MIDA1: Recap

Model Identification and Data Analysis I

Professor: Sergio Bittanti

Authors: Simone Staffa

Based on the notes provided by Giulio Alizoni

June 16, 2020

Contents

1 Random Variables Refresh

A random variable v can be described by three main properties:

- **Mean**: $E[v]$. Key property of expected value $E[\alpha_1 v_1 + \alpha_2 v_2] = \alpha_1 E[v_1] + \alpha_2 E[v_2]$
- **Variance**: $E[(v - E[v])^2] \geq 0$
- **Standard Deviation**: $\sqrt{Var[v]}$

Random Vector: a vector composed of random variables.

The **Cross Variance** between two elements of it can be defined as:

$$\lambda_{12} = E[(v_1 - E[v_1])(v_2 - E[v_2])] \text{ with } V = |v_1 v_2|^T$$

We can define the **Variance Matrix** as $Var[V] = \begin{bmatrix} \lambda_{1,1} & \lambda_{1,2} \\ \lambda_{2,1} & \lambda_{2,2} \end{bmatrix}$. This matrix has some properties:

- Symmetric
- Positive Semi-Definite: $det(Var[v]) \geq 0$

Defining the **covariance coefficient** $\rho = \frac{\lambda_{12}}{\sqrt{\lambda_{11} * \sqrt{\lambda_{22}}}}$, we obtain that $|\rho| \leq 1$

2 Random Process Introduction

A random process (also called **stochastic process**) is a sequence of random variables. We will focus on **Stationary Stochastic Process (SSP)** which have the following characteristics:

- **Mean** is constant (m)
- **Variance** is constant (λ^2)
- $\gamma(t_1, t_2) = E[(v(t_1) - m)(v(t_2) - m)]$ the **covariance function**, it depends only on $\tau = t_2 - t_1$ and can therefore be indicated with $\gamma(\tau)$. The variance can be also defined as $\gamma(0) = \lambda^2$

A **White Noise (WN)** is a SSP with $\gamma(0) = 0 \ \forall \tau \neq 0$. A WN is an unpredictable signal meaning that there is NO CORRELATION between $\eta(t_1)$ and $\eta(t_2)$. Usually a WN is defined as $\eta(t) \sim WN(0, \lambda^2)$. Having them with zero-mean is not mandatory.

3 Familes of SSP

3.1 Moving Average Process (MA)

$$MA(n) : v(t) = c_0 \eta(t) + c_1 \eta(t-1) + ... + c_n \eta(t-n)$$

Where n is the order of the process, $\eta(t) \sim WN(0, \lambda^2)$.

- By definition, **MA processes are stationary** because they are the result of the sum of stationary processes (white-noise)
- **Mean** is $E[v(t)] = (c_0 + c_1 + ... + c_n) E[\eta(t)] = 0$
- **Covariance function**:

$$\gamma(\tau) = \begin{cases} \lambda^2 * \sum_{i=0}^{n-\tau} c_i c_{i-\tau} & |\tau| \leq n \\ 0 & otherwise \end{cases} \tag{1}$$

- **Transfer function**: $W(z) = c_0 + c_1 z^{-1} + ... + c_n z^{-n} = \frac{c_0 z^n + c_1 z^{n-1} + ... + c_n}{z^n}$.
  All poles are then in the origin of the complex plane, while the zeroes depends on the values of the coefficient.

It exists also the $MA(\infty)$ representation:

$$v(t) = c_0\eta(t) + c_1\eta(t-1) + ... + c_i\eta(t-i) + ... + infinitesum.$$

In this way the variance in particular becomes an infinite sum as well:

$$\gamma(0) = (c_0^2 + c_1^2 + ... + c_i^2 + ...)\lambda^2.$$

We have then to impose the basic condition: $\sum_{i=0}^{\infty} c_i^2$ needs to be FINITE, because we need to have $|\gamma(\tau)| \leq \gamma(0)$. Under this condition, $MA(\infty)$ is well defined and a stationary process.

