02417: Time Series Analysis

Week 8 – the MARIMA estimation method and Multivariate time series

Peder Bacher
DTU Compute

Based on material previous material from the course

March 27, 2025

Week 8: Outline of the lecture

- Introduction to MARIMA
- How the MARIMA method works (how it includes an MA part)
- Chapter 9 – Multivariate time series

The MARIMA package and the Spliid method

Run examples to find out how the marima and the Spliid method works!

- `marima` is an R package implementing the Spliid method, see: Article
- Let's run some examples together, download "`Week8_example_marima.zip`" and unzip

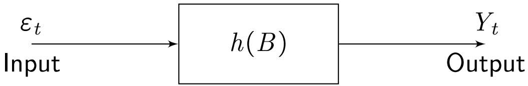
The MARIMA package and the Spliid method

Run examples to find out how the marima and the Spliid method works!

- `marima` is an R package implementing the Spliid method, see: Article
- Let's run some examples together, download "`Week8_example_marima.zip`" and unzip

The MARIMA package and the Spliid method

- Remember ARMA is noise through a transfer function:

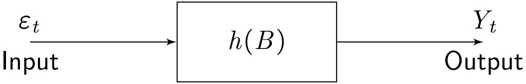


- e.g. ARMA(1,1)

$$Y_t = -\phi_1 Y_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$
$$Y_t = \frac{1 + \theta_1 B}{1 + \phi_1 B} \varepsilon_t$$

The MARIMA package and the Spliid method

- Remember ARMA is noise through a transfer function:



- e.g. ARMA(1,1)

$$Y_t = -\phi_1 Y_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

$$Y_t = \frac{1 + \theta_1 B}{1 + \phi_1 B} \varepsilon_t$$

- How the Spliid method includes the MA part:

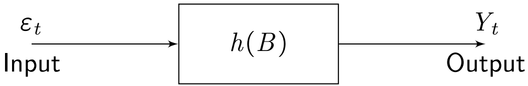
  Step 1: Make AR and estimate with LS

  Step 2: Take the residuals and lag as MA part;
  Put in model with the AR part, and estimate again with LS

  Step 3: Iterate until the residuals don't change

The MARIMA package and the Spliid method

- Remember ARMA is noise through a transfer function:



- e.g. ARMA(1,1)

$$Y_t = -\phi_1 Y_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$
$$Y_t = \frac{1+\theta_1 B}{1+\phi_1 B}\varepsilon_t$$

- How the Spliid method includes the MA part:

  Step 1: Make AR and estimate with LS

  Step 2: Take the residuals and lag as MA part;
  Put in model with the AR part, and estimate again with LS

  Step 3: Iterate until the residuals don't change

- Start with "`marima_from_scratch_arma.R`": Simulate ARMA and estimate

The MARIMA package and the Spliid method

Run examples to find out how the marima and the Spliid method works!

- Can we use the Spliid method with an external regressor!?

The MARIMA package and the Spliid method

Run examples to find out how the marima and the Spliid method works!

- Can we use the Spliid method with an external regressor!?
- Yes, just include it in the LS model (with lags)
- We can fit an ARMAX model with a transfer function!
- E.g. ARMAX(1,1):

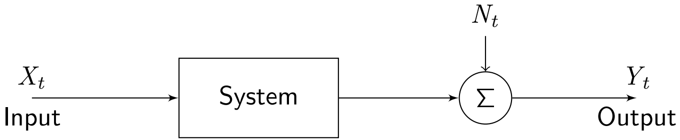
$$Y_t = -\phi_1 Y_{t-1} + \omega_1 x_{t-1} + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

$$Y_t = \frac{\omega_1 B}{1 + \phi_1 B} x_t + \frac{1 + \theta_1 B}{1 + \phi_1 B} \varepsilon_t$$

- Look into "`marima_from_scratch_armax.R`"

