MAT315 Time Series

Project 2: Box-Jenkins Method

Chin Zi Yi
Ton Sin Pei
Yang LiuQing
Wong Yong Zhi
Li Kai

July 2024

Introduction

- Box-Jenkins method is a mathematical method that is frequently used to forecast data.
- It applies to autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models.
- The different stages of Box-Jenkins method are:
  1. Model Identification
  2. Model Estimation
  3. Model Validation
  4. Forecasting
- Objective: Find the best fitted ARMA or ARIMA model.
- The software **R** will be used.

Box-Jenkins Method

1. Model Identification

1. Check Stationarity:
   - Augmented Dickey-Fuller (ADF) test is used.
   - Null Hypothesis: The series is non-stationary.
   - If a series is non-stationary, differencing is needed.
2. Determine Differencing Order (d):
   - Transforms a non-stationary series into a stationary one.

Box-Jenkins Method

1. Model Identification

3. Plot ACF and PACF:
   - ACF (Autocorrelation Function):
     - Helps to determine the order of the Moving Average (MA) part.
   - PACF (Partial Autocorrelation Function):
     - Helps to determine the order of the Autoregressive (AR) part.
4. Model Selection and Comparison:
   - Fit ARIMA models with different combinations of p, d, q.
   - Use the criteria AIC (Akaike Information Criterion) to select the best model.

Box-Jenkins Method

2. Model Estimation

- After identifying the best model, fit the coefficients to estimate the parameters.
- Fitted model is acquired in this step.

Box-Jenkins Method

3. Model Validation

1. Residual ACF:
   - ACF (Autocorrelation Function):
     - Checks the autocorrelation of residuals at different lags.
     - The ACF plot of residuals should show that most autocorrelations are within the 95% confidence interval.
2. Ljung-Box Test:
   - Autocorrelation of Residuals:
     - Null Hypothesis: The residuals are independently distributed.
     - Examines whether the residuals are independently distributed.
     - A high p-value (typically > 0.05) indicates that the residuals are not significantly autocorrelated, suggesting a good fit.
3. Normal Q-Q Plot:
   - Normal Distribution of Residuals:
     - Checks if the residuals are normally distributed.
     - The points in the Normal Q-Q plot should approximate a straight line if the residuals are normally distributed.

Box-Jenkins Method

4. Forecasting

- Forecast the next 100 observations using the best model.
- Include the 95% confidence interval of the predicted values.

Data Source

- The given time series data can be imported into **R** by:

```
> data <- read.table("Group2.txt", header = TRUE)
> ts_data <- ts(data)
> length(ts_data)
[1] 501
```

- The dataset consists of 501 observations.
- The necessary packages are:

```
> library(forecast)
> library(tseries)
> library(ggplot2)
```

Data Source

- The **R** code below is used to produce a time plot of the data.

```
> plot(ts_data)
```

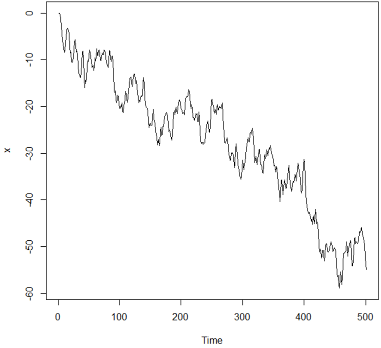

Figure: Time plot of the time series data.

Data Analysis

1. Model Identification

- The **R** code below is declaring two frequently used functions:
  - Function for checking stationarity:

```
> check_stationarity <- function(ts) {
+   adf_test <- adf.test(ts)
+   print(paste("ADF Statistic: ", adf_test$statistic))
+   print(paste("p-value: ", adf_test$p.value))
+   print(adf_test)
}
```

  - Function for plotting ACF and PACF:

```
> plot_acf_pacf <- function(ts) {
+  par(mfrow = c(1,2))
+  acf(ts, main = 'ACF')
+  pacf(ts, main = 'PACF')
+  par(mfrow = c(1,1))
}
```

Data Analysis

1. Model Identification

- Check stationarity of original data.

```
> check_stationarity(ts_data)
[1] "ADF Statistic:  -3.26815113718006"
[1] "p-value:  0.0761808384172313"

        Augmented Dickey-Fuller Test

data:  ts
Dickey-Fuller = -3.2682, Lag order = 7, p-value = 0.07618
alternative hypothesis: stationary
```

Data Analysis

1. Model Identification

- Performs the first order differencing on the original series, since it is not stationary as shown in the ADF Test.

```
> diff_data <- diff(ts_data)
```

- Time plot after performing first order differencing.

```
> plot(diff_data)
```

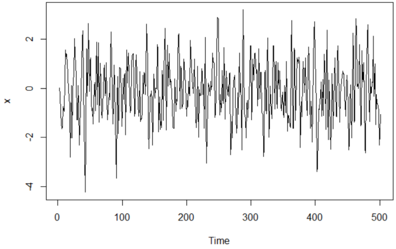

Figure: Time plot after performing first order differencing.