3.2 Auto Regressive Process (AR)

$$AR(n) : v(t) = a_1 v(t-1) + a_2 v(t-2) + ... + a_n v(t-n) + \eta(t)$$

Where n is the order of the process, $\eta(t) \sim WN(0, \lambda^2)$.

- **Mean** is computed applying $E[v(t)]$.
- **Covariance function** with $n = 1$ (case of $AR(1)$):

$$\gamma(\tau) = \begin{cases} a\gamma(\tau-1) & |\tau| \geq 0 \\ \frac{1}{1-a^2}\lambda^2 & \tau = 0 \end{cases} \tag{2}$$

  well defined if $|a| < 1$ (Yule Walker Equations)

- **Transfer function**: $W(z) = \frac{z^n}{z^n - a_1 z^{n-1} - ... - a_n}$.

  Note that poles' position depends on the value $a_1, ..., a_n$. If all poles are inside the unit circle, then the system is **stable**.

3.3 ARMA Process

$$ARMA(n_a, n_c) : v(t) = a_1 v(t-1) + ... + a_{n_a} v(t-n_a) + c_0\eta(t) + ... + c_1\eta(t-1) + ... + c_{n_c}\eta t - n_c$$

It is composed by an AR and an MA part. The process is stationary if STABLE, meaning that all the poles needs to be inside the unit circle.
An ARMA process can be expressed as $v(t) = \frac{C(z)}{A(z)}\eta(t)$ having:

$$C(z) = c_0 + c_1 z^{-1} + ... + c_{n_c} z^{-n_c}$$
$$A(z) = 1 - a_1 z^{-1} - ... - a_{n_a} z^{-n_a}$$
$$\rightarrow W(z) = \frac{C(z)}{A(z)}$$

Using the Long Division Algorithm one can obtain the **impulse response** representation of it, which is a sort of $MA(\infty)$ expression of the process, where $W(z) = w_0 + w_1 z^{-1} + w_2 z^{-2} + ...$ as the result of the long division between $C(z)$ and $A(z)$.

4 Spectral Representation

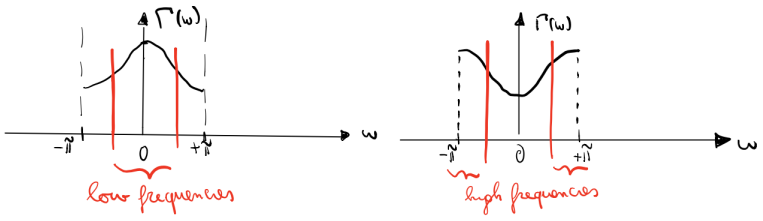
By the original definition, the spectrum of a stationary process is the *fourier transform* of its covariance function:

$$\Gamma(\omega) = \mathcal{F}[\gamma(\tau)] = \sum_{\tau=-\infty}^{\infty} \gamma(\tau) e^{-j\omega\tau}$$

When computing the spectrum, remember the Eulero formula: $\frac{e^{j\omega\tau}+e^{-j\omega\tau}}{2} = cos(\omega\tau)$
Spectrum properties:

- $\Gamma(\omega)$ is a **real function** of the real variable ω
- $\Gamma(\omega)$ is **periodic** with $T = 2\pi$
- $\Gamma(\omega) = \Gamma(-\omega)$ (**even function**) since both $\gamma(\bullet)$ and $cos(\bullet)$ are even functions
- $\Gamma(\omega) \geq 0 \ \forall\omega$

Remember that if the spectrum is higher with low frequencies, means that the signal moves very slowly. Vice versa, if the spectrum is higher with high frequencies the signal moves very quickly. There exist also the complex spectrum:



$$\Phi(z) = \sum_{\tau=-\infty}^{\infty} \gamma(\tau) z^{-\tau} \rightarrow \Gamma(\omega) = \Phi(z = e^{j\omega})$$