Multivariate models

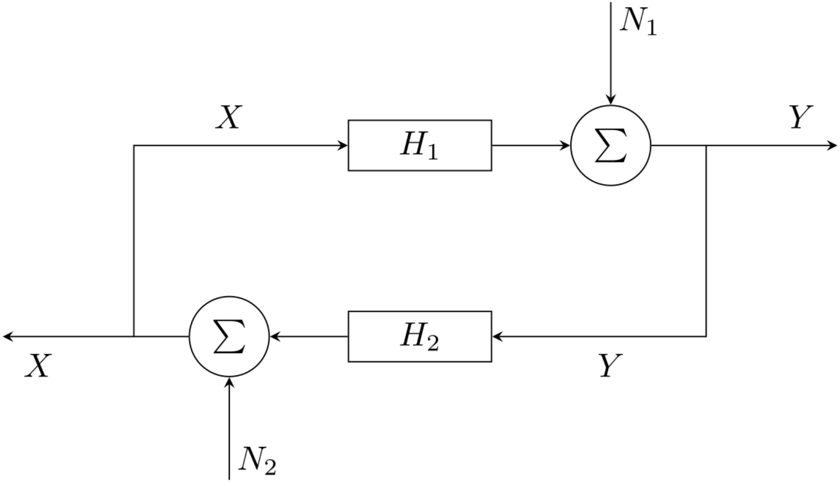
Re-consider the univariate transfer function model:



$$Y_t = h(B)X_t + N_t$$

- What if there is a feedback from Y to X?

Closed Loop Models



$$Y_t = h_1(B)X_t + N_{1,t}$$
$$X_t = h_2(B)\,Y_t + N_{2,t}$$

Closed Loop Models

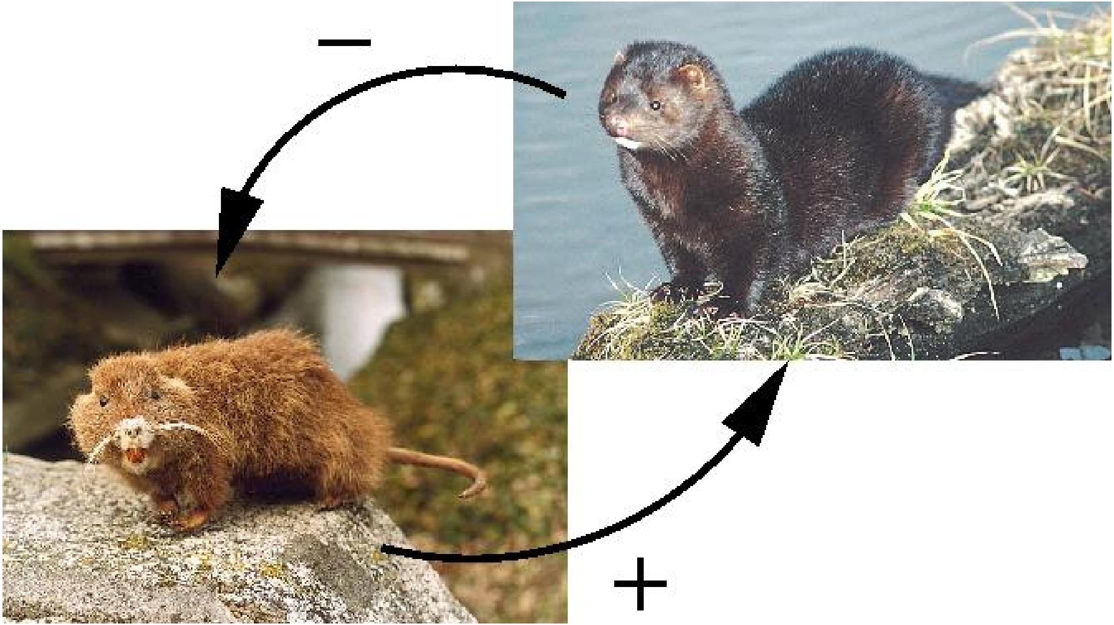
$$Y_t = h_1(B) X_t + N_{1,t}$$
$$X_t = h_2(B) Y_t + N_{2,t}$$

Or:

$$\begin{pmatrix} 1 & -h_1(B) \\ -h_2(B) & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} N_{1,t} \\ N_{2,t} \end{pmatrix}$$

- Two inputs (N_1, N_2);
- Two outputs (Y, X);
- Four transfer functions from input to output.