Data Analysis

1. Model Identification

- Check stationarity after differencing.

```
> check_stationarity(diff_data)
[1] "ADF Statistic:  -9.19205010729188"
[1] "p-value:  0.01"

        Augmented Dickey-Fuller Test

data:  ts
Dickey-Fuller = -9.1921, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(ts) : p-value smaller than printed p-value
```

Data Analysis

1. Model Identification

- Check the ACF and PACF of the time series data after differencing:

```
> plot_acf_pacf(diff_data)
```

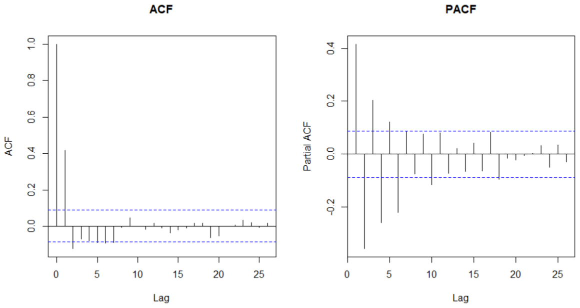

Figure: ACF and PACF of time series data after differencing.

Data Analysis

1. Model Identification

- The R code below is used to find optimal values for p,d,q of ARIMA based on AIC.

```
> best_aic <- Inf
> best_order <- c(0, 0, 0)
> top_results <- data.frame(AIC = numeric(), Order = character(), stringsAsFactors = FALSE)

> for (p in 0:3) {
+  for (d in 0:1) {
+   for (q in 0:3) {
+    arima_model <- tryCatch(Arima(ts_data, order = c(p, d, q)), error = function(e) NULL)
+     if (!is.null(arima_model)) {
+      current_aic <- AIC(arima_model)
+
+      top_results <- rbind(top_results,
+                           data.frame(AIC = current_aic, Order = paste(c(p, d, q), collapse = ",")))
+      top_results <- top_results[order(top_results$AIC), ]
+      if (nrow(top_results) > 10) {
+       top_results <- top_results[1:10, ]
+      }
+
+      if (current_aic < best_aic) {
+       best_aic <- current_aic
+       best_order <- c(p, d, q)
+ } } } } }
```

- Top 10 results with lowest AIC:

```
> print(top_results, row.names = FALSE)
     AIC Order
1354.822 1,1,2
1355.170 3,1,3
1356.707 1,1,3
1356.710 2,1,2
1358.678 3,1,2
1358.699 0,1,2
1358.707 2,1,3
1358.764 0,1,3
1359.262 1,1,1
1359.280 2,1,1

> print(paste("Best AIC: ", best_aic))
[1] "Best AIC:  1354.82201425864"
> print(paste("Best Order: ", paste(best_order, collapse = ",")))
[1] "Best Order:  1,1,2"
```

- Get the coefficient of the parameters of the best model ARIMA(1,1,2).

```
> best_model <- arima(ts_data, order = best_order)
> best_model

Call:
arima(x = ts_data, order = best_order)

Coefficients:
         ar1     ma1      ma2
      0.7598  0.0809  -0.7999
s.e.  0.1236  0.1035   0.0926

sigma^2 estimated as 0.8621:  log likelihood = -673.41,  aic = 1354.82
```

- We can look at the coefficients to have a higher accuracy fitted values for calculating the forecast values by hand.

```
> coef(best_model)
       ar1         ma1         ma2 
0.75975580  0.08086348 -0.79991888 
```

- By computing the fitted model of ARIMA(1,1,2):
  - $(1 - 0.75975580\mathrm{B})(1 - \mathrm{B})x_t = (1 + 0.08086348\mathrm{B} - 0.79991888\mathrm{B}^2)w_t$

$$\begin{aligned} \implies x_t = &1.75975580x_{t-1} - 0.75975580x_{t-2} + w_t \\ &+ 0.08086348w_{t-1} - 0.79991888w_{t-2}. \end{aligned}$$

Data Analysis

3. Model Validation

- Check the ACF of the residuals of our model ARIMA(1,1,2).

```
> acf(resid(best_model))
```

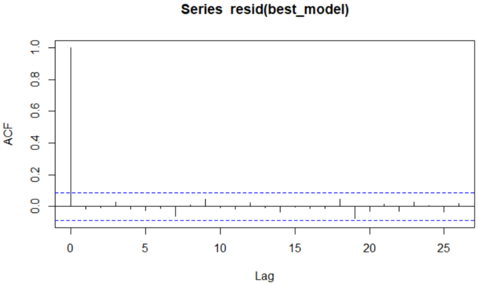

Figure: ACF of residuals.

