# Learnig and Estimation of Dynamical Systems

Dante Piotto

Spring semester 2023

# Chapter 1

# Introduction

Goal of the course: build matemathical models of dynamical systems from data
Password of slides: LEDS$2023
Office appointments to be decided with professor as needed

## 1.1 What is a system?

A slice of reality whose evoultion in time can be described by a certain number of measurable attributes (variables)
Inputs: independent variables(causes) which describe the action the surrounding environment on the system
Outputs: dependent variables (effects) which describe the reactoin of the syste
Mathematica model: a set of quantitative relationships between the system variables
Solving probelms in scientific disciplicnes by means of mathematical models:

1. determination of a mathematical model of the system
2. solution of the problem by using the model (i.e. in the mathematical world)
3. implementation fo the obtained solution on the real process

Competent model: a good model for solving a given problem in a certain problem context

- different mathematical models can be associated with the same system
- classification of models based on the modeling objectives

modeling objectives:

- inference

- control
- prediction
- filtering
- diagnosis
- predictive maintenance
- simulation
- speech and image recognition

### 1.1.1 Learning modesl from data

Data (set of samples)

$$u(1), u(2), \ldots, u(N) \qquad y(1), y(2), \ldots, y(N) \qquad u(t) \in \mathcal{U}, y(t) \in \mathcal{Y}$$

Target model(function)

$$\mathcal{M}_p(\theta)$$

$\mathcal{M}_p(\cdot)$ represents a function linking input and output samples, $\theta$ is a set of parameters and $p$ is a set of hyperparameters
Static models: $f : \mathcal{U} \to \mathcal{Y}$
Dynamic models: $f : (\mathcal{U}, \mathcal{Y}) \to \mathcal{Y}$

## 1.2 Types of learning

Supervised learning ($u, y$ known):

- $\mathcal{Y}$ is discrete: classification
- $\mathcal{Y}$ is continuous: regression

Unsupervised learning ($u, y$ uknown):

- $\mathcal{Y}$ is discrete: clustering
- $\mathcal{Y}$ is continuous: dimensionality reduction

Classification: assign the input to one of a finite number of classes
Regression: find an input-output relation
Reinforcement learning: finding suitable actions to take in a given situation in order to maximize a reward. Both $u, y$ are known but we haev to find the "optimal" output for a given input

This course deals with Supervised learning

## 1.3 fields related to learning from data

- Machine learning
- Pattern recognition
- Statistical learning
- Data mining
- System identification

### 1.3.1 System identification

System identification is the art and science of building mathematical models of dynamic systems from observed input-output data
Learning from data is a *data-driven(black box) approach*: a model is selected within a specified model class by using a selection criterion on the only basis of experimental (observed) data. No reference tot he physical nature of the system is made

- the obtained models have limited validity
- the model parameters may lack any physical meaning
- models relatively easy to construct and use
- ability to extract only some relevant aspects from complex frameworks

In contrast, *physical modeling* is a white box approach. The systems is partitiond into subsystems that aredescribed by using known laws of physics. Then, the model of the system is obtained by joining such relations.

#### Grey box approach

It often happens that a model based on physical modeling contains a number of uknown parameters: identification(learning) methods can be applied to estimate the uknown parameters.

## 1.4 Learning steps

The planned use of the model is important in designing the experiment to collect data (when possible).

# Chapter 2

# Stochastic Processes

Let us consider a random experiment, with sample space $\Omega$, and let us associate to each event $\omega_i$ in the sample space a signal $x(t, \omega_i)$. With a fixed $\omega$ we have a function of time $x(t)$, and at each fixed $t_i$, $x(\omega)$ is a random variable

Definition (discrete time stochastic process):
A function $x(t, \omega)$ where $t \in \{\dots, -2, -1 - 0, 1, 2, 1 \dots\}$ is time and $\omega \in \Omega$ is an outcome of the sample space.
$x(t, \omega_i)$ is called a *realization* of the stochastic process Given $t = t_1$ the first order cdf and pdf are:

$$F(x; t_1) = P(x(t_1) \leq x) \qquad f(x; t_1) = \frac{\delta F(x; t_1)}{\delta x}$$

Autocovariance: relation proven using linearity of the expectation operaror

## 2.1 stationary stochastic processes

A stochastic process is stationary if

$$F(x_1, x_2, \dots, x_k; t_1, t_2, \dots, t_k) = F(x_1, x_2, \dots, x_k; t_1 + \tau, t_2 + \tau, \dots, t_k + \tau) \tag{2.1}$$

$$\forall \tau, \forall k, \forall \{t_1, t_2, \dots, t_k\} \tag{2.2}$$

this property also holds for the pdf Consequences:

$$\mu_x(t) = \mu_x \tag{2.3}$$

$$\sigma_x^2(t) = \sigma_x^2 \tag{2.4}$$

$$r_x(t_1, t_2) = r_x(t_1 - t_2) = r_x(\tau) \tag{2.5}$$

$$c_x(t_1, t_2) = c_x(t_1 - t_2) = c_x(\tau) \tag{2.6}$$

A process is weakly stationary (or wide-sense stationary) if the 4 properties above hold

Toeplitz matrix: symmetric matrix with all elements belonging to a diagonal being equal.
Cross-correlation and cross-covariance can only be defined for stationary stochastic proceesses

### 2.1.1 cross-correlation and cross-covariance

## 2.2 vector stochastic processes

## 2.3 gaussian processes

## 2.4 white processes

it is not possible to define a continuous time white process

# Chapter 3

# Stochastic models

PSD= Power Spectral Density

Moving Average noise is also called pink noise Matlab considers the normalized autocorrelation.

