System Identification 2102531

Lecture Notes

Jitkomut Songsiri

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

Contents

Preface **v**

Notation **vi**

1 Introduction **1**

2 Reviews on dynamical systems **9**

3 Reviews on linear algebra **19**

4 Model Parametrization **46**

5 Input Signals **57**
Exercises 69

6 Linear least-squares **70**
Exercises 82

7 Significance tests for linear regression **86**

8 Variants of least-squares **94**
Exercises 103

9 Instrumental variable methods **106**
Exercises 114

10 Prediction Error Methods **115**
Exercises 125

11 Statistical Estimation **126**
Exercises 139

12 Subspace identification **142**

13 Model selection and model validation **158**
Exercises 171

14 Recursive identification **174**
Exercises 187

15 Applications of system identification 188
15.1 Rainfall Grid Interpolation from Rain Gauge and Rainfall Predicted from Satellite Data 188
15.2 Parameter estimation of Gumbel distribution for flood peak data 188
15.3 Solar Forecasting using Time Series Models 189
15.4 An Identification of Building Temperature System 191
15.5 Modeling of Photovoltaic System 191

References 193

Preface

This handout has been prepared as a teaching material for 2102531 (System Identification) course which is intended for senior undergraduate and graduate students. The contents provide firstly reviews on linear algebra, probability, statistics, linear system and random processes, which are fundamental concepts required for developing ideas in system identification processes. Core estimation techniques cover least-squares and its variants, instrumental variable, prediction error method, maximum likelihood, maximum a posteriori, minimum mean square, and subspace identification. These topics are summarized from various textbooks on system identification including L. Ljung, System Identification: Theory for the User, Soderstrom and P. Stoica, System Identification, P. Van Overschee and De Moor, Subspace identification for linear systems, R.C. Young, Recursive estimation and time-series analysis, and some textbooks on statistical learning: James and D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning with Applications in R.

The handout has been used and revised for six academic years since 2011. After 2015, we decided to spend less time on nonparametric approach but focus more on parametric methods that find more explicit examples from applications. In addition to computer problem exercises, we also have had term projects to allow students to explore more tools and estimation methods that are applicable to real-world applications. Examples of these projects are described in Chapter 15 where all of them are listed on `http://jitkomut.eng.chula.ac.th/ee531.html`. Materials of old topics that are not currently taught are also available there.

The author would like to thank Prof. Manop Wongsaisuwan for providing useful resources and comments when I was first assigned to teach this course. My thank also goes to all students in the past. Their feedback comments and study results have helped me design the content presentation that is aimed to suit them most.

Jitkomut Songsiri

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

November 12, 2018

Notation

Notation	Description
$\mathbf{R}$	set of real numbers
$\mathbf{R}^n$	set of real vectors of length n
$\mathbf{R}^{m\times n}$	set of real matrices of size $m \times n$
$\mathbf{C}$	set of complex numbers
$\mathbf{C}^n$	set of complex vectors of length n
$\mathbf{C}^{m\times n}$	set of complex matrices of size $m \times n$
$\mathbf{S}^n$	set of symmetric matrices of size $n \times n$
$\mathbf{var}$	variance of a random variable
$\mathbf{cov}$	Covariance matrix
$\mathbf{tr}$	Trace operator
$\mathbf{E}$	Expectation operator
$\mathcal{N}(T)$	nullspace of linear transformation T
$\mathcal{R}(T)$	range space of linear transformation T

Chapter 1

Introduction

Learning objectives of this chapter are

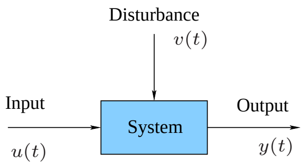
- to provide basic concepts about system identification,
- to describe pre-requisite skills for this course.

1. Introduction

- basic concept
- system identification methods
- procedures in system identification
- examples

Basic concept

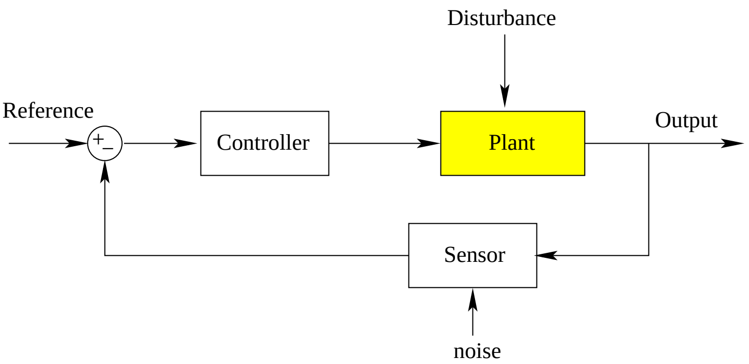
objective: how to build a system description from experimental data



estimation of system description can serve for many purposes:

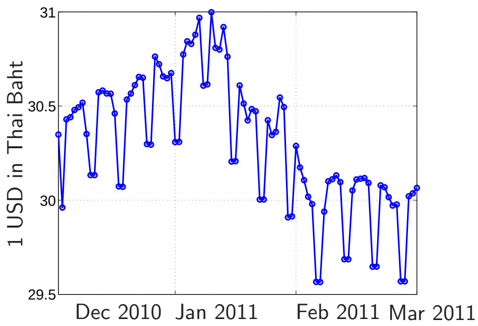
- obtain a mathematical model for controller design
- explain/understand observed phenomena (e.g., machine learning)
- forecast events in the future (e.g., time series analysis in econometrics)
- obtain a model of signal in filter design (e.g., signal processing)

System Identification for Controller Design



- for controller design, the plant is assumed known
- in system identification, we aim to estimate the parameters in a model

System Idenfitication for prediction



how to forecast the Thai Baht in Apr, May,... ?

need a model for prediction, e.g. $\hat{x}_{\mathrm{Apr}} = a_1 x_{\mathrm{Mar}} + a_2 x_{\mathrm{Feb}}$

Models

a description of the system, or a relationshop among observed signals

a model should capture the essential information about the system

Types of Models

- graph and tables, e.g., bode plots and step response
- mathematical models, e.g., differential and difference equations
- probablilistic models, e.g, probability density function

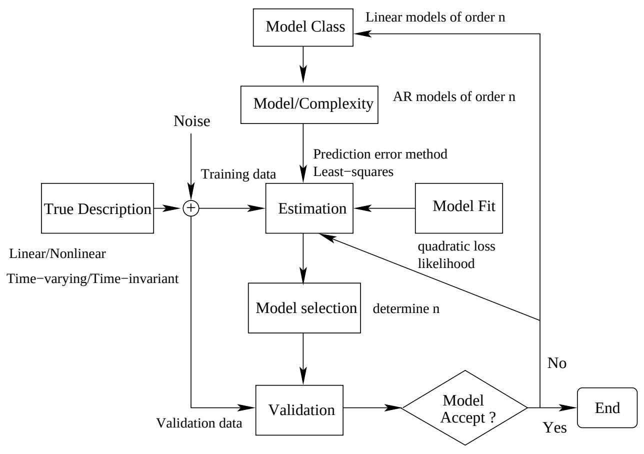
System identification is a process of obtaining models based on a data set collected from experiments

input and output signals from the system are recorded and analyzed to infer a model

System identification methods

- **Nonparametric approach**
  - aim at determining a (time/frequency) response directly without first selecting a possible set of models
  - gives basic information about the sytsem and is useful for validation
  - examples are transient analysis, frequency analysis, correlation analysis, and spectral analysis
- **Parametric approach**
  - require assumptions on a model class/structure
  - the search for the best model within the candidate set becomes a problem of determining the model parameters
  - typically more complicated than the nonparametric approach
  - results can be further used for controller design, simulation, etc.

Procedures in System Identification



Parametric Estimation

- Model classification:

  SISO/MIMO, Linear/Nonlinear, Time-invarian/Time varying, Discrete/Continuous

- searching the best model within a candidate set becomes a problem of determining the model parameters

- the selected parameter $\hat{x}$ from a model class $\mathcal{M}$ is optimal in some sense, i.e.,

$$\hat{x} = \operatorname*{argmin}_{x \in \mathcal{M}} f(x, \mathcal{D}),$$

  where f is a measure of goodness of fit (or loss function) and is a function of information data ($\mathcal{D}$)

- examples of f are quadratic loss, likelihood, entropy function, etc.

Estimation methods

- linear least-squares method (LS)

  simple to compute, no assumption on noise model

- statistical estimation methods, e.g., Maximum likelihood, Bayes

  use prior knowledge about the noise

- instrumental-variable method

  a modification of the LS method for correlated noise

- prediction-error method

  model the noise, applicable to a broad range of models

Model selection

- **Principle of parsimony:**

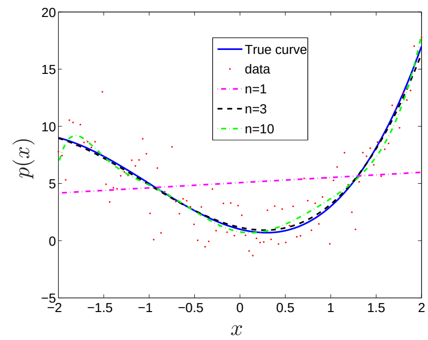
  one should pick a model with the smallest possible number of parameters that can adequately explain the data

- one can trade off between

Goodness of fit VS **Complexity**

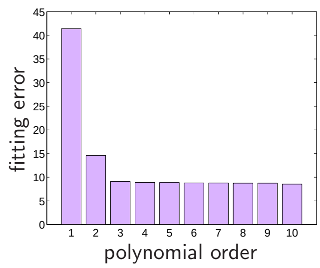
- related to the concept of bias VS variance in statistics
- examples of model selection criterions are FPE, AIC, BIC, etc.