Predetor-pray: Mink-Muskrat example



Transfer from N_1, N_2 to Y:

$$\begin{aligned} Y_t &= h_1(B)X_t + N_{1,t} \\ X_t &= h_2(B)Y_t + N_{2,t} \end{aligned}$$

Transfer from N_1, N_2 to Y:

$$
\begin{aligned}
Y_t &= h_1(B)X_t + N_{1,t} \\
X_t &= h_2(B)\,Y_t + N_{2,t} \\
Y_t &= h_1(B)(h_2(B)\,Y_t + N_{2.t}) + N_{1,t}
\end{aligned}
$$

Transfer from N_1, N_2 to Y:

$$
\begin{aligned}
Y_t &= h_1(B)X_t + N_{1,t} \\
X_t &= h_2(B)Y_t + N_{2,t} \\
Y_t &= h_1(B)(h_2(B)Y_t + N_{2.t}) + N_{1,t}
\end{aligned}
$$

Z-domain:

$$
Y(z) = H_1(z)(N_2(z) + H_2(z)Y(z)) + N_1(z)
$$

Transfer from N_1, N_2 to Y:

$$
\begin{aligned}
Y_t &= h_1(B)X_t + N_{1,t} \\
X_t &= h_2(B)Y_t + N_{2,t} \\
Y_t &= h_1(B)(h_2(B)Y_t + N_{2.t}) + N_{1,t}
\end{aligned}
$$

Z-domain:

$$
\begin{aligned}
Y(z) &= H_1(z)(N_2(z) + H_2(z)Y(z)) + N_1(z) \\
Y(z) - H_1(z)H_2(z)Y(z) &= H_1(z)N_2(z) + N_1(z)
\end{aligned}
$$

Transfer from N_1, N_2 to Y:

$$
\begin{aligned}
Y_t &= h_1(B)X_t + N_{1,t} \\
X_t &= h_2(B)Y_t + N_{2,t} \\
Y_t &= h_1(B)(h_2(B)Y_t + N_{2.t}) + N_{1,t}
\end{aligned}
$$

Z-domain:

$$
\begin{aligned}
Y(z) &= H_1(z)(N_2(z) + H_2(z)Y(z)) + N_1(z) \\
Y(z) - H_1(z)H_2(z)Y(z) &= H_1(z)N_2(z) + N_1(z) \\
Y(z) &= \frac{1}{1 - H_1(z)H_2(z)}N_1(z) + \frac{H_1(z)}{1 - H_1(z)H_2(z)}N_2(z)
\end{aligned}
$$

Transfer from N_1, N_2 to Y:

$$\begin{aligned} Y_t &= h_1(B)X_t + N_{1,t} \\ X_t &= h_2(B)Y_t + N_{2,t} \\ Y_t &= h_1(B)(h_2(B)Y_t + N_{2.t}) + N_{1,t} \end{aligned}$$

Z-domain:

$$\begin{aligned} Y(z) &= H_1(z)(N_2(z) + H_2(z)Y(z)) + N_1(z) \\ Y(z) - H_1(z)H_2(z)Y(z) &= H_1(z)N_2(z) + N_1(z) \\ Y(z) &= \frac{1}{1 - H_1(z)H_2(z)} N_1(z) + \frac{H_1(z)}{1 - H_1(z)H_2(z)} N_2(z) \end{aligned}$$

Transfer functions from N_1, N_2 to Y:

$$N_1 : \frac{1}{1 - H_1(z)H_2(z)} \text{ and } N_2 : \frac{H_1(z)}{1 - H_1(z)H_2(z)}$$

Multivariate transfer function

Model equation:

$$\begin{pmatrix} 1 & -h_1(B) \\ -h_2(B) & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} N_{1,t} \\ N_{2,t} \end{pmatrix}$$

Model equation in Z-domain:

$$\begin{pmatrix} 1 & -H_1(z) \\ -H_2(z) & 1 \end{pmatrix} \begin{pmatrix} Y(z) \\ X(z) \end{pmatrix} = \begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix}$$