## 3.1 Matlab stuff

rand(): generates white noise vector
cov(X): gives the variance of vector X
randn(): generates gaussian distributed white noise vector
autocorr(X): gives the normalized autocorrelation
to get the non normalized autocorrelation: autocorr(X)*cov(X)
filter(): useful to generate AR, MA, and ARMA processes

# Chapter 4

# Estmation problem

w.p.1: with probability 1 A biased estimator with small variance and a sufficiently small bias may be preferrable to an unbiased estimator with high variance. *bias-variance tradeoff*

# Chapter 5

# Linear regression

Let us consider the static model

$$y(t) = f(u(t)) + e(t) \quad \text{model class } \mathcal{M}_p(\theta)$$

If the function $f$ is linear in the parameters (elements of $\theta$), the modell can be written in the *linear regression* form:

$$y(t) = \varphi^T(t)\theta + e(t)$$

## 5.1 The Least Squares Method

available data set:

$$y(1), y(2), \ldots, y(N), u(1), u(2), \ldots, u(N)$$

If an estimate $\hat{\theta}$ were available, we would compute the misfit between $y(t)$ and its 'prediction' $\hat{y}(t)$:

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi^T(t)\hat{\theta}$$

where the error term $\epsilon(t)$ is the residual. The LS method finds the estimate $\hat{\theta}$ that minimizes the loss function

$$J(\theta) = \sum_{t=1}^{N} \varepsilon^2(t) = \sum_{t=1}^{N} (y(t) - \varphi^T(t)\hat{\theta})^2 \tag{5.1}$$

In matrix form:

$$\varepsilon = Y - \Phi\theta$$

where

$$\varepsilon = \begin{bmatrix} \varepsilon(1) \\ \varepsilon(2) \\ \vdots \\ \varepsilon(N) \end{bmatrix} \quad y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \quad \varphi = \begin{bmatrix} \varphi(1) \\ \varphi(2) \\ \vdots \\ \varphi(epsilonN) \end{bmatrix}$$

so that

$$J(\theta) = \sum_{t=1}^{N} \varepsilon^2(t) = \|\varepsilon\|^2 = \|Y - \Phi\theta\|^2 \tag{5.2}$$

The optimization problem to be solved is

$$\min_{\theta \in \mathcal{M}_p(\theta)} J(\theta)$$

The solution can be found by using the following relations:

$$\frac{\partial A^T x}{\partial x} = A, \quad \frac{\partial x^T A}{\partial x} = A, \quad \frac{\partial x^T A x}{\partial x} = (A + A^T)x$$

where $A$ is an $n \times n$ matrix and $x$ is a $n \times 1$ vector
From the above relations we obtain the *normal equations*

$$\Phi^T \Phi\theta = \Phi^T Y$$

*proof was covered in class*

$$\begin{aligned} J(\theta) &= Y^T Y - 2Y^T \Phi\theta + \theta^T \Phi^T \Phi\theta \\ \frac{\partial J(\theta)}{\partial \theta} &= 0 - 2\Phi^T Y + (\Phi^T \Phi + \Phi^T \Phi)\theta \\ \frac{\partial J(\theta)}{\partial \theta} &= -2\Phi^T Y + 2\Phi^T \Phi\theta \\ \frac{\partial J(\theta)}{\partial \theta} = 0 &\implies \Phi^T \Phi\theta = \Phi^T Y \end{aligned}$$

To complete the proof we must prove that the second derivative of the matrix is positive definite

$$\frac{\partial^2 J(\theta)}{\partial \theta^2} = 2\Phi^T \Phi$$

If the $N \times p$ matrix $\Phi$ is tall ($N > p$) and full rank, the LS estimate is given by

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{5.3}$$

### 5.1.1 Geometrical interpretation of the LS estimate

The estimated function is such that the sum of the squares of the distances, evaluated along the $y$-axis, between $y(t)$ and its 'prediction' $\hat{y}(t)$ is minimized. Consider the linear map described by $\Phi$:

$$\Phi : \mathbb{R}^p \to \mathbb{R}^N$$

Let $\Phi_1, \Phi_2, \ldots, \Phi_p$ be the columns of matrix $\Phi$. The LS problem consists in finding the linear combination of $\Phi_1, \Phi_2, \ldots, \Phi_p$ that approximates $Y$ as closely

as possible. Therefore, the solution is given by the orthogonal projection of $Y$ onto the subspace spanned by $\Phi_1, \Phi_2, \ldots, \Phi_p$, i.e. the orthogonal projection of $Y$ onto the image of $A$
$\hat{\theta}$ is such that $\varepsilon = Y - \Phi\hat{\theta}$ is ortogonal to $\Phi_1, \Phi_2, \ldots, \Phi_p$

$$\implies \quad \varepsilon^T \Phi = 0$$

$$\varepsilon = Y - \Phi\hat{\theta} = Y - \hat{Y}$$
$$\varepsilon^T \Phi = (Y - \Phi\hat{\theta} = Y - \hat{Y})^T \Phi = Y^T\Phi - \hat{\theta}\Phi^T\Phi$$
$$= Y^T\Phi - Y^T\Phi(\Phi^T\Phi)^{-1}\Phi^T\Phi = 0$$

The LS solution $\hat{\theta}$ can be obtained by considering the pseudoinverse of $\Phi$:

$$\hat{\theta} = \Phi^\dagger Y$$

If $\Phi$ is full rank, its pseudoinverse is just given by $\Phi^\dagger = (\Phi^T\Phi)^{-1}\Phi^T$

### 5.1.2 Statistical properties of the LS estimator

Assume the true model

$$y(t) = \varphi^T(t)\theta^* + w(t)$$

where $w(t)$ is a zero mean white process with variance $\sigma_w^2$. Let $\hat{\theta}_N$ be an estimate obtained by using N input-output samples
It follows that

$$E[\hat{\theta}] = E[(\Phi^T\Phi)^{-1}\Phi^T Y]$$
$$Y = \Phi\theta^* + w$$
$$E[\hat{\theta}]E = [(\Phi^T\Phi)^{-1}\Phi^T Y = \Phi\theta^* + w] = E[\theta^* + (\Phi^T\Phi)^{-1}\Phi^T w]$$
$$= \theta^*(\Phi^T\Phi)^{-1}\Phi^T E[w] = \theta^*$$

therefore the LS estimator is unbiased.