Example: Polynomial fitting



- a curve fitting problem of polynomial of order n
- the true order is $n = 3$

Example: Trade-off curve

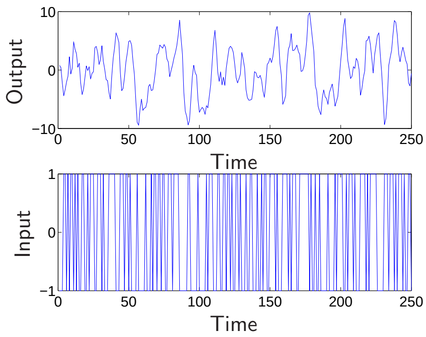


- shows the minimized loss as a decreasing function of model complexity
- the error begins to decrease as the model picks up the relevant features
- as the model order increases, the model tends to *over fit* the data
- in practice, the model order is determined by the "knee" in the curve

Model Validation

- a parametric estimation procedure picks out the *best* model
- a problem of model validation is to verify whether this best model is "good enough"
- test the estimated model (obtained from training data), with a new set of data (validation set)
- The tests verify whether the dynamic from the input and the noise model are adequate

Numerical Example



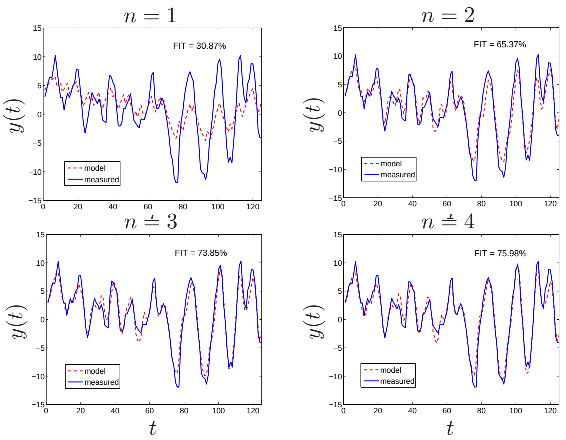
- feed a known input to the system and measure the output
- the input should contain rich information to excite the system

- fit the measured output to the model

$$(1 + a_1 q^{-1} + \cdots + a_n q^{-n}) y(t) = \\ (b_1 q^{-1} + \cdots + b_n q^{-n}) u(t) + (1 + c_1 q^{-1} + \cdots + c_n q^{-n}) \nu(t)$$

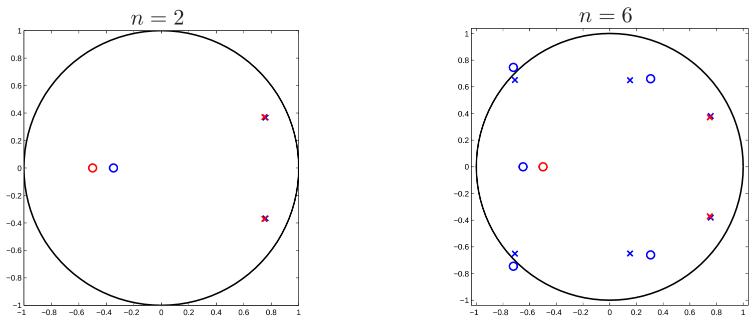
  with unknown parameters $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n$
- this model is known as *Autoregressive Moving Average with Exogenous input (ARMAX)*
- $\nu(t)$ represents the noise that enters to the system
- n is the model order, which is selected via *model selection*
- the parameters are estimated by the *Prediction-error method (PEM)*

Example of output prediction



(estimated by PEM and validated on a new data set)

Example of Zero-Pole location



- ○: zeros, ×: poles
- red: true system, blue: estimated models
- chance of zero-pole cancellation at higher order

Skills needed for System Identification

one should have

- concepts of dynamical systems (description, how to analyze their properties)
- probability and statistics (to understand probablilistic models, estimation methods, to statistically interpret results)
- linear algebra (many linear models involve matrix analysis)
- optimization (most model estimations are optimization problems)
- programming (for numerical methods to solve estimation problems)

References

Chapter 1,2 in
L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Second edition, 1999

Chapter 1-3 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

L. Ljung, *Perspective on System Identification*,
http://www.control.isy.liu.se/ljung/

Chapter 2

Reviews on dynamical systems

Students should review the topics of

- linear time-invariant system description and transfer function,
- properties of wide-sense stationary processes.

2. Reviews on dynamical systems

- linear systems: state-space equations
- random (stochastic) processes

Continous-time systems

- an autonomous system

$$\dot{x}(t) = Ax(t), \quad y = Cx(t)$$

- a system with inputs

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y = Cx(t) + Du(t)$$

- $x \in \mathbf{R}^n$ is the state, $y \in \mathbf{R}^m$ is the output, and $u \in \mathbf{R}^p$ is the control input
- $A \in \mathbf{R}^{n \times n}$ is the dynamic matrix
- $B \in \mathbf{R}^{p \times n}$ is the input matrix
- $C \in \mathbf{R}^{m \times n}$ is the output matrix
- $D \in \mathbf{R}^{m \times p}$ is the direct forward term

Solution of state-space equations

- an autonomous system

$$x(t) = e^{At}x(0), \quad y = Ce^{At}x(0)$$

e^{At} is the state-transition matrix; can be computed analytically

- a system with inputs

$$\begin{aligned} x(t) &= e^{tA}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau, \\ y(t) &= Ce^{At}x(0) + C\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau + Du(t) \end{aligned}$$

$x(t)$ consists of zero-input response and zero-state response

Discrete-time systems

- an autonomous system

$$x(t+1) = Ax(t), \quad y(t) = Cx(t)$$

  with solution

$$x(t) = A^t x(0)$$

- a system with inputs

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

  with solution

$$x(t) = A^t x(0) + \sum_{\tau=0}^{t-1} A^{t-1-\tau} Bu(\tau)$$

Transfer function of linear systems

explains a relationship from u to y

- continuous-time system: $Y(s) = H(s)U(s)$

$$H(s) = C(sI - A)^{-1}B + D$$

- discrete-time system: $Y(z) = H(z)U(z)$

$$H(z) = C(zI - A)^{-1}B + D$$

the inverse Laplace (z-) transform of H is the impulse response, $h(t)$

Important concepts of system analysis

- stability: if $x(t) \to 0$ when $t \to \infty$
  (eigenvalues of dynamic matrix, Lyapunov theory)
- controllability: how a target state can be acheived by applying a certain input
  (explained from A and B)
- observability: how to estimate $x(0)$ from the measurement y
  (explained A and C)

Stochastic Signals

- stationary processes
- ergodic processes
- correlation and covariance function
- power spectral density
- independent and uncorrelated processes
- Gaussian or normal processes
- white noise
- linear process with stochastic signals

Stochastic Processes

stochastic process is an entirely family (*ensemble*) of random time signals

$$\{x(t), \quad t \in T\}$$

$i.e.$, for each t in the index set T, $x(t)$ is a random variable

- a signal realization $x(t)$ is called *sample function* or a *sample path*
- if T is a countable set, $x(t)$ is called **discrete-time** stochastic process
- if T is a continuum, $x(t)$ is called **continuous-time** process
- a process can be either discrete- or continuous-valued

Joint distribution

let $x_1, \ldots, x_n$ be the n random variables by sampling the process $x(t)$

$$x_1 = x(t_1), \;\; x_2 = x(t_2), \ldots, \;\; x_n = x(t_n)$$

a stochastic process is specified by the collection of joint cdf (depend on time)

$$F(x_1, x_2, \ldots, x_n) = P(x(t_1) \leq x_1, \;\; x(t_2) \leq x_2, \ldots, \;\; x(t_n) \leq x_n)$$

- continuous-valued process:

$$f(x_1, \ldots, x_n) dx_1 \cdots dx_n = P(x_1 < x(t_1) \leq x_1 + dx_1, \ldots, x_n < x(t_n) \leq x_n + dx_n)$$

- discrete-valued process:

$$p(x_1, x_2, \ldots, x_n) = P(x(t_1) = x_1, x(t_2) = x_2, \ldots, x(t_n) = x_n)$$

Mean and variance of stochastic process

mean and variance function of a continous-time process are defined by

$$\begin{aligned}\mu(t) &= \mathbf{E}[x(t)] = \int_{-\infty}^{\infty} x f(x) dx \\ \mathbf{var}[x(t)] &= \int_{-\infty}^{\infty} (x - \mu(t))^2 f(x) dx\end{aligned}$$

- here f is the pdf of $x(t)$ (depend on time)
- mean and variance are *deterministic* functions of time

Correlation and Covariance

suppose X, Y are random variables with means μ_x and μ_y respectively

cross correlation

$$R_{xy} = \mathbf{E}[XY^T]$$

autocorrelation

$$R = \mathbf{E}[XX^T]$$

cross covariance

$$C_{xy} = \mathbf{E}\left[(X - \mu_x)(Y - \mu_y)^T\right]$$

autocovariance

$$C = \mathbf{E}\left[(X - \mu_x)(X - \mu_x)^T\right]$$

correlation = covariance when considering zero mean

Correlation and Covariance functions

suppose $x(t), y(t)$ are random processes

cross correlation

$$R_{xy}(t_1, t_2) = \mathbf{E} x(t_1) y(t_2)^T$$

autocorrelation

$$R(t_1, t_2) = \mathbf{E} x(t_1) x(t_2)^T$$

cross covariance

$$C_{xy}(t_1, t_2) = \mathbf{E}\left[(x(t_1) - \mu_x(t_1))(y(t_2) - \mu_y(t_2))^T\right]$$

where $\mu_x(t) = \mathbf{E} x(t)$ and $\mu_y(t) = \mathbf{E} y(t)$

autocovariance

$$C(t_1, t_2) = \mathbf{E}\left[(x(t_1) - \mu(t_1))(x(t_2) - \mu(t_2))^T\right]$$

Stationary processes

a process is called **strictly stationary** if the joint cdf of

$$x(t_1), x(t_2), \ldots, x(t_n)$$

is *the same as* that of

$$x(t_1+\tau), x(t_2+\tau), \ldots, x(t_n+\tau)$$

for *all time shifts* τ and for *all choices of sample times* $t_1, \ldots, t_k$

- first-order cdf of a stationary process must be independent of time

$$F_{x(t)}(x) = F_{x(t+\tau)}(x) = F(x), \quad \forall t, \tau$$

  implication: mean and variance are **constant** and **independent** of time

Wide-sense stationary Process

a process is **wide-sense** stationary if the two conditions hold:

1. $\mathbf{E}[x(t)] =$ constant for all t
2. $R(t_1, t_2) = R(t_1 - t_2)$ (only depends on the time gap)

the correlation/covariance functions are simplified to

$$R(\tau) = \mathbf{E}x(t+\tau)x(t)^T, \qquad R_{xy}(\tau) = \mathbf{E}x(t+\tau)y(t)^T$$

$$C(\tau) = \mathbf{E}x(t+\tau)x(t)^T - \mu_x\mu_x^T, \qquad C_{xy}(\tau) = \mathbf{E}x(t+\tau)y(t)^T - \mu_x\mu_y^T$$