- Check the residuals by using Ljung-Box Test.

```
> Box.test(resid(best_model), lag = 20, type = "Ljung-Box")

        Box-Ljung test

data:  resid(best_model)
X-squared = 9.9986, df = 20, p-value = 0.9682
```

Data Analysis

3. Model Validation

- Check the residuals using Normal Q-Q Plot.

```
> qqnorm(resid(best_model))
> qqline(resid(best_model),col = "red")
```

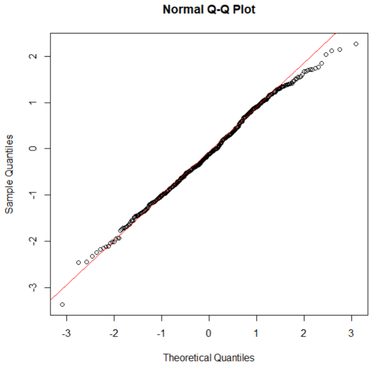

Figure: Normal Q-Q Plot.

- Forecasting the next 100 observations using ARIMA(1,1,2) model.

```
> data_for <- predict(best_model, n.ahead = 100)
> ts.plot(ts_data, data_for$pred,
+            ylab = "data", col = "blue", ylim = c(-76, 0.5))
> grid ()
```

- Additional plotting:

```
> U = data_for$pred + 2 * data_for$se
> L = data_for$pred - 2 * data_for$se
> xx = c(time (U), rev (time (U)))
> yy = c(L, rev(U))
> polygon(xx, yy, border = 8, col = gray (0.6, alpha = 0.2))
> lines(data_for$pred, type = "p", col = "red")
```

Data Analysis

4. Forecasting

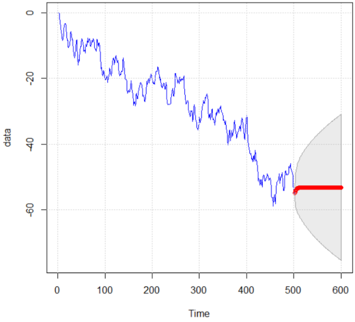

Figure: Forecasting values and its 95% confidence interval.

- **Example**: Forecasting $\hat{x}_{502}$ by hand.

```
> tail(ts_data)
Time Series:
Start = 496
End = 501
Frequency = 1
               x
[1,] -48.20913
[2,] -48.81779
[3,] -49.71260
[4,] -52.02617
[5,] -53.86378
[6,] -54.91672
```

- **Example**: Forecasting $\hat{x}_{502}$ by hand.

```
> tail(resid(best_model), n = 3)
Time Series:
Start = 499
End = 501
Frequency = 1
[1] -1.1521242 -1.1033624 -0.4891882
```

- Fitted model:

$$\begin{aligned} x_t &= 1.75975580 x_{t-1} - 0.75975580 x_{t-2} + w_t \\ &\quad + 0.08086348 w_{t-1} - 0.79991888 w_{t-2} \end{aligned}$$

- **Example**: Forecasting $\hat{x}_{502}$ by hand.

$$\begin{aligned}
\hat{x}_{502} &= 1.75975580 x_{501} - 0.75975580 x_{500} + \hat{w}_{502} \\
&\quad + 0.08086348 w_{501} - 0.79991888 w_{500} \\
&= 1.75975580(-54.91672) - 0.75975580(-53.86378) + 0 \\
&\quad + 0.08086348(-0.4891882) - 0.79991888(-1.1033624) \\
&= -54.87365432
\end{aligned}$$

- Verify in **R**:

```
> head(data_for$pred, n = 1)
Time Series:
Start = 502
End = 502
Frequency = 1
[1] -54.87366
```

Conclusion

- The best fitted model for the data is ARIMA(1,1,2).
- Fitted Model:

$$\begin{aligned} x_t = &\ 1.75975580 x_{t-1} - 0.75975580 x_{t-2} + w_t \\ &+ 0.08086348 w_{t-1} - 0.79991888 w_{t-2}, \end{aligned}$$

  with w_t being a Gaussian White Noise.
- The fitted model is validated by checking the correlograms.
- 100 observations is then forecasted with the first five being:

```
> data_for$pred[1:5]
[1] -54.87366 -54.44963 -54.12747 -53.88270 -53.69674
```

Thank You

References

- Box Jenkins Method.
- Time Series Analysis: ARIMA Modelling using R software.