### 5.1.3 LS estimation fo ARX models

The closed form solution of the LS problem is obtained in a similar fashion to the FIR model case.

#### Identifiablity

- $u(t)$ has to be persistently exciting of order $\geq n$
- In order to have a well-condition matrix $H$ (i.e.) with a low condition number), it is also necessary that $n$ is not greater than the minimal order of an ARX model compatible with the data

**Statistical properties**

True model:

$$y(t) = \varphi^T(t)\theta^* + w(t)$$

It follows that

$$\hat{\theta} = \left(\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\left(\varphi^T(t)\theta^* + w(t)\right)$$

$$\hat{\theta} = \theta^* + \left(\frac{1}{N}\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\frac{1}{N}\sum_{t=1}^{N}\varphi(t)w(t)$$

The estimate is biased: $E[\hat{\theta}] \neq \theta^*$ Consisstency:

$$\lim_{N\to\infty}\hat{\theta}_N = \theta^* + \lim_{N\to\infty}\left(\frac{1}{N}\sum\varphi(t)\varphi^T(t)\right)^{-1}\frac{1}{N}\sum\varphi(t)w(t)$$

$$\lim_{N\to\infty}\hat{\theta}_N = \theta^* + E[\varphi(t)\varphi^T(t)]^{-1}E[\varphi(t)w(t)] = \theta^* - \Sigma_\varphi^{-1}r_{\varphi w}$$

where $\Sigma_\varphi^{-1} = E(\varphi(t)\varphi^T(t)$ and $r_{\varphi w}$ is the cross correlation. Because $u(t)$ is pe of order $n$, and $w(t)$ is white, $\Sigma_\varphi$ is invertible, and $r_{\varphi w} = 0$ because $y(t-n)$ does not depend on $w(t)$ for $n > 0$ and $u(t)$ in general does not depend on $w$, therefore

$$\lim_{N\to\infty}\hat{\theta} = \theta^*$$

It is also possible to prove that

$$\sqrt{N}(\hat{\theta}\theta^* \to \mathcal{N}(0, P), \text{ for } N \to \infty$$

with

$$P = \sigma_w^2\Sigma_\varphi^{-1}$$

A consistent estimate of $\sigma_w^2$ is given by

$$\sigma_w^2 = J(\hat{\theta})$$

Then, $cov(\hat{\theta})$ can be estimated as

$$\frac{\hat{\sigma}_w^2\hat{\Sigma}_\varphi^{-1}}{N} = \hat{\sigma}_w^2(H^TH)^{-1}$$

The equivalence can be derived considering

$$\hat{\Sigma}_\varphi = \frac{1}{N}\sum\varphi(t)\varphi^T(t) = \frac{H^TH}{N}$$

### 5.1.4 ARX optimal (one step ahead) predictor

True model:

$$y(t) = \varphi^T(t)\theta^* + w(t)$$

Problem: find the optimal (minimal variance) prediction of $y(t)$ given the past data $y(t-1), u(t-1), y(t-2), u(t-2), \ldots$ It is easy to show that the optimal predictor $\hat{y}(t|t-1)$ is given by

$$y(t|t-1) = \varphi^T(t)\theta^*$$

we know that

$$y(t) = -a_1^* y(t-1) - \cdots - a_n^* y(t-n) + b_1^* u(t-1) + \cdots + b_n^* u(t-n) + w(t)$$

because $w(t)$ is a white process, the best estimate we can make for it is its mean, 0.

For any other predictor $\hat{\hat{y}}(t)$

$$E[(y(t) - \hat{\hat{y}}(t))^2] \geq E[(y(t) - \hat{y}(t|t-1))^2] = \sigma_w^2$$

As a consequence, the LS estimation $\hat{\theta}$ leads to a predictive model and the residual $\varepsilon(t)$ can be seen as a prediction error. In fact, from

$$\lim_{N\to\infty} \hat{\theta} = \theta^*$$

it follows that

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta} \xrightarrow[N\to\infty]{} \theta^*$$

### 5.1.5 LS estimation of AR models

$$y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = e(t)$$

Linear regression form:

$$y(t) = \varphi^T(t)\theta + e(t)$$

where:

$$\varphi(t) = \begin{bmatrix} -y(t-1) & -y(t-2) & \cdots & -y(t-n) \end{bmatrix}^T$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}^T$$

$$Y = -H_y(n)\theta + \varepsilon$$

LS estimate:

$$\hat{\theta} = -(H_y^T(n)H_y(n))^{-1}H_y^T(n)Y$$

Statistical properties and optimale predictor: similar considerations to those of the ARX case.