Example

determine the mean and the autocorrelation of a random process

$$x(t) = A\cos(\omega t + \phi)$$

where the random variables A and ϕ are independent and ϕ is uniform on $(-\pi, \pi)$

since A and ϕ are independent, the mean is given by

$$\mathbf{E}x(t) = \mathbf{E}[A]\mathbf{E}[\cos(\omega t + \phi)]$$

using the uniform distribution in ϕ, the last term is

$$\mathbf{E}\cos(\omega t + \phi) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \cos(\omega t + \phi)d\phi = 0$$

therefore, $\mathbf{E}x(t) = 0$

using trigonometric identities, the autocorrelation is determined by

$$\mathbf{E}x(t_1)x(t_2) = \frac{1}{2}\mathbf{E}A^2\mathbf{E}[\cos\omega(t_1 - t_2) + \cos(\omega t_1 + \omega t_2 + 2\phi)]$$

since

$$\mathbf{E}[\cos(\omega t_1 + \omega t_2 + 2\phi)] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \cos(\omega t_1 + \omega t_2 + 2\phi)d\phi = 0$$

we have

$$R(t_1, t_2) = (1/2)\mathbf{E}[A^2]\cos\omega(t_1 - t_2)$$

hence, the random process in this example is wide-sense stationary

Power Spectral Density

Wiener-Khinchin Theorem: if a process is wide-sense stationary, the autocorrelation function and the power spectral density form a Fourier transform pair:

$$S(\omega) = \int_{-\infty}^{\infty} e^{-i\omega\tau}R(\tau)d\tau \qquad \text{continuous}$$

$$S(\omega) = \sum_{k=-\infty}^{k=\infty} R(k)e^{-i\omega k} \qquad \text{discrete}$$

the autocorrelation function at $\tau = 0$ indicates the average power:

$$R(0) = \mathbf{E}[x(t)x(t)^T] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)d\omega$$

(similarly, use discrete inverse Fourier transform for discrete systems)

Properties

- $R(-t) = R(t)^T$ (if the process is scalar, then $R(-t) = R(t)$)
- non-negativity: that is for any $a_i, a_j \in \mathbf{R}^n$, with $i, j = 1, \dots N$, we have

$$\sum_i^N \sum_j^N a_i^T R(i-j)a_j \geq 0,$$

  which follows from

$$\sum_i^N \sum_j^N a_i^T R(i-j)a_j = \sum_i^N \sum_j^N \mathbf{E}[a_i^T x(i)x(j)^T a_j] = \mathbf{E}\left[\left(\sum_i^N a_i^T x(i)\right)^2\right] \geq 0.$$

- $S(\omega)$ is self-adjoint, *i.e.*, $S(\omega)^* = S(\omega)$ for all ω
- diagonals of $S(\omega)$ are real-valued

Ergodic Processes

a stochastic process is *ergodic* if

$$\mathbf{E}[x(t)] = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t)dt \qquad \text{(continuous)}$$

$$\mathbf{E}[x(t)] = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} x(k) \qquad \text{(discrete)}$$

(time average = ensemble average)

- one typically gets statistical information from emsemble averaging
- ergodic hypothesis means this information can also be obtained from averaging a single sample $x(t)$ over *time*

with ergodic assumption,

continous time

$$R(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t+\tau)x(t)^T dt$$

$$R_{xy}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_{-T/2}^{T/2} x(t+\tau)y(t)^T dt$$

discrete time

$$R(\tau) = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} x(k+\tau)x(k)^T$$

$$R_{xy}(\tau) = \lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} x(k+\tau)y(k)^T$$

Independent and Correlated Processes

stationary processes $x(t)$ and $y(t)$ are called **independent** if

$$f_{XY}(x,y) = f_X(x)f_Y(y)$$

(the joint pdf is equal to the product of marginals)

and are called **uncorrelated** if

$$C_{xy}(\tau) = 0, \quad \forall\tau$$

- independent processes are always uncorrelated
- the opposite may not be true

White noise

a zero-mean process with the following properties:

continuous time

$$R(\tau) = S_0\delta(\tau), \qquad S(\omega) = \int_{-\infty}^{\infty} S_0\delta(\tau)e^{-i\omega\tau}d\tau = S_0$$

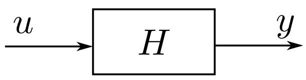
discrete time

$$R(k) = S_0\delta(k) = \begin{cases} S_0, & k = 0 \\ 0, & k \neq 0 \end{cases}, \qquad S(\omega) = \sum_{k=-\infty}^{\infty} S_0\delta(k)e^{-i\omega k} = S_0$$

(constant spectrum)

Linear systems with random input

let y be the response to input u under a linear causal system H



Facts: if $u(t)$ is a wide-sense stationary process and H is stable then

- $y(t)$ is also a wide-sense stationary process
- spectrum of u and y are related by

$$S_y(\omega) = H(\omega)S_u(\omega)H(\omega)^*$$

where $H(\omega)^*$ is the complex conjugate transpose of $H(\omega)$

Random walk

a process $x(t)$ is a random walk if

$$x(t) = x(t-1) + w(t-1)$$

where $w(t)$ is a white noise with covariance Σ

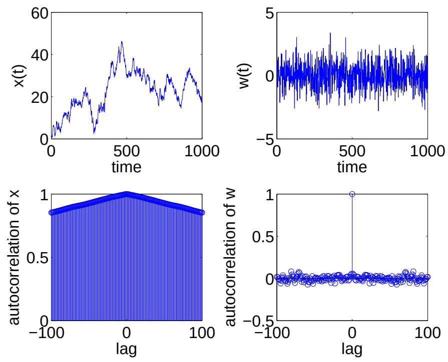
- $x(t)$ obeys a linear (unstable) system with a random input
- with back substitution, we can express $x(t)$ as

$$x(t) = w(t-1) + w(t-2) + \cdots + w(0)$$

- $x(t)$ is *non-stationary* because $R(t, t+\tau)$ depends on t

$$R(t, t+\tau) = \mathbf{E}[x(t)x(t+\tau)^T] = t\Sigma$$

time plot of random walk and its normalized *sample* autocorrelation (correlogram)



correlogram of x gradually decays

References

Chapter 9 in

A. Leon-Garcia, *Probaility, Statistics, and Random Processes for Electrical Engineering*, 3rd edition, Pearson, 2009

Chapter 3 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 3

Reviews on linear algebra

Students should review the following topics in linear algebra.

- Vectors and matrices are extensively used in our analysis. Students should be familiar with formulating problems in vector or matrix forms. Getting to know various kinds of matrix structures and understand their properties help us provide in-depth analyses of a problem.
- Methods of solving linear equations and related issues provide a basis for solving linear least-squares problems, a simple method of system identification that has numerous applications.
- Concepts of norm linear space and inner product space will be used to explain the orthogonality condition for least-squares problems.
- Different matrix factorization methods are used in solving linear equations or linear least-squares problems numerically.
- Many system identification problems have matrix variables. Functions of vectors now can be extended to functions of matrices. Basis calculus such as first and second derivatives and also the chain rule will be explained.

3. Reviews on Linear algebra

- matrices and vectors
- linear equations
- range and nullspace of matrices
- norm and inner product spaces
- matrix factorizations
- function of vectors, gradient and Hessian
- function of matrices

Vector notation

n-vector x:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- also written as $x = (x_1, x_2, \dots, x_n)$
- set of n-vectors is denoted $\mathbf{R}^n$ (Euclidean space)
- x_i: ith **element** or **component** or **entry** of x
- x is also called a column vector
- $y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}$ is called a row vector

unless stated otherwise, a vector typically means a column vector

Special vectors

zero vectors: $x = (0, 0, \dots, 0)$

all-ones vectors: $x = (1, 1, \cdots, 1)$ (we will denote it by $\mathbf{1}$)

standard unit vectors: e_k has only 1 at the kth entry and zero otherwise

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

(standard unit vectors in $\mathbf{R}^3$)

unit vectors: any vector u whose norm (magnitude) is 1, *i.e.*,

$$\|u\| \triangleq \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2} = 1$$

example: $u = (1/\sqrt{2}, 2/\sqrt{6}, -1/\sqrt{2})$

Inner products

definition: the inner product of two n-vectors x, y is

$$x_1y_1 + x_2y_2 + \cdots + x_ny_n$$

also known as the **dot product** of vectors x, y

notation: $x^T y$

properties ✎

- $(\alpha x)^T y = \alpha(x^T y)$ for scalar α
- $(x + y)^T z = x^T z + y^T z$
- $x^T y = y^T x$

Euclidean norm

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{x^T x}$$

properties

- also written $\|x\|_2$ to distinguish from other norms
- $\|\alpha x\| = |\alpha|\|x\|$ for scalar α
- $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
- $\|x\| \geq 0$ and $\|x\| = 0$ only if $x = 0$

interpretation

- $\|x\|$ measures the *magnitude* or length of x
- $\|x - y\|$ measures the *distance* between x and y

Matrix notation

an $m \times n$ matrix A is defined as

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad \text{or} \quad A = [a_{ij}]_{m \times n}$$

- a_{ij} are the **elements**, or **coefficients**, or **entries** of A
- set of $m \times n$-matrices is denoted $\mathbf{R}^{m \times n}$
- A has m rows and n columns (m, n are the **dimensions**)
- the (i, j) entry of A is also commonly denoted by A_{ij}
- A is called a **square** matrix if $m = n$

Special matrices

zero matrix: $A = 0$

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$a_{ij} = 0$, for $i = 1, \ldots, m, j = 1, \ldots, n$

identity matrix: $A = I$

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

a square matrix with $a_{ii} = 1, a_{ij} = 0$ for $i \neq j$

diagonal matrix: a square matrix with $a_{ij} = 0$ for $i \neq j$

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_n \end{bmatrix}$$

triangular matrix:

a square matrix with zero entries in a triangular part

upper triangular

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

$a_{ij} = 0$ for $i \geq j$

lower triangular

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$a_{ij} = 0$ for $i \leq j$

Block matrix notation

example: 2×2-block matrix A

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

for example, if B, C, D, E are defined as

$$B = \begin{bmatrix} 2 & 1 \\ 3 & 8 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 & 7 \\ 1 & 9 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad E = \begin{bmatrix} -4 & 1 & -1 \end{bmatrix}$$

then A is the matrix

$$A = \begin{bmatrix} 2 & 1 & 0 & 1 & 7 \\ 3 & 8 & 1 & 9 & 1 \\ 0 & 1 & -4 & 1 & -1 \end{bmatrix}$$

note: dimensions of the blocks must be compatible

Column and Row partitions

write an $m \times n$-matrix A in terms of its columns or its rows

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} b_1^T \\ b_2^T \\ \vdots \\ b_m^T \end{bmatrix}$$

- a_j for $j = 1, 2, \ldots, n$ are the columns of A
- b_i^T for $i = 1, 2, \ldots, m$ are the rows of A

example: $A = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 9 & 0 \end{bmatrix}$

$$a_1 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \quad a_2 = \begin{bmatrix} 2 \\ 9 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad b_1^T = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}, \quad b_2^T = \begin{bmatrix} 4 & 9 & 0 \end{bmatrix}$$

Matrix-vector product

product of $m \times n$-matrix A with n-vector x

$$Ax = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix}$$

- dimensions must be compatible: $\#$ columns in $A = \#$ elements in x

if A is partitioned as $A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}$, then

$$Ax = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

- Ax is a linear combination of the column vectors of A
- the coefficients are the entries of x