One can also anti-transform the spectrum to obtain the covariance function:

$$\gamma(\tau) = \int_{-\pi}^{+\pi} \Gamma(\omega) e^{j\omega\tau} d\omega$$

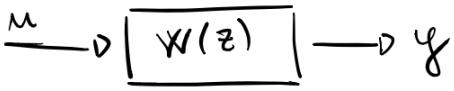
The spectrum of a white noise is constant and equal to its variance. Given $\eta(t) \sim WN(0, \lambda^2)$

$$\Gamma_\eta(\omega) = \gamma(0) = Var[\eta(t)] = \lambda^2$$

4.1 Fundamental Theorem of Spectral Analysis

The spectrum of a process y that takes in input a process u is:

$$\Gamma_y(\omega) = |W(e^{j\omega})|^2 * \Gamma_u(\omega)$$



5 Canonical Representation of a Stationary Process

To solve the multiplicity of ARMA models for a stationary process, in which there are many different ARMA representations for the same process, one can use the canonical representation. From a signal, one can build the spectrum (or equivalently the covariance function). Once we have the spectrum, we can derive the canonical spectral factor to solve the prediction problem. Indeed, given a rational process, there is one and only one ARMA representation which is canonical.

Properties of Canonical Representation (referring to the transfer function $W(z)$):

- Numerator and denominator have the same degree
- Numerator and denominator are monic (the term with the highest power has coefficient equal to 1)
- Numerator and denominator are coprime (no common factors to that can be simplified)
- Numerator and denominator are stable polynomials: all poles and zeroes of $W(z)$ are inside the unit disk

6 Prediction Problem

6.1 Fake Problem

In the fake problem we want to find the predictor from the noise.
We want to compute $\hat{v}(t+r)$ given the past of η. We can write:

$$\hat{v(t+r)} = \hat{W(z)}\eta(t+r) =$$
$$\hat{w}_0\eta(t+r) + \hat{w}_1\eta(t+r-1) + ... + \hat{w}_{r-1}\eta(t+1) + \hat{w}_r\eta(t) + \hat{w}_{r+1}\eta(t-1) + ... = \alpha(t) + \beta(t).$$

Since η is a WN and we don't know anything of its future, $\alpha(t)$ is FULLY UNPREDICTABLE.
While $\beta(t)$ is predictable, cause it depends on the past of η. Thus we can write:

$$\hat{v}(t+r|t) = \hat{v}(t|t-r) = \beta(t)$$

To evaluate the performance of our prediction we can use the **prediction error** and its variance:

$$\epsilon(t) = v(t) - \hat{v}(t+r|t)$$
$$Var[\epsilon(t)] = \lambda^2(\hat{w}_0^2 + \hat{w}_1^2 + ... + \hat{w}_{r-1}^2)$$

Note that the variance of the error increases with r, meaning that more distant prediction result to be less precise.
Practically $\hat{v}(t+r|t) = \hat{W}_r(z)\eta(t)$, where $\hat{W}_r(z)$ is the result of the r-step LONG DIVISION of the numerator and denominator of $W(z)$ in canonical form.

6.2 True Problem

In the true problem we want to find the predictor from data, meaning the past values of $v(\bullet)$.
The solution to this is simply:

$$W_r(z) = \hat{W}(z)^-1\hat{W}_r(z)$$

with $\hat{W}_r(z)$ being the transfer function of the predictor, $W(z)$ being the transfer function of the original system in canonical form and $\hat{W}_r(z)$ being the solution to the fake problem. Here "r" are the number of steps of the predictor.
There is a shortcut that can be used with ARMA process, in general:

$$\hat{v}(t|t-1) = \frac{C(z)-A(z)}{C(z)}v(t), \text{ with } \hat{W}(z) = \frac{C(z)}{A(z)}$$

6.3 Prediction with eXogenous Variables

An exogenous variable is another input variable $u(t)$ for the system which differently from the WN is a deterministic variable.

6.3.1 ARX Model

$$ARX(n_{a,b}) : v(t) = a_1v(t-1) + ... + a_{n_a}v(t-n_a) + b_1u(t-1) + ... + b_{n_b}u(t-n_b) + \eta(t)$$
or in operator form
$$A(z)v(t) = B(z)u(t-1) + \eta(t)$$

meaning that the transfer function from u to v is $\frac{B(z)}{A(z)}$ and the one from η to v is $\frac{1}{A(z)}$.