## 5.2 Recursive least squares

Let $\hat{theta}(t-1)$ be a LS estimate obtained form data collected up to time $t-1$:

$$\hat{\theta}(t-1) = \left( \sum_{k=1}^{t-1} \varphi(k)\varphi^T(k) \right)^{-1} \sum_{k=1}^{t-1} \varphi(k)y(k) \qquad (1)$$

Recursive identification methods consist of updating $\hat{\theta}(t-1)$ by some "simple modification" once data at time $t$ becomes available to compute $\hat{\theta}(t)$

Starting from (1) and

$$S(t) = HH^T = \sum_{k=1}^{t} \varphi(k)\varphi^T(k) = S(t-1) + \varphi(t)\varphi^T(t)$$

$$\sum_{k=1}^{t}$$

$$varphi(k)y(k) = \sum_{k=1}^{t-1} \varphi(k)y(k) + \varphi(t)y(t)$$

$$\hat{theta}(t) = S(t)^{-1} \sum_{k=1}^{t} \varphi(k)y(k)$$

It is possible to obtain

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$$

where

$$K(t) = S(t)^{-1}\varphi(t)$$

and we can recall that

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$$

is the prediction error. computation was developed in class This leads to the following recursive least squares algorithm:

**RLS I**

1. $S(T) = S(t-1) + \varphi(t)\varphi^T(t)$
2. $K(t) = S(t)^{-1}\varphi(t)$
3. $\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$
4. $\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$

By defining $R(t) = \frac{S(t)}{t}$ it is possible to derive a RLS algorithm for

$$\hat{\theta}(t) = \left( \frac{1}{t} \sum_{k=1}^{t} \varphi(k)\varphi^T(k) \right)^{-1} \frac{1}{t} \sum_{k=1}^{t} \varphi(k)y(k)$$

in fact, $R(t)$ can be easily updated

$$R(t) = \frac{S(t)}{t} = \frac{S(t-1) + \varphi(t)\varphi^T(t)}{t} = \frac{t-1}{t} R(t-1) + \frac{\varphi(t)\varphi^T(t)}{t}$$

Following the same steps used to derive RLS I we get (computation was developed in class):

**RLS II**

1. $R(T) = \frac{t-1}{t} R(t-1) + \frac{1}{t}\varphi(t)\varphi^T(t)$
2. $K(t) = \frac{1}{t} R(t)^{-1}\varphi(t)$
3. $\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$
4. $\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$

In order to not compute a matrix inversion at each step and instead updating the inverse itself, we can rely on the following result

**Matrix inversion lemma (Woodbury identity)**

Let $A, C$ be square and invertible. Then

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

$$S(t)^{-1} = (S(t-1) + \varphi(t)\varphi^T(t) =^{-1}$$
$$A = S(t-1)$$
$$B = \varphi(t)$$
$$C = 1$$
$$D = \varphi^T(t)$$

$$S(t)^{-1} = S(t-1)^{-1} - S(t-1)^{-1}\varphi(T)(1 + \varphi^T(t)S(t-1)^{-1}\varphi(t))^{-1}\varphi^T(t)S(t-1)^{-1}$$

note that $(1 + \varphi^T(t)S(t-1)^{-1}\varphi(t))$ is a scalar, so we can write:

$$S(t)^{-1} = S(t-1)^{-1} - \frac{S(t-1)^{-1}\varphi(T)\varphi^T(t)S(t-1)^{-1}}{1 + \varphi^T(t)S(t-1)^{-1}\varphi(t)}$$

We can now modify RLS I and RLS II in order to avoid matrix inversion

**RLS III**

1. $S(T) = S(t-1)^{-1} - \frac{S(t-1)^{-1}\varphi(T)\varphi^T(t)S(t-1)^{-1}}{1+\varphi^T(t)S(t-1)^{-1}\varphi(t)}$
2. $K(t) = S(t)^{-1}\varphi(t)$
3. $\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$
4. $\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$

**RLS IV**

1. $P(T) = \frac{t}{t-1}P(t-1) - \frac{t}{t-1}\frac{P(t-1)^{-1}\varphi(T)\varphi^T(t)P(t-1)^{-1}}{t-1+\varphi^T(t)P(t-1)\varphi(t)}$
2. $K(t) = P(t)\frac{1}{t}phi(t)$
3. $\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$
4. $\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t)$

with $P(t) = R(t)^{-1}$

**Initialization**

The RLS algorithm needs to be initialized. One possibility is to start with an initial batch estimate, which will be better the more data is available. It is also possible to start with some other type of guess on the model parameters, potentially also very bad, e.g. a vector of zeros. However, a guess for the gain matrix (either $S(0), R(0), P(0)$ depending on the specific algorithm) is necessary. With the RLS IV, a good initialization would be

$$P(0) = \alpha I_p$$

If we are confident about the initial guess of the parameters, a small value of $\alpha$ is appropriate. On the flipside, if we expect the inital estimator to be bad, the value of $\alpha$ shall be quite large.

### 5.2.1 Asymptotic behaviour of the RLS algorithm

$$y(t) = \varphi^T(t)\theta^* + w(t)$$

$$\lim_{t\to\infty} \hat{\theta}(t) = \theta^* \quad \text{w.p. } 1$$

$R(t)$ in the RLS II algo is an estimate of the covariance matrix. Looking at the RLS II we can observe that from step 2, $K(t)$ approaches 0. Therefore, asymptotically, there is no correction.

$$lim_{t1\to\infty} K(t) = \frac{\Sigma_\varphi^{-1}}{\infty}\varphi(t) = 0$$

### 5.2.2 Recursive wighted least squares

A modification of the RLS algorithm aimed at tracking parameter variations by giving less importance to past data and more importance to recenet data.

$$J(\theta) = \sum_{t=1}^{N} \lambda^{N-t} \varepsilon^2(t) = \varepsilon^T W \varepsilon$$

where

$$W = \text{diag}\begin{bmatrix} \lambda^{N-1} & \lambda^{N-2} & \cdots & \lambda & 1 \end{bmatrix}$$

and $\lambda, 0 < \lambda < 1$ is the *forgetting factor*. We have

$$\hat{\theta} = \left( \sum_{t=1}^{N} \lambda^{N-t} \varphi(t) \varphi^T(t) \right)^{-1} \sum_{t=1}^{N} \lambda^{N-t} \varphi(t) y(t)$$

The scalar $\lambda$ should be chosen by a trade-off between the ability to track parameter changes on one hand, and good estimation accuracy on the other hand.