Multivariate transfer function

Model equation:

$$\begin{pmatrix} 1 & -h_1(B) \\ -h_2(B) & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} N_{1,t} \\ N_{2,t} \end{pmatrix}$$

Model equation in Z-domain:

$$\begin{pmatrix} 1 & -H_1(z) \\ -H_2(z) & 1 \end{pmatrix} \begin{pmatrix} Y(z) \\ X(z) \end{pmatrix} = \begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix}$$

Thus, from the result on the previous slide:

$$\begin{pmatrix} Y(z) \\ X(z) \end{pmatrix} = \frac{1}{1 - H_1(z)H_2(z)} \begin{pmatrix} 1 & H_1(z) \\ H_2(z) & 1 \end{pmatrix} \begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix}$$

Multivariate transfer function

Model equation:

$$\begin{pmatrix} 1 & -h_1(B) \\ -h_2(B) & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} N_{1,t} \\ N_{2,t} \end{pmatrix}$$

Model equation in Z-domain:

$$\begin{pmatrix} 1 & -H_1(z) \\ -H_2(z) & 1 \end{pmatrix} \begin{pmatrix} Y(z) \\ X(z) \end{pmatrix} = \begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix}$$

Thus, from the result on the previous slide:

$$\begin{pmatrix} Y(z) \\ X(z) \end{pmatrix} = \frac{1}{1 - H_1(z)H_2(z)} \begin{pmatrix} 1 & H_1(z) \\ H_2(z) & 1 \end{pmatrix} \begin{pmatrix} N_1(z) \\ N_2(z) \end{pmatrix}$$

Multivariate transfer function:

$$\begin{pmatrix} \frac{1}{1-H_1(z)H_2(z)} & \frac{H_1(z)}{1-H_1(z)H_2(z)} \\ \frac{H_2(z)}{1-H_1(z)H_2(z)} & \frac{1}{1-H_1(z)H_2(z)} \end{pmatrix}$$

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

- The individual time series may have been transformed and differenced

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

- The individual time series may have been transformed and differenced

- The variance-covariance matrix of the multivariate white noise process $\{\boldsymbol{\epsilon}_t\}$ is denoted $\boldsymbol{\Sigma}$.

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

- The individual time series may have been transformed and differenced
- The variance-covariance matrix of the multivariate white noise process $\{\boldsymbol{\epsilon}_t\}$ is denoted $\boldsymbol{\Sigma}$.
- The matrices $\boldsymbol{\phi}(B)$ and $\boldsymbol{\theta}(B)$ have elements which are polynomials in the backshift operator

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

- The individual time series may have been transformed and differenced
- The variance-covariance matrix of the multivariate white noise process $\{\boldsymbol{\epsilon}_t\}$ is denoted $\boldsymbol{\Sigma}$.
- The matrices $\boldsymbol{\phi}(B)$ and $\boldsymbol{\theta}(B)$ have elements which are polynomials in the backshift operator
- The diagonal elements have leading terms of unity

Multivariate ARMA models

- The multivariate ARMA process

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1 \boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p \boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1 \boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q \boldsymbol{\epsilon}_{t-q}$$

  where $\{\boldsymbol{\epsilon}_t\}$ is white noise, is called a Vector ARMA (VARMA) process.

- The model can be written

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

- The individual time series may have been transformed and differenced
- The variance-covariance matrix of the multivariate white noise process $\{\boldsymbol{\epsilon}_t\}$ is denoted $\boldsymbol{\Sigma}$.
- The matrices $\boldsymbol{\phi}(B)$ and $\boldsymbol{\theta}(B)$ have elements which are polynomials in the backshift operator
- The diagonal elements have leading terms of unity
- The off-diagonal elements have leading terms of zero (i.e. they normally start in B)

Air pollution in cities NO and NO_2

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.9 & -0.1 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 30 & 21 \\ 21 & 23 \end{bmatrix}$$

Air pollution in cities NO and NO_2

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.9 & -0.1 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} 30 & 21 \\ 21 & 23 \end{bmatrix}$$

Matrix formulation using the backshift operator:

Air pollution in cities NO and NO_2

$$\begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 0.9 & -0.1 \\ 0.4 & 0.8 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} \xi_{1,t} \\ \xi_{2,t} \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 30 & 21 \\ 21 & 23 \end{bmatrix}$$

Matrix formulation using the backshift operator:

$$\begin{bmatrix} 1 - 0.9B & 0.1B \\ -0.4B & 1 - 0.8B \end{bmatrix} \boldsymbol{X}_t = \boldsymbol{\xi}_t \quad \text{or} \quad \boldsymbol{\phi}(B)\boldsymbol{X}_t = \boldsymbol{\xi}_t$$

Stationarity and Invertability

The multivariate ARMA process

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

is stationary if

$$\det(\boldsymbol{\phi}(z^{-1})) = 0 \quad \Rightarrow \quad |z| < 1$$

Stationarity and Invertability

The multivariate ARMA process

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

is stationary if

$$\det(\boldsymbol{\phi}(z^{-1})) = 0 \quad \Rightarrow \quad |z| < 1$$

is invertible if

$$\det(\boldsymbol{\theta}(z^{-1})) = 0 \quad \Rightarrow \quad |z| < 1$$

Auto Covariance Matrix Functions

$$\boldsymbol{\Gamma}_k = E[(\boldsymbol{Y}_{t-k} - \boldsymbol{\mu}_Y)(\boldsymbol{Y}_t - \boldsymbol{\mu}_Y)^T] = \boldsymbol{\Gamma}_{-k}^T$$

Auto Covariance Matrix Functions

$$\boldsymbol{\Gamma}_k = E[(\boldsymbol{Y}_{t-k} - \boldsymbol{\mu}_Y)(\boldsymbol{Y}_t - \boldsymbol{\mu}_Y)^T] = \boldsymbol{\Gamma}_{-k}^T$$

Example for bivariate case $\boldsymbol{Y}_t = (Y_{1,t} \;\; Y_{2,t})^T$:

$$\boldsymbol{\Gamma}_k = \begin{bmatrix} \gamma_{11}(k) & \gamma_{12}(k) \\ \gamma_{21}(k) & \gamma_{22}(k) \end{bmatrix} = \begin{bmatrix} \gamma_{11}(k) & \gamma_{12}(k) \\ \gamma_{12}(-k) & \gamma_{22}(k) \end{bmatrix}$$

Auto Covariance Matrix Functions

$$\boldsymbol{\Gamma}_k = E[(\boldsymbol{Y}_{t-k} - \boldsymbol{\mu}_Y)(\boldsymbol{Y}_t - \boldsymbol{\mu}_Y)^T] = \boldsymbol{\Gamma}_{-k}^T$$

Example for bivariate case $\boldsymbol{Y}_t = (Y_{1,t} \;\; Y_{2,t})^T$:

$$\boldsymbol{\Gamma}_k = \begin{bmatrix} \gamma_{11}(k) & \gamma_{12}(k) \\ \gamma_{21}(k) & \gamma_{22}(k) \end{bmatrix} = \begin{bmatrix} \gamma_{11}(k) & \gamma_{12}(k) \\ \gamma_{12}(-k) & \gamma_{22}(k) \end{bmatrix}$$

We can describe these by plotting

- each autocovariance or autocorrelation function for $k = 0, 1, 2, \ldots$ and
- each cross-covariance or cross-correlation function for $k = 0, \pm 1, \pm 2, \ldots$

Identification using Autocovariance Matrix Functions

Sample Correlation Matrix Function; $\boldsymbol{R}_k$ near zero for pure moving average processes of order q when $k > q$

Sample Partial Correlation Matrix Function; $\boldsymbol{S}_k$ near zero for pure autoregressive processes of order p when $k > p$

Identification using (multivariate) prewhitening

- Fit univariate models to each individual series
- Investigate the residuals as a multivariate time series
- Model selection procedure!
- The cross correlations can then be compared with $\pm 2/\sqrt{N}$

This is **not** the same form of prewhitening as in Chapter 8

Identification using (multivariate) prewhitening

- Fit univariate models to each individual series
- Investigate the residuals as a multivariate time series
- Model selection procedure!
- The cross correlations can then be compared with $\pm 2/\sqrt{N}$