6.3.2 ARMAX Model

Similarly to ARX models, it is defined as $A(z)v(t) = C(z)\eta(t) + B(z)u(t-1)$..
It can be represented also with the Box & Jenkins model, in which the WN is considered as a disturb (or noise) and $G(z)$ is the effect of the exogenous variable:

$$y(t) = G(z)u(t) + W(z)\eta(t)$$

The new predictor formula becomes:

$$\hat{y}(t|t-1) = \frac{C(z)-A(z)}{C(z)}y(t) + \frac{B(z)}{C(z)}u(t-1)$$

7 Prediction Error minimization methods

Identification consists in estimating a model from data. We can define the prediction error as before: $\epsilon(t) = v(t) - \hat{v}(t+r|t)$. We want to minimize this prediction error, and representing it as a WN (fully unpredictable).
Steps:

1. **Data collection**: $u(1), ..., u(n)$ and $y(1), ..., y(n)$
2. **Choice of the model family**: $M(\theta)$ and corresponding $\epsilon_\theta(t)$, where θ is a vector of parameters
3. **Choice of optimization criterion**: for example we can have

$$J(\theta) = \frac{1}{N}\sum_{t=1}^{N}\epsilon_\theta(t)^2 \text{ mean squared error}$$
$$J(\theta) = \frac{1}{N}\sum_{t=1}^{N}|\epsilon_\theta(t)| \text{ mean absolute error}$$

Optimization: model parameters are obtained with $\theta = minJ(\theta) \rightarrow \frac{dJ(\theta)}{d\theta} = 0$

Validation: final analysis of the results to evaluate if they satisfy our requirements

7.1 Least Square method - LS

Consider the ARX model:

$$M(\theta) : y(t) = a_1(t-1) + ... + a_{n_a}y(t-n_a) + b_1u(t-1) + ... + u_{n_b}(t-n_b) + \xi(t) = \theta^T\phi(t) + \eta(t)$$

being $\theta = |a_1, ...a_{n_a}, b_1, ..., b_{n_b}|$ and $\phi(t) = |y(t-1), ...y(t-n_a), u(t-1), ..., u(t-n_b)|$.
Considering as optimization criterion the Mean Squared Error, we impose its derivative equal to zero and we obtain the **normal equations** and the parameters estimate ($\hat{\theta}$).

$$\sum_{t=1}^{N}\phi(t)\phi^T(t)\theta = \sum_{t=1}^{N}y(t)\phi^T(t) \rightarrow \hat{\theta} = [\sum_{t=1}^{N}\phi(t)\phi^T(t)]^{-1} * \sum_{t=1}^{N}y(t)\phi^T(t)$$

Depending on the value of the second derivative, the matrix $\frac{d^2J(\theta)}{d\theta^2}$, we can have two situations:

1. **Positive Definite Matrix**: one point of minimum $\hat{\theta}$ and $J(\theta)$ is a bowl (a paraboloid with vertex in $\hat{\theta}$
2. **Positive Semidefinite Matrix**: infinite many solutions ($J(\theta)$ is similar to a section of a pipe)

Let's now consider the $R(N)$ matrix, defined as $R(N) = \frac{1}{N}\sum_{t=1}^{N}\theta(t)\theta^T(t)$.
In an ARX(1,1) we have the following:

$$R(N) = \begin{bmatrix} \frac{1}{N}y(t-1)^2 & \frac{1}{N}y(t-1)u(t-1) \\ \frac{1}{N}u(t-1)y(t-1) & \frac{1}{N}u(t-1)^2 \end{bmatrix}$$

Note that the two elements in the main diagonal are respectively the sample variance of y and the sample variance of u.