Recursive weighted LS: determine a recursive form of

$$\hat{\theta}(t) = \left( \sum_{k=1}^{t} \lambda^{t-k} \varphi(k) \varphi^T(k) \right)^{-1} \sum_{k=1}^{t} \lambda^{t-k} \varphi(k) y(k)$$

Define

$$S(t) = \sum_{k=1}^{t} \lambda^{t-k} \varphi(k) \varphi^T(k) = \lambda S(t-1) + \varphi(t) \varphi^T(t)$$

Following the same reasoning useed to derive the previous recursive algorighms we can derive the RWLS algorithms

**RWLS I**

1. $S(t) = \lambda S(t-1) + \varphi(t) \varphi^T(t)$
2. $K(t) = S(t)^{-1} \varphi(t)$
3. $\varepsilon(t) = y(t) - \varphi^T(t) \hat{\theta}(t-1)$
4. $\hat{\theta}(t-1) + K(t) \varepsilon(t)$

# Chapter 6

# Prediction error methods

Let us consider the ARMAX model

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})w(t)$$

or

$$y(t)+a_1y(t-1)+\cdots+a_ny(t-n) = b_1u(t-1)+\cdots+b_nu(t-n)+w(t)+c_1w(t-1)+\cdots+c_nw(t-n)$$

Then

$$y(t) = \varphi^T(t)\theta + w(t)$$

where

$$\varphi(t) = \begin{bmatrix} -y(t-1) & \cdots & -y(t-n) & u(t-1) & \dots & u(t-n) & w(t-1) & \cdots & w(t-n) \end{bmatrix}^T$$

In a real data setting we may assume the model

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})\varepsilon(t)$$

where $\varepsilon(t)$ is the residual. Problem: the residual can be computed once the estimate $\hat{\theta}$ is available, so that it is a function of $\theta, \varepsilon(t,\theta)$

$$y(t) = \varphi^T(t,\theta)\theta + \varepsilon(t)$$

This is no longer a linear regression and the least squares method cannot be applied. If we consider

$$\frac{C(z^{-1}}{A(Z^{-1}}$$

as a disturbance $d(t)$ and only try to estimate the plant, we obtain

$$y(t) = \bar{\varphi}^T(t)\bar{\theta} + e(t)$$

and we can use the least squares method. However, $e(t)$ is now a coloured process and this estimator is not unbiased. Let us assume a true model exists $\theta^*$.

$$y(t) = \bar{\varphi}^T(t)\theta^+ + e(t)$$

$$\hat{\bar{\theta}}_{LS} = \left(\frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}(t)\bar{\varphi}^T(t)\right)\frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}(t)y(t)$$

$$= \theta^+ + \left(\frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}(t)\bar{\varphi}^T(t)\right)^{-1}\frac{1}{N}\sum_{t=1}^{N}\bar{\varphi}(t)e(t)$$

$$\lim_{N\to\infty}\hat{\bar{\theta}}_{LS} = \theta^* + \Sigma_{\hat{\varphi}}^{-1}r_{\bar{\varphi}e}$$

$$r_{\bar{\varphi}e} = E[\bar{\varphi}(t)e(t)] = E[\begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n) \\ u(t-1) \\ \vdots \\ u(t-n) \end{bmatrix} e(t)]$$

$$e(t) = w(t)c_1w(t-1) + \cdots + c_nw(t-n)$$

$$r_{\bar{\varphi}e} = \begin{bmatrix} \\ \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ with n zeros}$$

$y(t-1) = f(w(t-1), w(t-2, \dots)$
$e(t) = f(w(t), w(t-1), \dots, w(t-n)$ so there is a correlation between the terms of $y$ and those of $e$, so the first elements of the above vector are not zero, therefore

$$\lim_{N\to\infty}\hat{\bar{\theta}}_{LS} = \theta^* + \Sigma_{\hat{\varphi}}^{-1}r_{\bar{\varphi}e} \neq 0$$

so the estimator is biased.

General model structure:

$$y(t) = G(z^{-1})u(t) + H(z^{-1})w(t)$$

where $w(t)$ is a zero mean white process with variance $\sigma_w^2$ and uncorrelated with $u(t)$
Available measurements:

$$y(1-n), y(2-n), \dots, y(N), u(1-n), u(2-n), \dots, u(N)$$

Prediction error method: find the estimate $\hat{\theta}$ that minimizes the loss function

$$J(\theta) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon^2(t,\theta) = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \hat{y}(t|t-1,\theta))^2$$

where $\hat{y}(t|t-1,\theta)$ is the *optimal one step ahead prediction* of $y(t)$. Optimal (minimal variance) prediction: a prediction of $y(t)$ given $\theta$ and the past I/O data up to time $t-1$ s.t. the variance of the prediction error $\varepsilon(t)$ is minimal The model can be rewritten as

$$y(t) = G(z^{-1})u(t) + (H(z^{-1}) - 1)w(t) + w(t)$$

by replacing $w(t)$ with $\frac{1}{H(z^{-1})}y(t) - \frac{G(z^{-1})}{H(z^{-1})}u(t)$, after some stes we get

$$y(t) = \left(1 - \frac{1}{H(z^{-1})}\right)y(t) + \frac{G(z^{-1})}{H(z^{-1})}u(t) + w(t)$$

from which it is easy to prove

$$\hat{y}(t|t-1,\theta) = \left(1 - \frac{1}{H(z^{-1})}\right)y(t) + \frac{G(z^{-1})}{H(z^{-1})}u(t)$$