This is **not** the same form of prewhitening as in Chapter 8

Remember the result in two dimensions:

$$
\begin{aligned}
Y_t &= h_1(B)X_t + N_{1,t} && (1)\\
X_t &= h_2(B)Y_t + N_{2,t} && (2)\\
Y_t &= h_1(B)(h_2(B)Y_t + N_{2.t}) + N_{1,t} && (3)
\end{aligned}
$$

Identification using (multivariate) prewhitening

- Fit univariate models to each individual series
- Investigate the residuals as a multivariate time series
- Model selection procedure!
- The cross correlations can then be compared with $\pm 2/\sqrt{N}$

This is **not** the same form of prewhitening as in Chapter 8

Remember the result in two dimensions:

$$
\begin{aligned}
Y_t &= h_1(B) X_t + N_{1,t} && (1) \\
X_t &= h_2(B) Y_t + N_{2,t} && (2) \\
Y_t &= h_1(B)(h_2(B) Y_t + N_{2.t}) + N_{1,t} && (3)
\end{aligned}
$$

in general the multivariate model $\boldsymbol{\phi}(B) Y_t = \boldsymbol{\theta}(B) \epsilon_t$ is equivalent to

$$\textbf{diag}(\det(\boldsymbol{\phi}(B))) Y_t = \textbf{adj}(\boldsymbol{\phi}(B)) \boldsymbol{\theta}(B) \epsilon_t$$

Therefore the corresponding univariate models will have much higher order, so although this is often done in the literature: Don't take this approach!

Multivariate ARMA(p,q) processes (centered data)

- Matrices with polynomials in B as elements:

$$\boldsymbol{\phi}(B)\boldsymbol{Y}_t = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

  So the coefficients are now matrices:

$$\boldsymbol{Y}_t + \boldsymbol{\phi}_1\boldsymbol{Y}_{t-1} + \ldots + \boldsymbol{\phi}_p\boldsymbol{Y}_{t-p} = \boldsymbol{\epsilon}_t + \boldsymbol{\theta}_1\boldsymbol{\epsilon}_{t-1} + \ldots + \boldsymbol{\theta}_q\boldsymbol{\epsilon}_{t-q}$$

- In general, no analytic solution exits.
- Therefore, estimation algorithms (or numerical optimization) is necessary.

Estimation procedures

For multivariate ARX(p):

- Least squares estimation is possible

For multivariate ARMAX(p,q):

- The Spliid method (Henrik Spliid, 1983)
- Maximum likelihood

Go and have a look into `marima_from_scratch_armax_bivariate.R`, it's a very short example on how to simulate and fit a bivariate ARMAX(1,1) with MARIMA.

Highlights

- Closed loop model as multivariate transfer function

$$\begin{pmatrix} 1 & -h_1(B) \\ -h_2(B) & 1 \end{pmatrix} \begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} N_{1,t} \\ N_{2,t} \end{pmatrix}$$

- Multivariate ARMA models

$$\boldsymbol{\phi}(B)(\boldsymbol{Y}_t - \boldsymbol{c}) = \boldsymbol{\theta}(B)\boldsymbol{\epsilon}_t$$

  is stationary if

$$\det(\boldsymbol{\phi}(z^{-1})) = 0 \quad \Rightarrow \quad |z| < 1$$

  is invertible if

$$\det(\boldsymbol{\theta}(z^{-1})) = 0 \quad \Rightarrow \quad |z| < 1$$

- Auto covariance matrix functions

$$\boldsymbol{\Gamma}_k = E[(\boldsymbol{Y}_{t-k} - \boldsymbol{\mu}_Y)(\boldsymbol{Y}_t - \boldsymbol{\mu}_Y)^T] = \boldsymbol{\Gamma}_{-k}^T$$

- All VARMA models can be written as VAR(1)

Exercises and Assignment 3

- A new exercise was uploaded, gives you "hands-on" of fitting ARX and ARMAX to data from an experimental set up
- Upload Assignment 3 in the afternoon, take a look at it