Product with standard unit vectors

post-multiply with a column vector

$$Ae_k = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{mk} \end{bmatrix} = \text{the } k\text{th column of } A$$

pre-multiply with a row vector

$$\begin{aligned} e_k^T A &= \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \\ &= \begin{bmatrix} a_{k1} & a_{k2} & \cdots & a_{kn} \end{bmatrix} = \text{the } k\text{th row of } A \end{aligned}$$

Trace

Definition: trace of a square matrix A is the sum of the diagonal entries in A

$$\mathbf{tr}(A) = a_{11} + a_{22} + \cdots + a_{nn}$$

example:

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 0 & -1 & 5 \\ 3 & 4 & 6 \end{bmatrix}$$

trace of A is $2 - 1 + 6 = 7$

properties ✎

- $\mathbf{tr}(A^T) = \mathbf{tr}(A)$
- $\mathbf{tr}(\alpha A + B) = \alpha\,\mathbf{tr}(A) + \mathbf{tr}(B)$
- $\mathbf{tr}(AB) = \mathbf{tr}(BA)$

Eigenvalues

$\lambda \in \mathbf{C}$ is called an **eigenvalue** of $A \in \mathbf{C}^{n\times n}$ if

$$\det(\lambda I - A) = 0$$

equivalent to:

- there exists nonzero $x \in \mathbf{C}^n$ s.t. $(\lambda I - A)x = 0$, *i.e.*,

  $$Ax = \lambda x$$

  any such x is called an **eigenvector** of A (associated with eigenvalue λ)

- there exists nonzero $w \in \mathbf{C}^n$ such that

  $$w^T A = \lambda w^T$$

  any such w is called a **left eigenvector** of A

Computing eigenvalues

- $\mathcal{X}(\lambda) = \det(\lambda I - A)$ is called the **characteristic polynomial** of A
- $\mathcal{X}(\lambda) = 0$ is called the **characteristic equation** of A
- eigenvalues of A are the root of characteristic polynomial

Properties

- if A is $n \times n$ then $\mathcal{X}(\lambda)$ is a polynomial of order n
- if A is $n \times n$ then there are n eigenvalues of A
- even when A is real, eigenvalues and eigenvectors can be complex, *e.g.*,

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 0 & 1 \\ -6 & -2 & 0 \\ 19 & 5 & -4 \end{bmatrix}$$

- if A and λ are real, we can choose the associated eigenvector to be real
- if A is real then eigenvalues must occur in complex conjugate pairs
- if x is an eigenvector of A, so is αx for any $\alpha \in \mathbf{C}$, $\alpha \neq 0$
- an eigenvector of A associated with λ lies in $\mathcal{N}(\lambda I - A)$

Important facts

denote $\lambda(A)$ an eigenvalue of A

- $\lambda(\alpha A) = \alpha\lambda(A)$ for any $\alpha \in \mathbf{C}$
- $\mathbf{tr}(A)$ is the sum of eigenvalues of A
- $\det(A)$ is the product of eigenvalues of A
- A and A^T share the same eigenvalues ✎
- $\lambda(\overline{A^T}) = \overline{\lambda(A)}$ ✎
- $\lambda(A^T A) \geq 0$
- $\lambda(A^m) = (\lambda(A))^m$ for any integer m
- A is invertible if and only if $\lambda = 0$ is not an eigenvalue of A ✎

Eigenvalue decomposition

if A is diagonalizable then A admits the decomposition

$$A = TDT^{-1}$$

- D is diagonal containing the eigenvalues of A
- columns of T are the corresponding eigenvectors of A
- note that such decomposition is not unique (up to scaling in T)

recall: A is diagonalizable iff all eigenvectors of A are independent

Inverse of matrices

Definition:

a *square* matrix A is called **invertible** or **nonsingular** if there exists B s.t.

$$AB = BA = I$$

- B is called an **inverse** of A
- it is also true that B is invertible and A is an inverse of B
- if no such B can be found A is said to be **singular**

assume A is invertible

- an inverse of A is unique
- the inverse of A is denoted by A^{-1}

assume A, B are invertible

Facts ✎

- $(\alpha A)^{-1} = \alpha^{-1} A^{-1}$ for nonzero α
- A^T is also invertible and $(A^T)^{-1} = (A^{-1})^T$
- AB is invertible and $(AB)^{-1} = B^{-1}A^{-1}$
- $(A + B)^{-1} \neq A^{-1} + B^{-1}$

Inverse of 2×2 matrices

the matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

is invertible if and only if

$$ad - bc \neq 0$$

and its inverse is given by

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

example:

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 3 \end{bmatrix}, \quad A^{-1} = \frac{1}{7} \begin{bmatrix} 3 & -1 \\ 1 & 2 \end{bmatrix}$$

Invertible matrices

☃ **Theorem:** for a square matrix A, the following statements are equivalent

1. A is invertible
2. $Ax = 0$ has only the trivial solution ($x = 0$)
3. the reduced echelon form of A is I
4. A is invertible if and only if $\det(A) \neq 0$

Inverse of special matrices

diagonal matrix

$$A = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_n \end{bmatrix}$$

a diagonal matrix is invertible iff the diagonal entries are all nonzero

$$a_{ii} \neq 0, \quad i = 1, 2, \ldots, n$$

the inverse of A is given by

$$A^{-1} = \begin{bmatrix} 1/a_1 & 0 & \cdots & 0 \\ 0 & 1/a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1/a_n \end{bmatrix}$$

the diagonal entries in A^{-1} are the inverse of the diagonal entries in A

triangular matrix:

upper triangular

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

$a_{ij} = 0$ for $i \geq j$

lower triangular

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$a_{ij} = 0$ for $i \leq j$

a triangular matrix is invertible iff the diagonal entries are all nonzero

$$a_{ii} \neq 0, \quad \forall i = 1, 2, \ldots, n$$

- product of lower (upper) triangular matrices is lower (upper) triangular
- the inverse of a lower (upper) triangular matrix is lower (upper) triangular

symmetric matrix: $A = A^T$

✎

- for any square matrix A, AA^T and A^TA are always symmetric
- if A is symmetric and invertible, then A^{-1} is symmetric
- if A is invertible, then AA^T and A^TA are also invertible

Symmetric matrix

$A \in \mathbf{R}^{n\times n}$ is called *symmetric* if $A = A^T$

Facts: if A is symmetric

- all eigenvalues of A are real
- all eigenvectors of A are orthogonal
- A admits a decomposition
$$A = UDU^T$$
where $U^TU = UU^T = I$ (U is unitary) and D is diagonal

(of course, the diagonals of D are eigenvalues of A)

Unitary matrix

a matrix $U \in \mathbf{R}^{n\times n}$ is called **unitary** if

$$U^TU = UU^T = I$$

example: $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

Facts:

- a real unitary matrix is also called **orthogonal**
- a unitary matrix is always invertible and $U^{-1} = U^T$
- columns vectors of U are mutually orthogonal
- norm is preserved under a unitary transformation:

$$y = Ux \quad \Longrightarrow \quad \|y\| = \|x\|$$

Idempotent Matrix

$A \in \mathbf{R}^{n \times n}$ is an **idempotent** (or projection) matrix if

$$A^2 = A$$

examples: identity matrix

Facts: Let A be an idempotent matrix

- eigenvalues of A are all equal to 0 or 1
- $I - A$ is idempotent
- if $A \neq I$, then A is singular

Projection matrix

a square matrix P is a **projection** matrix if and only if $P^2 = P$

- P is a linear transformation from $\mathbf{R}^n$ to a subspace of $\mathbf{R}^n$, denoted as S
- columns of P are the projections of standard basis vectors
- S is the range of P
- from $P^2 = P$, it means if P is applied twice on a vector in S, it gives the same vector
- examples:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

Orthogonal projection matrix

a projection matrix is called **orthogonal** if and only if $P = P^T$

- P is bounded, $i.e.$, $\|Px\| \leq \|x\|$

$$\|Px\|_2^2 = x^T P^T P x = x^T P^2 x = x^T P x \leq \|Px\| \|x\|$$

  (by Cauchy-Schwarz inequality – more on this later)
- if P is an orthogonal projection onto a line spanned by a unit vector u,

$$P = uu^T$$

  (we see that $\mathbf{rank}(P) = 1$ as the dimension of a line is 1)
- another example: $P = A(A^T A)^{-1} A^T$ for any matrix A

Nilpotent matrix

$A \in \mathbf{R}^{n\times n}$ is *nilpotent* if

$$A^k = 0, \quad \text{for some positive integer } k$$

Example: any triangular matrices with 0's along the main diagonal

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad \text{(shift matrix)}$$

also related to deadbeat control for linear discrete-time systems

Facts:

- the characteristic equation for A is $\lambda^n = 0$
- all eigenvalues are 0

Positive definite matrix

a symmetric matrix A is **positive semidefinite**, written as $A \succeq 0$ if

$$x^T A x \geq 0, \quad \forall x \in \mathbf{R}^n$$

and **positive definite**, written as $A \succ 0$ if

$$x^T A x > 0, \quad \text{for all } \textit{nonzero } x \in \mathbf{R}^n$$

Facts: $A \succeq 0$ if and only if

- all eigenvalues of A are non-negative
- all principle minors of A are non-negative

example: $A = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \succeq 0$ because

$$\begin{aligned} x^T A x &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= x_1^2 + 2x_2^2 - 2x_1x_2 \\ &= (x_1 - x_2)^2 + x_2^2 \geq 0 \end{aligned}$$

or we can check from

- eigenvalues of A are 0.38 and 2.61 (real and positive)
- the principle minors are 1 and $\begin{vmatrix} 1 & -1 \\ -1 & 2 \end{vmatrix} = 1$ (all positive)

note: $A \succeq 0$ does not mean all entries of A are positive!