For any other predictor $y^p(t)$ we have

$$E[(y(t) - y^p(t))^2] \geq E[(y(t) - \hat{y}(t|t-1,\theta))^2]$$

$$\begin{aligned} E[(y(t) - y^p(t))^2] &= E[(Gu(t) + Hw(t) - y^p(t))^2] \\ &= E[\left(\left(1 - \frac{1}{H}\right)y + \frac{G}{H}u + w(t) - y^p(t)\right)^2] \end{aligned}$$

We can notice that $w(t)$ is uncorrelated with all other members of the sum as $y^p(t)$ depends only on the first $t-1$ samples, therefore

$$= E[\left(\left(1 - \frac{1}{H}\right)y + \frac{G}{H}u - y^p(t)\right)^2] + E[w(t)^2] = E[\left(\left(1 - \frac{1}{H}\right)y + \frac{G}{H}u - y^p(t)\right)^2] + \sigma_{w^2}$$

$$\lim_{N\to\infty} J''(\theta) = 2E[\psi(t,\theta)[\psi^T(t,\theta)]] - 2E[\varepsilon(t,\theta)\frac{\partial^2\varepsilon(t,\theta)}{\partial\theta^2}]$$

assume $\theta^*$ exists, then

### 6.0.1 Identification of ARMAX models

$$\varepsilon(t,\theta) = -c_1\epsilon(t-1,\theta) - c_2\epsilon(t-2,\theta), \cdots - c_n\epsilon(t-n,\theta) + y(t) + a_1y(t-1) + \cdots + b_nu(t-n)$$

Identification of ARARX models: the optimal predictor is

$$\hat{y}(t|t-1\theta) = (1 - a(z^{-1})D(z^{-1}))y(t) + B(z^{-1})D(z^{-1})u(t)$$

which can be seen as

$$\hat{y}(t|t-1\theta) = (1 - \bar{A}(z^{-1})) - \bar{B}(z^{-}1)u(t)$$

which is equivalent to an ARX model. One could apply LQ estimation to the model and try to identify the model by finding common roots between $\bar{A}$ and $\bar{B}$ or if just a predictive model is required LQ estimation itself is sufficient.

### 6.0.2 Statistical properties of PEM estimators

Assume that a true model exists and

- the input is persistently exciting of sufficiently high order
- The Hessian $J''(\theta)$ is nonsingular at least locally around the minimum points of $J(\theta)$

In this case, the PEM estimate is consistent (proof skipped)

$$\lim_{N\to\infty} \hat{theta} = \theta^*$$

Moreover

$$\sqrt{N}(\hat{\theta} - \theta^*) \to \mathcal{N}(0, P), \quad \text{for } N \to \infty$$

with

$$P = \sigma_w^2 (E[\psi(t, \theta^*)\psi(t, \theta^*)^T])^{-1} = \sigma_w^2 \Sigma_\psi^{-1}$$

If $w(t)$ is gaussian distributed

$$w(t) \sim \mathcal{N}(0, \sigma_w^2)$$

The PEM estimate is also asymptotically efficient, therefore $P$ is the lowest covariance matrix of the estimate that can be obtained.

### 6.0.3 MISO ARX models

$$u(t) \in \mathbb{R}^r \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}$$

$$A(z^{-1})y(t) = B_1(z^{-1})u_1(t) + \cdots + B_r u_r(t) + e(t)$$

$$B_1(z^{-1}) = b_{11}z^{-1} + b_{12}z^{-2} + \cdots + b_{1n}z^{-n}$$

$$B_2(z^{-1}) \text{ similar } P = n + rn = (r+1)n$$

$$H = \begin{bmatrix} -H_y(n) & H_{u_1}(n) & \dots & H_{u_n}(n) \end{bmatrix}$$

$$\hat{\theta}_{LS} \left( \frac{H^T H}{N} \right)^{-1} \frac{H^T Y}{N}$$

$$H^T H \theta = H^T Y$$

# Chapter 7

# Statistical hypothesis testing

# Chapter 8

# Model complexity selection and regulariztion

Model complexity selection: given a model class $\mathcal{M}(\theta)$, estimate the model complexity $p$, i.e. choose the best model class $\mathcal{M}_p(\theta)$

*Training set*: the set of data used for learning the model
*Underfitting*: the model is not rich enough to fit the data well
*Overfitting*: the model is too rich and adapts too closely to the training data
*Validation set*: the set of data used for evaluating the predictive capabilities of the models obtained with the training set in order to estimate the model complexity.

- General rule: 60-70% of the data used for training set, rest for validation
- when using the training set, the higher the model complexity, the better the data fitting. the prediction error is thus underestimated
- when dealing with real data, the loss function exhibits a monotone decrease in the training set and a *U-shape* in the validation set.

ARX MODEL:

$$\hat{\theta}_{LS} \to \theta^* \text{ for } N \to \infty$$
$$\varepsilon(t) \to w(t) \text{ for } N \to \infty$$
$$J(\hat{\theta}_{LS}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t, \hat{\theta}_{LS})^2 = \hat{\sigma}_\varepsilon^2$$
$$J(\hat{\theta}_{LS}) \to \sigma_w^2 \text{ for } N \to \infty$$
$$n = 4 \quad \theta^* = \begin{bmatrix} a_1^* & a_3^* & a_3^* & a_4^* & b_1^* & b_2^* & b_3^* & b_4^* \end{bmatrix}$$

If the validation set is used repeatedly to estimate the model complexity, the prediction error may be underestimated as well. This happens for very

complex models like neural networks. For this reason, when dealing with neural networks the dataset is split into three parts. The third set is called *test set* and is used for model assessment, i.e. for testing the predictive capabilities of the final chosen model. This requires a very large number of available samples. It is possible to design criteria that allow to estimate the model complexity by using the training test

