkins methodology has well-defined steps to fit an ARMA model to time-series data.
- Model identification utilizes the properties of the ACF and PACF graphs of AR, MA, ARMA processes.
- Parameter estimation can be complicated and is handled by estimation software.
- There are often many possible models to fit the data. Model selection criterion can use AIC, AICc, BIC based on the principle of parsimony.
- Model validation to check the residuals before applying it for prediction.

References

- NIST/SEMATECH e-Handbook of Statistical Methods, `http://www.itl.nist.gov/div898/handbook/`
- George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung (June 2015). Time Series Analysis: Forecasting and Control (5th Edition). NJ: Wiley. ISBN: 978-1-118-67502-1.
- Brockwell, Peter; Davis, Richard (2009). Time Series: Theory and Methods (2nd ed.). New York: Springer. ISBN 9781441903198.
- Hyndman, R. J., Athanasopoulos, G. (2018). "Forecasting: principles and practice", 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2. `https://otexts.com/fpp3/`