Properties: if $A \succeq 0$ then

- all the diagonal terms of A are nonnegative
- all the leading blocks of A are positive semidefinite
- $BAB^T \succeq 0$ for any B
- if $A \succeq 0$ and $B \succeq 0$, then so is $A + B$
- A has a square root, denoted as a symmetric $A^{1/2}$ such that

$$A^{1/2}A^{1/2} = A$$

Schur complement

a consider a symmetric matrix X partitioned as

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

Schur complement of A in X is defined as

$$S_1 = C - B^T A^{-1} B, \quad \text{if} \;\; \det A \neq 0$$

Schur complement of C in X is defined as

$$S_2 = A - BC^{-1}B^T, \quad \text{if} \;\; \det C \neq 0$$

we can show that

$$\det X = \det A \det S_1 = \det C \det S_2$$

Schur complement of positive definite matrix

Facts:

- $X \succ 0$ if and only if $A \succ 0$ and $S_1 \succ 0$
- if $A \succ 0$ then $X \succeq 0$ if and only if $S_1 \succeq 0$

analogous results for S_2

- $X \succ 0$ if and only if $C \succ 0$ and $S_2 \succ 0$
- if $C \succ 0$ then $X \succeq 0$ if and only if $S_2 \succeq 0$

Linear equations

a general linear system of m equations with n variables is described by

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

where a_{ij}, b_j are constants and $x_1, x_2, \ldots, x_n$ are unknowns

- equations are linear in $x_1, x_2, \ldots, x_n$
- existence and uniqueness of a solution depend on a_{ij} and b_j

Linear equation in matrix form

the linear system of m equations in n variables

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots &= \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

in matrix form: $Ax = b$ where

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Three types of linear equations

- **square** if $m = n$ (A is square)

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- **underdetermined** if $m < n$ (A is fat)

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- **overdetermined** if $m > n$ (A is skinny)

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Existence and uniqueness of solutions

existence:

- no solution
- a solution exists

  uniqueness:

  - the solution is unique
  - there are infinitely many solutions

every system of linear equations has zero, one, or infinitely many solutions

there are no other possibilities

Nullspace

the **nullspace** of an $m \times n$ matrix is defined as

$$\mathcal{N}(A) = \{x \in \mathbf{R}^n \mid Ax = 0\}$$

- the set of all vectors that are mapped to zero by $f(x) = Ax$
- the set of all vectors that are orthogonal to the rows of A
- if $Ax = b$ then $A(x + z) = b$ for all $z \in \mathcal{N}(A)$
- also known as **kernel** of A
- $\mathcal{N}(A)$ is a subspace of $\mathbf{R}^n$ ✎

Zero nullspace matrix

- A has a zero nullspace if $\mathcal{N}(A) = \{0\}$
- if A has a zero nullspace and $Ax = b$ is solvable, the solution is unique
- columns of A are independent

✌ **equivalent conditions:** $A \in \mathbf{R}^{n \times n}$

- A has a zero nullspace
- A is invertible or nonsingular
- columns of A are a basis for $\mathbf{R}^n$

Range space

the **range** of an $m \times n$ matrix A is defined as

$$\mathcal{R}(A) = \{y \in \mathbf{R}^m \mid y = Ax \text{ for some } x \in \mathbf{R}^n\}$$

- the set of all m-vectors that can be expressed as Ax
- the set of all linear combinations of the columns of $A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$

$$\mathcal{R}(A) = \{y \mid y = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n, \quad x \in \mathbf{R}^n\}$$

- the set of all vectors b for which $Ax = b$ is solvable
- also known as the **column space** of A
- $\mathcal{R}(A)$ is a subspace of $\mathbf{R}^m$ ✎

Full range matrices

A has a full range if $\mathcal{R}(A) = \mathbf{R}^m$

✌ **equivalent conditions:**

- A has a full range
- columns of A span $\mathbf{R}^m$
- $Ax = b$ is solvable for *every* b
- $\mathcal{N}(A^T) = \{0\}$

Rank and Nullity

rank of a matrix $A \in \mathbf{R}^{m \times n}$ is defined as

$$\mathbf{rank}(A) = \dim \mathcal{R}(A)$$

nullity of a matrix $A \in \mathbf{R}^{m \times n}$ is

$$\mathbf{nullity}(A) = \dim \mathcal{N}(A)$$

Facts ✌

- $\mathbf{rank}(A)$ is maximum number of independent columns (or rows) of A

$$\mathbf{rank}(A) \leq \min(m, n)$$

- $\mathbf{rank}(A) = \mathbf{rank}(A^T)$

Full rank matrices

for $A \in \mathbf{R}^{m \times n}$ we always have $\mathbf{rank}(A) \leq \min(m, n)$

we say A is **full rank** if $\mathbf{rank}(A) = \min(m, n)$

- for **square** matrices, full rank means nonsingular (invertible)
- for **skinny** matrices ($m \geq n$), full rank means columns are independent
- for **fat** matrices ($m \leq n$), full rank means rows are independent

Theorems

- Rank-Nullity Theorem: for any $A \in \mathbf{R}^{m \times n}$,

$$\mathbf{rank}(A) + \dim \mathcal{N}(A) = n$$

- the system $Ax = b$ has a solution if and only if $b \in \mathcal{R}(A)$
- the system $Ax = b$ has a unique solution if and only if

$$b \in \mathcal{R}(A), \quad \text{and} \quad \mathcal{N}(A) = \{0\}$$

Vector space

a vector space or linear space (over $\mathbf{R}$) consists of

- a set $\mathcal{V}$
- a vector sum $+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$
- a scalar multiplication : $\mathbf{R} \times \mathcal{V} \to \mathcal{V}$
- a distinguished element $0 \in \mathcal{V}$

which satisfy a list of properties

$\mathcal{V}$ is called a vector space over $\mathbf{R}$, denoted by $(\mathcal{V}, \mathbf{R})$

if elements, called *vectors* of $\mathcal{V}$ satisfy the following main operations:

1. **vector addition:**
$$x, y \in \mathcal{V} \quad \Rightarrow \quad x + y \in \mathcal{V}$$

2. **scalar multiplication:**
$$\text{for any } \alpha \in \mathbf{R}, x \in \mathcal{V} \quad \Rightarrow \quad \alpha x \in \mathcal{V}$$

- the definition 2 implies that a vector space contains the **zero vector**
$$0 \in \mathcal{V}$$

- the two conditions can be combined into one operation:
$$x, y \in \mathcal{V}, \;\; \alpha \in \mathbf{R} \quad \Rightarrow \quad \alpha x + \alpha y \in \mathcal{V}$$

Inner product space

a vector space with an additional structure called *inner product*

an inner product space is a vector space $\mathcal{V}$ over $\mathbf{R}$ with a map

$$\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to \mathbf{R}$$

for all $x, y, z \in V$ and all scalars $a \in \mathbf{R}$, it satisfies

- conjugate symmetry: $\langle x, y \rangle = \overline{\langle y, x \rangle}$
- linearity in the first argument:
$$\langle ax, y \rangle = a\langle x, y \rangle, \quad \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$$
- positive definiteness
$$\langle x, x \rangle \geq 0, \quad \text{and} \quad \langle x, x \rangle = 0 \Longleftrightarrow x = 0$$

Examples of inner product spaces

- $\mathbf{R}^n$
$$\langle x, y \rangle = y^T x = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n$$
- $\mathbf{R}^{m \times n}$
$$\langle X, Y \rangle = \mathbf{tr}(Y^T X)$$
- $\mathcal{L}_2(a, b)$: space of real functions defined on (a, b) for which its second-power of the absolute value is Lebesgue integrable, $i.e.$,
$$f \in \mathcal{L}_2(a, b) \quad \Longrightarrow \quad \sqrt{\int_a^b |f(t)|^2 dt} < \infty$$
the inner product of this space is
$$\langle f, g \rangle = \int_a^b f(t) g(t) dt$$

Orthogonality

let $(\mathcal{V}, \mathbf{R})$ be an inner product space

- x and y are **orthogonal**:

$$x \perp y \quad \Longleftrightarrow \quad \langle x, y \rangle = 0$$

- **orthogonal complement** in $\mathcal{V}$ of $S \subset \mathcal{V}$, denoted by $S^{\perp}$, is defined by

$$S^{\perp} = \{x \in \mathcal{V} \mid \langle x, s \rangle = 0, \ \ \forall s \in S\}$$

- $\mathcal{V}$ admits the **orthogonal decomposition**:

$$\mathcal{V} = \mathcal{M} \oplus \mathcal{M}^{\perp}$$

where $\mathcal{M}$ is a subspace of $\mathcal{V}$

Orthonormal basis

$\{\phi_n, n \geq 0\} \subset \mathcal{V}$ is an **orthonormal (ON)** set if

$$\langle \phi_i, \phi_j \rangle = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

and is called an **orthonormal basis** for $\mathcal{V}$ if

1. $\{\phi_n, n \geq 0\}$ is an ON set
2. span$\{\phi_n, n \geq 0\} = \mathcal{V}$

we can construct an orthonormal basis from the *Gram-Schmidt* orthogonalization

Orthogonal expansion

let $\{\phi_i\}_{i=1}^n$ be an orthonormal basis for a vector $\mathcal{V}$ of dimension n

for any $x \in \mathcal{V}$, we have the orthogonal expansion:

$$x = \sum_{i=1}^{n} \langle x, \phi_i \rangle \phi_i$$

meaning: we can project x into orthogonal subspaces spanned by each ϕ_i

the norm of x is given by

$$\|x\|^2 = \sum_{i=1}^{n} |\langle x, \phi_i \rangle|^2$$

can be easily calculated by the sum square of projection coefficients

Adjoint of a Linear Transformation

let $A : \mathcal{V} \to \mathcal{W}$ be a linear transformation

the **adjoint** of A, denoted by A^* is defined by

$$\langle Ax, y\rangle_{\mathcal{W}} = \langle x, A^* y\rangle_{\mathcal{V}}, \quad \forall x \in \mathcal{V}, y \in \mathcal{W}$$

A^* is a linear transformation from $\mathcal{W}$ to $\mathcal{V}$

one can show that

$$\begin{aligned} \mathcal{W} &= \mathcal{R}(A) \oplus \mathcal{N}(A^*) \\ \mathcal{V} &= \mathcal{R}(A^*) \oplus \mathcal{N}(A) \end{aligned}$$

Example

$A : \mathbf{C}^n \to \mathbf{C}^m$ and denote $A = \{a_{ij}\}$

for $x \in \mathbf{C}^n$ and $y \in \mathbf{C}^m$, and with the usual inner product in $\mathbf{C}^m$, we have

$$\begin{aligned} \langle Ax, y\rangle_{\mathbf{C}^m} &= \sum_{i=1}^m (Ax)_i \, \overline{y}_i = \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j \right) \overline{y}_i \\ &= \sum_{j=1}^n x_j \left(\sum_{i=1}^m a_{ij} \overline{y}_i \right) = \sum_{j=1}^n x_j \overline{\left(\sum_{i=1}^m \overline{a_{ij}} y_i \right)} \\ &= \sum_{j=1}^n x_j \overline{\left(\overline{A}^T y \right)_j} \triangleq \langle x, \overline{A}^T y\rangle_{\mathbf{C}^n} \end{aligned}$$

hence, $A^* = \overline{A}^T$

Basic properties of A^*

Let $A^* : \mathcal{W} \to \mathcal{V}$ be the adjoint of A

facts:

- $\langle A^* y, x\rangle = \langle y, Ax\rangle \Leftrightarrow (A^*)^* = A$
- A^* is a linear transformation
- $(\alpha A)^* = \overline{\alpha} A^*$ for $\alpha \in \mathbf{C}$
- let A and B be linear transformations, then

$$(A + B)^* = A^* + B^* \quad \text{and} \quad (AB)^* = B^* A^*$$

Normed vector space

a **normed vector space** is a vector space $\mathcal{V}$ over a **R** with a map

$$\|\cdot\| : \mathcal{V} \to \mathbf{R}$$

called **norm** that satisfies

- homogenity
$$\|\alpha x\| = |\alpha|\|x\|, \qquad \forall x \in \mathcal{V}, \forall \alpha \in \mathbf{R}$$
- triangular inequality
$$\|x + y\| \leq \|x\| + \|y\|, \qquad \forall x, y \in \mathcal{V}$$
- positive definiteness
$$\|x\| \geq 0, \quad \|x\| = 0 \Longleftrightarrow x = 0, \qquad \forall x \in \mathcal{V}$$

Cauchy-Schwarz inequality

for any x, y in an inner product space $(\mathcal{V}, \mathbf{R})$

$$|\langle x, y\rangle| \leq \|x\|\|y\|$$

moreover, for $y \neq 0$,

$$\langle x, y\rangle = \|x\|\|y\| \quad \Longleftrightarrow \quad x = \alpha y, \quad \exists \alpha \in \mathbf{R}$$

proof. for any scalar α

$$0 \leq \|x + \alpha y\|^2 = \|x\|^2 + \alpha^2\|y\|^2 + \overline{\alpha}\langle x, y\rangle + \alpha\langle y, x\rangle$$

if $y = 0$ then the inequality is trivial

if $y \neq 0$, then we can choose $\alpha = -\dfrac{\langle x, y\rangle}{\|y\|^2}$

and the C-S inequality follows

Example of vector and matrix norms

$x \in \mathbf{R}^n$ and $A \in \mathbf{R}^{m\times n}$

- 2-norm
$$\|x\|_2 = \sqrt{x^Tx} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$
$$\|A\|_F = \sqrt{\mathbf{tr}(A^TA)} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$
- 1-norm
$$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|, \quad \|A\|_1 = \textstyle\sum_{ij}|a_{ij}|$$
- ∞-norm
$$\|x\|_\infty = \max_k\{|x_1|, |x_2|, \ldots, |x_n|\}, \quad \|A\|_\infty = \max_{ij}|a_{ij}|$$

Operator norm

matrix operator norm of $A \in \mathbf{R}^{m \times n}$ is defined as

$$\|A\| = \max_{\|x\| \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

also often called **induced norm**

properties:

1. for any x, $\|Ax\| \leq \|A\|\|x\|$
2. $\|aA\| = |a|\|A\|$ (scaling)
3. $\|A + B\| \leq \|A\| + \|B\|$ (triangle inequality)
4. $\|A\| = 0$ if and only if $A = 0$ (positiveness)
5. $\|AB\| \leq \|A\|\|B\|$ (submultiplicative)

examples of operator norms

- **2-norm** or **spectral norm**

$$\|A\|_2 \triangleq \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

- **1-norm**

$$\|A\|_1 \triangleq \max_{\|x\|_1=1} \|Ax\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|$$

- **∞-norm**

$$\|A\|_\infty \triangleq \max_{\|x\|_\infty=1} \|Ax\|_\infty = \max_{i=1,\ldots,m} \sum_{j=1}^{n} |a_{ij}|$$

note that the notation of norms may be duplicative

Matrix factorizations

- LU factorization
- QR factorization
- singular value decomposition
- Cholesky factorization

LU factorization

for any $n \times n$ matrix A, it admits a decomposition

$$A = PLU$$

with row pivoting

- P permutation matrix, L unit lower triangular, U upper triangular
- the decomposition exists if and only if A is nonsingular
- it is obtained from the Gaussian elimination process

QR factorization

a tall matrix $A \in \mathbf{R}^{m \times n}$ with $m \geq n$ is decomposed as

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

- $Q \in \mathbf{R}^{m \times n}$ is an orthogonal matrix ($Q^T Q = I$)
- $R \in \mathbf{R}^{n \times n}$ is an upper triangular
- if $\mathbf{rank}(A) = n$, then n columns in $Q_1 \in \mathbf{R}^{m \times n}$ forms an orthonormal basis for $\mathcal{R}(A)$ and that R_1 is invertible
- if $\mathbf{rank}(A) < n$ then R_1 contains a zero in the diagonal
- QR is obtained by many methods, $e.g.$, Gram Schmidt, Householder transform

Singular value decomposition

ler $A \in \mathbf{R}^{m \times n}$ with $\mathbf{rank}(A) = r \leq \min(m, n)$ then

$$A = U \begin{bmatrix} \Sigma_+ & 0 \\ 0 & 0 \end{bmatrix} V^T, \quad \Sigma_+ = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix}$$

$$U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad U_1 \in \mathbf{R}^{m \times r}, U_2 \in \mathbf{R}^{m \times (m-r)}, \quad U^T U = I_m$$

$$V = \begin{bmatrix} V_1 & V_2 \end{bmatrix}, \quad V_1 \in \mathbf{R}^{n \times r}, V_2 \in \mathbf{R}^{n \times (n-r)}, \quad V^T V = I_n$$

- the singular values of A:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0, \quad p = \min(m, n)$$

  are the square root of the eigenvalues of $A^T A$

- columns of U are the eigenvectors of $A^T A$
- columns of V are the eigenvectors of AA^T
- the reduced form of SVD is $A = U_1 \Sigma_+ V_1^T$
- the Frobenious norm of A is $\|A\|_F = \mathbf{tr}(\Sigma_+)$
- $\|A\|_2$ is the maximum singular value of A
- $\mathbf{rank}(A)$ is the number of *nonzero* singular value of A

Cholesky factorization

every **positive definite** matrix A can be factored as

$$A = LL^T$$

where L is lower triangular with positive diagonal elements

- L is called the *Cholesky factor* of A
- can be interpreted as 'square root' of a positive define matrix

Derivative and Gradient

Suppose $f : \mathbf{R}^n \to \mathbf{R}^m$ and $x \in \mathbf{int\,dom}\, f$

the **derivative** (or **Jacobian**) of f at x is the matrix $Df(x) \in \mathbf{R}^{m \times n}$:

$$Df(x)_{ij} = \frac{\partial f_i(x)}{\partial x_j}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n$$

- when f is scalar-valued ($i.e.$, $f : \mathbf{R}^n \to \mathbf{R}$), the derivative $Df(x)$ is a row vector
- its transpose is called the **gradient** of the function:

$$\nabla f(x) = Df(x)^T, \qquad \nabla f(x)_i = \frac{\partial f(x)}{\partial x_i}, \quad i = 1, \ldots, n$$

  which is a column vector in $\mathbf{R}^n$

Second Derivative

suppose f is a scalar-valued function ($i.e.,\ f : \mathbf{R}^n \to \mathbf{R}$)

the second derivative or **Hessian matrix** of f at x, denoted $\nabla^2 f(x)$ is

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, n$$

example: the quadratic function $f : \mathbf{R}^n \to \mathbf{R}$

$$f(x) = (1/2)x^T P x + q^T x + r,$$

where $P \in \mathbf{S}^n, q \in \mathbf{R}^n$, and $r \in \mathbf{R}$

- $\nabla f(x) = Px + q$
- $\nabla^2 f(x) = P$

Chain rule

assumptions:

- $f : \mathbf{R}^n \to \mathbf{R}^m$ is differentiable at $x \in \mathbf{int\,dom}\, f$
- $g : \mathbf{R}^m \to \mathbf{R}^p$ is differentiable at $f(x) \in \mathbf{int\,dom}\, g$
- define the composition $h : \mathbf{R}^n \to \mathbf{R}^p$ by

$$h(z) = g(f(z))$$

then h is differentiable at x, with derivative

$$Dh(x) = Dg(f(x))Df(x)$$

special case: $f : \mathbf{R}^n \to \mathbf{R}$, $g : \mathbf{R} \to \mathbf{R}$, and $h(x) = g(f(x))$

$$\nabla h(x) = g'(f(x))\nabla f(x)$$

example: $h(x) = f(Ax + b)$

$$Dh(x) = Df(Ax + b)A \quad \Rightarrow \quad \nabla h(x) = A^T \nabla f(Ax + b)$$

example: $h(x) = (1/2)(Ax - b)^T P(Ax - b)$

$$\nabla h(x) = A^T P(Ax - b)$$

Function of matrices

we typically encounter some scalar-valued functions of matrix $X \in \mathbf{R}^{m \times n}$

- $f(X) = \mathbf{tr}(A^T X)$ (linear in X)
- $f(X) = \mathbf{tr}(X^T A X)$ (quadratic in X)

definition: the derivative of f (scalar-valued function) with respect to X is

$$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} & \cdots & \frac{\partial f}{\partial x_{1n}} \\ \frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} & \cdots & \frac{\partial f}{\partial x_{2n}} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f}{\partial x_{m1}} & \frac{\partial f}{\partial x_{m2}} & \cdots & \frac{\partial f}{\partial x_{mn}} \end{bmatrix}$$

note that the differential of f can be generalized to

$$f(X + dX) - f(X) = \langle \frac{\partial f}{\partial X}, dX \rangle + \text{higher order term}$$