Bringing $N \to \infty$: $\bar{R} = \begin{bmatrix} \bar{R}_{yy} & \bar{R}_{yu} \\ \bar{R}_{uy} & \bar{R}_{uu} \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(0) & \gamma_{yu}(0) \\ \gamma_{uy}(0) & \gamma_{uu}(0) \end{bmatrix}$

In a general ARX(n_a,n_b) model, we obtain that:

$$\bar{R}_u u = \begin{bmatrix} \gamma_{uu}(0) & \gamma_{uu}(1) & \gamma_{uu}(2) & ... \\ \gamma_{uu}(1) & \gamma_{uu}(0) & ... & \\ \gamma_{uu}(2) & .. & & \end{bmatrix}$$

This matrix is called **Toepliz matrix**: inside each diagonal there's the same element; the one inside the main diagonal is the variance. *(Note that for $\bar{R}_{yy}$ is the same but with $y(\bullet)$).*

7.2 Maximum Likelihood method - ML

This method is based on ARMAX models instead of ARX: no more linearity in the parameters and no normal equations.

$$M(\theta) : A(z)y(t) = B(z)u(t-1) + C(z)\eta(t)$$

The procedure is still the same to find $\hat{\theta}$ from data is the same: collect data, compute $J(\theta)$ and then minimize it. The performance index $J(\theta)$ based on the prediction error still can be the mean square error. Differently from the LS method, the function is now non-convex, thus we need an iterative method to solve the minimization problem.

7.2.1 The Newton Method

Let's suppose without loss of generality, that θ is a scalar. The basic idea of this iterative procedure is to **approximate** $J(\theta)$ **with a quadratic function** $V(\theta)$. The minimum of this function for the r^{th} iteration will be considered as the estimated vector of parameters $\hat{\theta}^{r+1}$ of the following iteration.
By letting $r \to \infty$ we obtain the minimum $\hat{\bar{\theta}}$. But there is no guarantee that the minimum found is a global minimum. One simple method to deal with this problem is to execute multiple times the algorithm with different initializations and take the best among the different runs.
Procedure:

1. At iteration "r" we have to estimate $\theta\hat{?}r$
2. From $\hat{\theta}^r$ we can obtain $A^r(z), B^r(z), C^r(z)$
3. We can obtain $\epsilon^r(t)$ by means of those: $C^r(z)\epsilon^r(t) = A^r(z)y(t) - B^r(z)u(t-1)$
4. Filter data to obtain the gradient vector $\Psi^r(t) = -\frac{d\epsilon^r(t)}{d\theta}$
5. Use Gauss-Newton formula to compute $\hat{\theta}^{(r+1)} = \theta^{(r)} + (\sum_t \Psi(t)\Psi^T(t))^-1 \sum_t \Psi(t)\epsilon(t)$
6. Repeat until convergence

7.3 Performance of prediction error identification methods

If we construct the prediction error as usual:

$$\epsilon_\theta(t) = y(t) - \hat{y}_\theta(t)$$

Both $y(t)$ and $\hat{y}_\theta(t)$ are sequence of points. Thus the performance index $J(\theta)$ depends on specific points that are provided. To highlight it, we add the subscript N:

$$J_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} \epsilon_\theta(t)^2$$

Then also the estimated parameters $\hat{\theta_N}$ depends on the data points.
If the prediction error can be seen as a stationary process,we expect that with $N \to \infty$:

$$J_N(\theta) \to \bar{J}(\theta) = Var[\epsilon_\theta(t)]$$
$$\hat{\theta}_N \to \bar{\theta}$$

$\bar{J}(\theta)$ does not depend on the particular outcome of the random experiment and so will have a unique minimum.

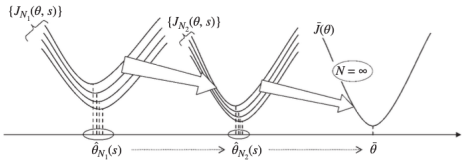


Figure 1: When $N \to \infty$ we have a unique deterministic curve

The performance of a prediction error identification method can be studied asymptotically by analysing the properties of $M(\bar{\theta})$. Having a system $S \in M(\theta)$, we can define $S = M(\theta^\cdot)$ with $\theta^\cdot$ fixed and we can study the rate of convergence of $\hat{\theta_N}$ to $\theta^\cdot$ by performing the variance of their difference. Example with LS:

$$Var[\hat{\theta_N} - \theta^\cdot] - \frac{1}{N}[\frac{1}{N}\sum_t \phi(t)\phi^T(t)]^{-1}\lambda^2$$

$$\text{with } \lambda^2 = \frac{1}{N}\sum_t \epsilon_\theta(t)^2$$

8 Model Complexity Selection

After computing $\hat{\theta}_N$, we have $J_N(\hat{\theta}_N)^{(n)}$ for a model of order n. How can we select the best value of n, namely the best model complexity?