### 8.0.1 The F-test

Let $\mathcal{M}_{p_1}(\theta), \mathcal{M}_{p_2}(\theta)$ such that $p_1 < p_2$ ($p = 2n$ for ARX models, $p = n$ for AR models etc.) Consider the test quantity

$$x = N \frac{J(\hat{\theta_N^1}) J(\hat{\theta_N^2})}{J(\hat{\theta_N^2})}$$

where $\hat{\theta}_N^1, \hat{\theta}_N^1$ are PEM (or LS) estimates: intuitively:

- $x$ large: the decrease int he loss function is significant, hence $\mathcal{M}_{p_2}(\theta)$ is better
- x small: $\mathcal{M}_{p_1}(\theta)$ and $\mathcal{M}_{p_2}(\theta)$ are almost equivalent so that $\mathcal{M}_{p_1}(\theta)$ should be chosen accorfing to the parsimony principle

How to quantify "large" and "small"?

- If $\mathcal{M}_{p_1}(\theta)$ is not large enough to include the true system:

$$J(\hat{\theta}_N^1) - J(\hat{\theta}_N^2) \text{ is } O(1) \quad x \text{ is of magnitude } N$$

- If $\mathcal{M}_{p_1}(\theta)$ is large enough:

$$x \to \chi^2(p_2 - p_1), \quad \text{for } N \to \infty$$

The following statistical test can be performed:

$$\Big\{ H_0 : \mathcal{M}_{p_1}(\theta) \text{ is suitable to describe the system} H_1 : \mathcal{M}_{p_1}(\theta) \text{ is not suitable}$$

that is:

$$\Big\{ H_0 : x \tilde{\chi}^2(p_2 - p_1) H_1 : \text{not } H_0$$

after the choice of the significance level:

$$\Big\{ x \leq \chi^2_\alpha(p_2 - p_1) \implies \text{accept } H_0 x > \chi^2_\alpha(p_2 - p_1) \implies \text{accept } H_1$$

### 8.0.2 The final prediction error (FPE) criterion

Let $\hat{\theta}_N$ be a PEM (or LS) estimate of a model of complexity $p$ and assume that it is then used to predict future data. Assume also that a true model exists and consider the prediction error variance (the expectation is with respect to future data):

$$V(V\hat{\theta}_N) = E\left[(y(t) - \hat{y}(t|t-1, \hat{\theta}_N))^2\right]$$

by replacing $y(t) = \varphi^T(t)\theta^* + w(t)$ in $V(\hat{\theta}N)$ and computing the expectation we get

$$V(\hat{\theta}_N) = \sigma_\omega^2 + (\hat{\theta}_N - \theta^*)^T \Sigma_\varphi (\hat{\theta}_N - \theta^*)$$

consider now the criterion function

$$FPE = E[V(\hat{\theta}_N)]$$

where the expectation is with respect to past data. By taking into account that:

- $$E[(\hat{\theta}_N - \theta^*)^T \Sigma_\varphi (\hat{\theta}_N - \theta^*)] = E[\text{trace}(\Sigma_\varphi (\hat{\theta}_N - \theta^*)(\hat{\theta}_N - \theta^*)^T]$$

  [1]

- Asymptotically: $\sqrt{N}(\hat{\theta}_N - \theta^*) \sim \mathcal{N}(0, \sigma_\Omega^2 \Sigma_\varphi^{-1})$

it is easy to obtain

$$FPE \approx \sigma_\omega^2 \left(1 + \frac{p}{N}\right)$$

in practice we need an asymptotically unbiased estimate of $\sigma_\omega^2$

$$\hat{\sigma}_\omega^2 = \frac{1}{N-p} \sum_{t=1}^{N} \varepsilon(t, \hat{\theta}_N)^2$$

so that

$$FPE = \hat{\sigma}_e^2 \frac{N+p}{N} = \frac{N+p}{N-p} J(\hat{\theta}_N) = \frac{N\left(1 + \frac{p}{N}\right)}{N\left(1 - \frac{p}{N}\right)} J(\hat{\theta}_N) J(\hat{\theta}_N) = J(\hat{\theta}_N) + \frac{2\frac{p}{N}}{1 - \frac{p}{N}} J(\hat{\theta}_N)$$

Note that for large $N$:

$$FPE \approx J(\hat{\theta}_N) + \frac{2p}{N} J(\hat{\theta}_N)$$

for large $N$, this criterion belongs to the family of *criteria with complexity terms* (terms that penalise complex models).

[1] E[trace($v^T A v$)] =E[trace($A v v^T$)]

### 8.0.3 Criteria with complexity terms

These criteria are obtained by penalizing in some way the decrease of $J(\hat{\theta}_N)$with increasing orders. The order giving the smallest value fo the criterion is selected. General form:

$$V(\hat{\theta}_N) = Nlog(\hat{\theta}_N) + f(N,p)$$

where $f(N,p)$ penalizes high order models.

### 8.0.4 Akaike information criterien (AIC)

$$AIC = NlogJ(\hat{\theta}_N) + 2p$$

AIC and FPE are asymptotically equivalent. They do not give consistend estimates of $n$ (the probability of overestimating the order is non-null). To get consistent estimates, the penalizing funciton must be such that

$$\begin{cases} f(N,p) = kpg(N) \\ \lim_{N\to\infty} g(N) = \infty \\ \lim_{N\to\infty} \dfrac{g(N)}{N} = 0 \end{cases}$$

**Minimum description lenght (MDL) criterion**

$$MDL = NlogJ(\hat{\theta}_N) + 2plog(N)$$

MDL leads. in general, to models of lower complexity wrt AIC and FPE. Even though the derivation is different, the MDL approach is formally equivalent to the *bayesian information criterion (BIC)*

# Chapter 9

# model assesment (validation)

Consists in evaluating the capability of the identified model to describe the process that has generated the data in a way compatible with its planned use.