Derivative of a trace function

let $f(X) = \mathbf{tr}(A^T X)$

$$\begin{aligned} f(X) &= \sum_i (A^T X)_{ii} = \sum_i \sum_k (A^T)_{ki} X_{ki} \\ &= \sum_i \sum_k A_{ki} X_{ki} \end{aligned}$$

then we can read that $\frac{\partial f}{\partial X} = A$ (by the definition of derivative)

we can also note that

$$f(X + dX) - f(X) = \mathbf{tr}(A^T(X + dX)) - \mathbf{tr}(A^T X) = \mathbf{tr}(A^T dX) = \langle dX, A \rangle$$

then we can read that $\frac{\partial f}{\partial X} = A$

- $f(X) = \mathbf{tr}(X^T A X)$

$$\begin{aligned} f(X + dX) - f(X) &= \mathbf{tr}((X + dX)^T A (X + dX)) - \mathbf{tr}(X^T A X) \\ &\approx \mathbf{tr}(X^T A dX) + \mathbf{tr}(dX^T A X) \\ &= \langle dX, A^T X \rangle + \langle AX, dX \rangle \end{aligned}$$

  then we can read that $\frac{\partial f}{\partial X} = A^T X + AX$

- $f(X) = \|Y - XH\|_F^2$ where Y and H are given

$$\begin{aligned} f(X + dX) &= \mathbf{tr}((Y - XH - dXH)^T (Y - XH - dXH)) \\ f(X + dX) - f(X) &\approx -\mathbf{tr}(H^T dX^T (Y - XH)) - \mathbf{tr}((Y - XH)^T dXH) \\ &= -\mathbf{tr}((Y - XH) H^T dX^T) - \mathbf{tr}(H (Y - XH)^T dX) \\ &= -2 \langle (Y - XH) H^T, dX \rangle \end{aligned}$$

  then we identifiy that $\frac{\partial f}{\partial X} = -2(Y - XH)H^T$

Derivative of a $\log\det$ function

let $f : \mathbf{S}^n \to \mathbf{R}$ be defined by $f(X) = \log\det(X)$

$$\begin{aligned}
\log\det(X + dX) &= \log\det(X^{1/2}(I + X^{-1/2}dXX^{-1/2})X^{1/2}) \\
&= \log\det + \log\det(I + X^{-1/2}dXX^{-1/2})X^{1/2}) \\
&= \log\det X + \sum_{i=1}^{n} \log(1 + \lambda_i)
\end{aligned}$$

where λ_i is an eigenvalue of $X^{-1/2}dXX^{-1/2})X^{1/2}$

$$\begin{aligned}
f(X + dX) - f(X) &\approx \sum_{i=1}^{n} \lambda_i \quad (\log x \approx x, \ \ x \to 0) \\
&= \mathbf{tr}(X^{-1/2}dXX^{-1/2}) \\
&= \mathbf{tr}(X^{-1}dX)
\end{aligned}$$

we identify that $\frac{\partial f}{\partial X} = X^{-1}$

References

H. Anton, *Elementary Linear Algebra*, 10th edition, Wiley, 2010

K.B. Petersen and M.S. Pedersen, et.al., *The Matrix Cookbook*, Technical University of Denmark, 2008

S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, 2004

R.A. Horn and C.R. John Son, *Matrix Analysis*, 2nd edition, Cambridge Press, 2013

Chapter 2 in

T. Katayama, *Subspace methods for system identification*, Springer, 2006

Chapter 4

Model Parametrization

In this course, we first consider the class of discrete-time linear time-invariant models. This allows us to discuss about basic methods of system identification that lead to tractable solutions. In practice, input/output data are typically measured from sampled-data systems. A stochastic general model structure in discrete-time is therefore explained. Various time series models typically used in applications such as Autoregressive Moving Average (ARMA) are special classes of the general model structure. Another representation of linear time-invariant systems is to use state-space models that cover a wide range of applications and can be estimated by a common method called subspace identification.

Learning objectives of this topic are

- to understand a general model structure of linear time-invariant systems in discrete-time,
- to explain time series models and special cases.

4. Model Parametrization

- model classification
- general model structure
- time series models
- state-space models
- uniqueness properties

Model Classification

- SISO/MIMO models
- linear/nonlinear models
- parametric/nonparametric models
- time-invariant/time-varying models
- time domain/frequency domain models
- lumped/distributed parameter models
- deterministic/stochastic models

General model structure

$$\mathcal{M}(\theta): \quad y(t) = G(q^{-1};\theta)u(t) + H(q^{-1};\theta)e(t)$$
$$\mathbf{E}e(t)e(s)^T = \Lambda(\theta)\delta(t,s)$$

- $y(t)$ is ny-dimensional output
- $u(t)$ is nu-dimensional input
- $e(t)$ is an i.i.d. random variable with zero mean (white noise)
- q^{-1} is backward shift operator
- H, G, Λ are functions of the parameter vector θ
- this model is a genearal linear model in u and e

Feasible set of parameters

θ takes the values such that

- H^{-1} and $H^{-1}G$ are asymptotically stable
- $G(0;\theta) = 0$ and $H(0;\theta) = I$
- $\Lambda(\theta) \succeq 0$

General SISO model structure

$$A(q^{-1})y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t), \quad \mathbf{E}[e(t)e(t)^T] = \lambda^2$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_p q^{-p} \\ B(q^{-1}) &= b_1 q^{-1} + b_2 q^{-2} + \cdots + b_n q^{-n} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \cdots + c_m q^{-m} \\ D(q^{-1}) &= 1 + d_1 q^{-1} + \cdots + d_s q^{-s} \\ F(q^{-1}) &= 1 + f_1 q^{-1} + \cdots + f_r q^{-r} \end{aligned}$$

note that $B(0) = 0$ (causal system)

Special cases

output error structure

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + e(t)$$

in this case $H(q^{-1};\theta) = 1$

the output error is the difference between the measurable output $y(t)$ and the model output $B(q^{-1})/F(q^{-1})u(t)$

if $A(q^{-1}) = 1$ in the general model structure

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t)$$

- G and H have no common paramater
- possible to estimate G consistently even if choice of H is not appropriate

Time series models

stationary models

- ARMAX: AutoRegressive Moving Average model with Exogenous inputs
- ARMA: AutoRegressive Moving Average model
- ARX: AutoRegressive model with Exogenous inputs
- AR: AutoRegressive model
- MA: Moving Average model

non-stationary models

- ARIMA: AutoRegressive Integrated Moving Average model
- ARCH, GARCH (not discussed here)

ARMAX models

an autoregressive moving average model with an exogenous input:

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

where

$$\begin{aligned} A(q^{-1}) &= I - (A_1 q^{-1} + \cdots + A_p q^{-p}) \\ B(q^{-1}) &= B_1 q^{-1} + B_2 q^{-2} + \cdots + B_m q^{-m} \\ C(q^{-1}) &= I + C_1 q^{-1} + \cdots + C_r q^{-r} \end{aligned}$$

and $e(t)$ is white noise with covariance Σ

the parameter vector is

$$\theta = (A_1, \ldots, A_p, B_1, \ldots, B_m, C_1, \ldots, C_r)$$

(the noise covariance could be a parameter to be estimated too)

Special cases of ARMAX models

- ARMA: $A(q^{-1})y(t) = C(q^{-1})e(t)$
- AR: $A(q^{-1})y(t) = e(t)$
- MA: $y(t) = C(q^{-1})e(t)$
- FIR: $y(t) = B(q^{-1})u(t) + e(t)$
- ARX: $A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$

applying the backward shift operator explicitly

$$\begin{aligned} y(t) = A_1 y(t-1) + \cdots + A_p y(t-p) \\ + B_1 u(t-1) + \cdots + B_m u(t-m) \\ e(t) + C_1 e(t-1) + \cdots + C_r e(t-r) \end{aligned}$$

special cases:

- autoregressive moving average (ARMA) models

$$y(t) = A_1 y(t-1) + \cdots + A_p y(t-p) + e(t) + C_1 e(t-1) + \cdots + C_r e(t-r)$$

- autoregressive (AR) models

$$y(t) = A_1 y(t-1) + \cdots + A_p y(t-p) + e(t)$$

- moving average (MA) models

$$y(t) = e(t) + C_1 e(t-1) + \cdots + C_r e(t-r)$$

  y consists of a finite sum of stationary white noise (e), so y is also stationary
- finite impulse response (FIR) models

$$y(t) = B_1 u(t-1) + \cdots + B_m u(t-m) + e(t)$$

- autoregressive with exogenous input (ARX) models

$$y(t) = A_1 y(t-1) + \cdots + A_p y(t-p) + B_1 u(t-1) + \cdots + B_m u(t-m) + e(t)$$

Equivalent representation of AR(1)

write the first-order AR model recursively

$$\begin{aligned} y(t) &= Ay(t-1) + e(t) \\ &= A(Ay(t-2) + e(t-1)) + e(t) \\ &= A^2 y(t-2) + Ae(t-1) + e(t) \\ &= A^2(Ay(t-3) + e(t-2)) + Ae(t-1) + e(t) \\ &= A^3 y(t-3) + A^2 e(t-2) + Ae(t-1) + e(t) \\ &\vdots \\ &= \sum_{k=0}^{\infty} A^k e(t-k) \end{aligned}$$

- by assuming that i) t can be extended to negative index and ii) $|\rho(A)| < 1$
- y can be represented as *infinite moving average*

State-space form of AR models

define the state variable

$$x(t) = (y(t-1), y(t-2), \dots, y(t-p))$$

the state-space form of AR model is

$$x(t+1) = \begin{bmatrix} A_1 & A_2 & \cdots & A_p \\ I & 0 & & 0 \\ \vdots & \ddots & & \vdots \\ 0 & 0 & I & 0 \end{bmatrix} x(t) + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} e(t)$$

- the characteristic polynomial of the dynamic matrix is

$$\det \tilde{A}(z) = \det(z^p - (A_1 z^{p-1} + A_2 z^{p-2} + \cdots + A_p))$$

- the AR process is stationary if its dynamic matrix $\mathcal{A}$ is stable

Non-uniqueness of MA models

consider examples of two MA models

$$\begin{aligned} y(t) &= e(t) + (1/5)e(t-1), \quad e(t) \sim \mathcal{N}(0, 25) \\ x(t) &= v(t) + 5v(t-1), \quad v(t) \sim \mathcal{N}(0, 1) \end{aligned}$$

that cannot be distinguished because of normality of the noise

- note that MA and AR processes are the inverse to each other (by swapping the role of y and e)

$$y(t) = -(1/5)y(t-1) + e(t), \quad x(t) = -5x(t-1) + v(t)$$

- an MA model is called **invertible** if it corresponds to a *causal* infinite AR representation – e.g., process with coefficient $1/5$

Properties of ARMA models

important properties of ARMA model:

$$A(q^{-1})y(t) = C(q^{-1})e(t)$$

- the process is **stationary** if the roots of the determinant of

$$A(z) = I - (A_1 z + A_2 z^2 + \cdots + A_p z^p)$$

  are outside the unit circle

- the process is said to be **causal** if it can be written as

$$y(t) = \sum_{k=0}^{\infty} \Psi(k)e(t-k), \quad \sum_{k=0}^{\infty} |\Psi(k)| \leq \infty$$

  (the process cannot depend on the future input)

- the process is **causal** if and only if the roots of the determinant of $A(z)$ lie outside the unit circle
- the process is **invertible** if the roots of the determinant of

$$C(z) = I + C_1 z + \cdots + C_r z^r$$

lie outside the unit circle

Non-stationary models

examples of non-stationarity and the use of differencing

- random walk: $x(t) = x(t-1) + w(t)$

$$z(t) \triangleq x(t) - x(t-1) = w(t)$$

$z(t)$ is white noise which is stationary

- linear static trend: $x(t) = a + bt + w(t)$

$$z(t) \triangleq x(t) - x(t-1) = b + w(t) - w(t-1)$$

$z(t)$ is a MA process

can we recover the original model from the fitted differenced series ?