- **Naive approach**: compute the performance index J_N for multiple increasing values of n until we find the best performance
- **Cross Validation**: divide the dataset with identification and validation set; use the former to build the model and evaluate the performance index using the validation set (we are wasting some of the data, because they canno be used in the identification process)
- **Final Prediction Error**: $FPE = \frac{N+n}{N-n} J_N(\hat{\theta}_N)^{(n)}$
  We are giving a penalty to the models with high complexity. The FPE functions is not monotonically decreasing, and the complexity corresponding to its minimum value can be chosen as complexity of the model
- **Akaike Information Criterion**: $AIC = \frac{2n}{N} + \ln J_N(\hat{\theta}_N)^{(n)}$
  For high values of N, this is equivalent to FPE.
  The first term is the one regarding the complexity of the model, while the second is the one regarding the fitting of the data
- **Minimum Description Length**: $MDL = \frac{n}{N}\ln N + \ln J_N(\hat{\theta}_N)^{(n)}$
  Asymptotically is simliar to AIC but with lower penalization $\to$ MDL leads to more parsimonious models

9 Durbin-Levinson algorithm

A recursive algorithm to estimate the parameters of an $AR(n+1)$ starting from an $AR(n)$ model. Using this algorithm we avoid to invert several matrices, which is an expensive procedure. Procedure for $AR(n) \to AR(n+1)$:

$$a_{n+1}^{(n+1)} = \frac{1}{\lambda_{(n)}^2}\left[\gamma(n+1) - \sum_1^n a_i^{(n)}\gamma(n+1-i)\right]$$

$$a_i^{(n+1)} = a_i^{(n)} - a_{n+1}^{(n+1)} a_{n+1-i}^{(n)},$$

$$\lambda_{(n+1)}^2 = \left[1 - (a_{n++1}^{(n+1)})^2\right]\lambda_{(n)}^2.$$

9.1 Time series analysis made easy

Now we can see easily estimate the order of MA and AR processes.

9.1.1 MA Models

We know that $\gamma_y(\tau) = 0$ where $\tau > n$. From that we can determine the order of our MA model by finding the value of τ for which the covariance function goes to zero.

9.1.2 AR Models

In this case the $PARCOV(\tau)$ function may be useful. Considering the two models:

$$AR(k-1) : y(t) = a_1^{(k-1)} y(t-1) + ... + a_{k-1}^{(k-1)} y(t-k+1) + \eta(t)$$
$$AR(k) : y(t) = a_1^{(k)} y(t-1) + ... + a_k^{(k)} y(t-k) + \eta(t)$$
$$PARCOV(\tau) = a_\tau^{(\tau)}$$

Namely, the PARtial COVariance function is the last identifier parameter a_τ of an $AR(\tau)$ mode. One then can estimate $\gamma(\tau)$ and then $PARCOV(\tau)$ to finally decide the order of the system. Indeed, if the order of the "true" AR model is n then:

$$PARCOV(\tau) = 0 \ \forall \tau > n$$

9.1.3 More on PARCOV

In general the partial covariance function can be used to determine if the model to be used should be an AR or an MA:

- if, at a certain point, the covariance function goes to zero before the partial covariance does, the process is an MA
- if the partial covariance function goes to zero first, the process is an AR

10 Recursive Least Squares - RLS

All the algorithms seen so far are batch methods that work "*offline*", that use all the data at once. Now we will see a recursive method that is able to update the estimate by adding new data. The latter methods help to overcome the limitation of when the data are coming some at a time, thus this method can work "***online***".