Linear regression models:

$$y(t) = \varphi^T(t)\theta^* + w(t), \quad w(t) \text{ zero mean and white} \quad E[u(t)w(t-\tau)] = 0, \forall\tau$$

The PEM estimate $\hat{\theta}_N$ is consistent, then

$$\hat{\theta}_N \to \theta^* \text{ for } N \to \infty, \quad \varepsilon(t, \hat{\theta}_N) = y(t) - \varphi^T(t)\hat{\theta}_N \to w(t) \text{ for } N \to \infty$$

If we assume that our real data are well described by a linear regression model, we can make the following assumptions about the residual $\varepsilon(t, \hat{\theta}_N)$:

1. $\varepsilon(t, \hat{\theta}_N)$ is a zero mean white process
2. $\varepsilon(t, \hat{\theta}_N)$ is is uncorrelated with the input signal $u(t)$

It is thus possible to perform (on the training set) the following *test on residuals*:

- test of whiteness of $\varepsilon(t, \hat{\theta}_N)$
- test of cross-correlation between $\varepsilon(t, \hat{\theta}_N)$ and $u(t)$

## 9.1 Whiteness test

Sequence of residuals: $\varepsilon(1, \hat{\theta}_N), \varepsilon(2, \hat{\theta}_N), \ldots, \varepsilon(N, \hat{\theta}_N)$

$$\begin{cases} H_0 : \varepsilon(t, \hat{\theta}_N) \text{ is a zero mean white process} \\ H_1 : \text{ not } H_0 \end{cases}$$

Consider the sample variance $\hat{r}_\varepsilon(0)$ and the first $m$ sample autocorrelations of $\varepsilon(t)$ and define the vector

$$\hat{r}_\varepsilon = \begin{bmatrix} \hat{r}_\varepsilon(1) \\ \hat{r}_\varepsilon(2) \\ \vdots \\ \hat{r}_\varepsilon(m) \end{bmatrix}$$

Under $H_0$:

$$\hat{r}_\varepsilon(0) \to \sigma_w^2 \text{ for } N \to \infty, \quad \hat{r}_\varepsilon(\tau) \to 0 \text{ for } N \to \infty, \forall \tau \neq 0$$

and it is possible to prove that

$$\sqrt{N} \xrightarrow[N\to\infty]{\sim} \mathcal{N}(0, P), \quad P = \lim_{N\to\infty} E[N\hat{r}_\varepsilon \hat{r}_\varepsilon^T] = \sigma_w^4 I \quad (*)$$

as a consequence

$$x = N \frac{\hat{r}_\varepsilon^T \hat{r}_\varepsilon}{\hat{r}_\varepsilon^2(0)} \xrightarrow[N\to\infty]{\sim} \chi^2(m)$$

This leads to the statistical test

$$\Big\{ x \leq \chi_\alpha^2(m) \implies \text{accept } H_0 x > \chi_\alpha^2(m) \implies \text{accept } H_1$$

where $\alpha$ is the chosen significance level. Define the normalized text quantities

$$\hat{\gamma}(\tau) = \frac{\hat{r}_\varepsilon(\tau)}{\hat{r}_\varepsilon(0)}, \quad \tau = 1, 2, \ldots, m$$

From (*) it follows that:

$$\sqrt{N}\hat{\gamma}(\tau) \xrightarrow[N\to\infty]{\sim} \mathcal{N}(0, 1), \quad \tau = 1, 2, \ldots, m$$

so that a set of $m$ gaussian tests can also be performed. The whiteness model test can be of help in model order estimation.

## 9.2 Test of cross-correlation

$$\begin{cases} H_0 : \varepsilon(t\hat{\theta}_N) \text{ and } u(t) \text{ are uncorrelated} \\ H_1 : \text{ not } H_0 \end{cases}$$

Consider the following vector of sample cross correlations:

$$\hat{r}_{\varepsilon u} = \begin{bmatrix} \hat{r}_{\bar{\tau}+1} \\ \hat{r}_{\bar{\tau}+2} \\ \vdots \\ \hat{r}_{\bar{\tau}+m} \end{bmatrix}$$

Consider also the vector

$$\varphi_u(t,m) = \begin{bmatrix} u(t-1) & u(t-2) & \cdots & u(t-m) \end{bmatrix}^T$$

and the sample autocorrelation matrix $\hat{\Sigma}_u(m)$, which is an estimate of $E[\varphi_u(t,m)\varphi_u^T(t,m)]$

It is possible to prove that

$$\sqrt{N}\hat{r}_{\varepsilon u} \xrightarrow[N\to\infty]{\sim} \mathcal{N}(0,P), \quad P = \lim_{N\to\infty} E[N\hat{r}_{\varepsilon u}\hat{r}_{\varepsilon u}^T] = \sigma_w^2 \Sigma_u(m)$$

As a consequence

$$x = N\frac{\hat{r}_{\varepsilon u}^T \hat{\Sigma_u^{-1}} \hat{r}_{\varepsilon u}}{\hat{r}_{\varepsilon u}(0)} \xrightarrow[N\to\infty]{\sim} \chi^2(m)$$

This leads to the statistical test

$$\Big\{ H_0 : x \le \chi_\alpha^2(m) \implies \text{accept } H_0 H_0 : x > \chi_\alpha^2(m) \implies \text{accept } H_1$$

where $\alpha$ is the chosen significance level. $\bar{\tau}$ must3 be chosen with care. In some methods $\hat{r}_{\varepsilon u}(\tau)$ is constrained to be zero for some values of $\tau$ by construction.

# Chapter 10

# Maximum likelihood estimation

# Chapter 11

# Classification: probabilistic models