Integrated model

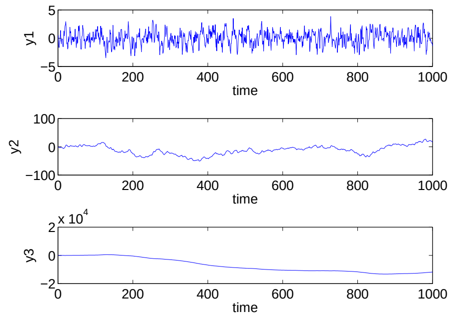
denote L a lag operator; a series $x(t)$ is **integrated** of order d if

$$(I-L)^d x(t)$$

is stationary (after d^{th} differencing)

- we use $I(d)$ to denote the integrated model of order d
- random walk is the first-order integrated model
- the lag of differencing is used to reduce a series with a trend
- for example, 12-lag of differencing removes additive seasonal effect

example: y_1 is a first-order AR process with coefficient 0.4 and is $I(0)$



- $y_2(t) = \sum_{k=0}^{t} y_1(k)$ (cumulative sum of y_1 is $I(1)$ – no exact reverting)
- $y_3(t) = \sum_{k=0}^{t} y_2(k)$ (cumulative sum of y_2 is $I(2)$ – momentum effect)

ARIMA models

$x(t)$ is an ARIMA process if the dth differences of $x(t)$ is an ARMA(p,r)

$$A(L)(I-L)^d x(t) = C(L)e(t)$$

and denoted by ARIMA(p, d, r)

examples of scalar ARIMA models

- $x(t) = x(t-1) + e(t) + ce(t-1)$ can be arranged as

$$(1-L)x(t) = (1+cL)e(t)$$

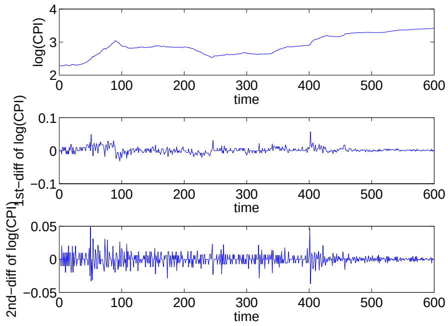
  which is ARIMA(0,1,1) or sometimes called *integrated moving average*

- $x(t) = ax(t-1) + x(t-1) - ax(t-2) + w(t)$ can be arranged as

$$(1-aL)(1-L)x(t) = w(t)$$

  which is ARIMA(1,1,0)

example: $\log$ of CPI - consumer production index and its first, second differences



- $\log$ CPI shows the momentum type – characteristics of $I(2)$
- the first difference has no momentum but no mean-reverting
- the second difference seems to be mean-reverting and behaves like white noise

State-space models

a linear stochastic model:

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + \nu(t) \\ y(t) &= C(\theta)x(t) + \eta(t) \end{aligned}$$

$\nu(t)$ and $\eta(t)$ are white noise sequences with zero means and

$$\mathbf{E} \begin{bmatrix} \nu(t) \\ \eta(t) \end{bmatrix} \begin{bmatrix} \nu(s) \\ \eta(s) \end{bmatrix}^T = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta(t,s)$$

- $\nu(t)$ is the *process noise*
- $\eta(t)$ is the *measurement noise*
- needs to transform to the so-called *innovation form* to compare with the standard model

Uniqueness properties

question: can we describe a system *adequately* and *uniquely* ?

define $\mathcal{D}$ the set of θ for which

$(\hat{G}, \hat{H}, \hat{\Lambda})$ gives a *perfect description* of the true system

three possibilities of this set can occur:

- the set $\mathcal{D}$ is empty or underparametrization
- the set $\mathcal{D}$ contains one point
- the set $\mathcal{D}$ consists of several points or overparametrization

Uniqueness properties for a scalar ARMA model

let the true ARMA model be given by

$$A(q^{-1})y(t) = C(q^{-1})e(t), \quad \mathbf{E}e(t)^2 = \lambda^2$$

$\mathcal{D}$ is the set of $\hat{A}, \hat{B}, \hat{C}, \hat{\lambda}$ for which

$$\frac{C(q^{-1})}{A(q^{-1})} = \frac{\hat{C}(q^{-1})}{\hat{A}(q^{-1})}, \quad \hat{\lambda}^2 = \lambda^2$$

in order for these equalities to have a solution, we must have

$$\deg(\hat{A}) \geq \deg(A), \quad \deg(\hat{C}) \geq \deg(C)$$

or,

$$n^* \triangleq \min\left\{\deg(\hat{A}) - \deg(A), \deg(\hat{C}) - \deg(C)\right\} \geq 0$$

- A and C have no common factor
- $\frac{C(q^{-1})}{A(q^{-1})}$ and $\frac{\hat{C}(q^{-1})}{\hat{A}(q^{-1})}$ must have the same poles and zeros

these imply

$$\hat{A}(q^{-1}) = A(q^{-1})D(q^{-1}), \quad \hat{C}(q^{-1}) = C(q^{-1})D(q^{-1})$$

where $D(q^{-1})$ has arbitrary coefficients

$$\deg(D) = \min\{\deg(\hat{A}) - \deg(A), \deg(\hat{C}) - \deg(C)\} = n^*$$

- $n^* > 0$: infinitely many solutions of $\hat{C}, \hat{A}, \hat{\lambda}$ (by varying D)
- $n^* = 0$: this gives $D(q^{-1}) = 1$, or at least one of $\hat{A}$ and $\hat{C}$ has the same degree as the true polynomial

Non-uniqueness of general state-space models

consider the multivariable model

$$\begin{aligned} x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + \nu(t) \\ y(t) &= C(\theta)x(t) + \eta(t) \end{aligned}$$

$\nu(t)$ and $\eta(t)$ are independent zero-mean white noise with covariance R_1, R_2

also consider a second model

$$\begin{aligned} z(t+1) &= \bar{A}(\theta)z(t) + \bar{B}(\theta)u(t) + \bar{\nu}(t) \\ y(t) &= \bar{C}(\theta)z(t) + \eta(t) \end{aligned}$$

where $\mathbf{E}[\bar{\nu}(t)\bar{\nu}(s)^T] = \bar{R}_1\delta(t,s)$ and

$$\bar{A} = QAQ^{-1}, \quad \bar{B} = QB, \quad \bar{C} = CQ^{-1}, \quad \bar{R}_1 = QR_1Q^T$$

for some nonsingular matrix Q

the two models are equivalent:

- they have the same transfer function from u to y

$$G(q^{-1}) = \bar{C}(qI - A)^{-1}\bar{B} = CQ^{-1}(qI - QAQ^{-1})^{-1}QB = C(qI - A)^{-1}B$$

- the outputs y from the two models have the same second-order properties, $i.e.$, the spectral densities are the same

$$\begin{aligned} S_y(\omega) &= \bar{C}(e^{\mathrm{i}\omega} - \bar{A})^{-1}\bar{R}_1(e^{\mathrm{i}\omega} - \bar{A})^{-*}\bar{C}^* + R_2 \\ &= CQ^{-1}(e^{\mathrm{i}\omega} - \bar{A})^{-1}QR_1Q^*(e^{\mathrm{i}\omega} - \bar{A})^{-*}Q^{-*}C^* + R_2 \\ &= C[Q^{-1}(e^{\mathrm{i}\omega} - \bar{A})Q]^{-1}R_1[Q^*(e^{\mathrm{i}\omega} - \bar{A})^*Q^{-*}]^{-1}C^* + R_2 \\ &= C(e^{\mathrm{i}\omega} - A)^{-1}R_1(e^{\mathrm{i}\omega} - A)^{-*}C^* + R_2 \end{aligned}$$

the model is not unique since Q can be chosen arbitrarily

Choosing a class of model structures

important factors:

- **Flexibility:** the model structure should describe most of the different system dynamics expected in the application
- **Parsimony:** the model should contain the smallest number of free parameters required to explain the data adequately
- **Algorithm complexity:** the form of model structure can considerably influence the computational cost
- **Properties of the criterion function:** for example, the asymptotic properties of prediction-error method depends crucially on the criterion function and the model structure

References

Chapter 6 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

P.S.P. Cowpertwait and A.V. Metcalfe, *Introductory Time Series with R*, Springer, 2009

R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications: with R Examples*, 3rd edition, Springer, 2011

Chapter 12-13 in
D. Ruppert and D.S. Matteson, *Statistics and Data Analysis for Financial Engineering*, 2nd edition, Springer, 2015

Chapter 5

Input Signals

From the input/output relationship (of a linear system), $y = Gu$, we need to acquire both input and output signals in order to estimate G. Obviously, applying zero input to the system yields zero output but this scheme is useless for the purpose of plant estimation. There are various patterns of common input signals such as step, ramp, square pulses, or sinusoidal inputs that are all easy to synthesize. Among these choices, we may have the following questions:

- Can any of those input signals be used in system identification?
- If there is a criterion for input signal to be satisfied, should that condition depend on the system of interest?

Learning objectives of this chapter are

- getting to know common input signals used in system identification such as step input, sum of sinusoidal waveforms, or pseudo random binary sequence (PRBS),
- to understand the properties of PRBS signal, some of which are similar to those of white noise input,
- to understand a property of input signals called *persistent exciting order* which provides information about model identifiability when such input is applied.

5. Input signals

- Common input signals in system identification
  - step function
  - sum of sinusoids
  - ARMA sequences
  - pseudo random binary sequence (PRBS)
- spectral characteristics
- persistent excitation

Step function

a step function is given by

$$u(t) = \begin{cases} 0, & t < 0 \\ u_0, & t \geq 0 \end{cases}$$

where the amplitude u_0 is arbitrarily chosen

- related to rise time, overshoots, static gain, etc.
- useful for systems with a large signal-to-noise ratio

Sum of sinusoids

the input signal $u(t)$ is given by

$$u(t) = \sum_{k=1}^{m} a_k \sin(\omega_k t + \phi_k)$$

where the angular frequencies $\{\omega_k\}$ are distinct,

$$0 \leq \omega_1 < \omega_2 < \ldots < \omega_m \leq \pi$$

and the amplitudes and phases a_k, ϕ_k are chosen by the user

Characterization of sinusoids

let S_N be the average of a sinusoid over N points

$$S_N = \frac{1}{N}\sum_{t=1}^{N} a\sin(\omega t + \phi)$$

Let μ be the mean of the sinusoidal function

$$\mu = \lim_{N\to\infty} S_N = \begin{cases} a\sin\phi, & \omega = 2n\pi, \quad n = 0, \pm 1, \pm 2, \ldots \\ 0, & \text{otherwise} \end{cases}$$

- $u(t) = \sum_{k=1}^{m} a_k \sin(\omega_k t + \phi_k)$ has zero mean if $\omega_1 > 0$
- WLOG, assume zero mean for $u(t)$ (we can always subtract the mean)

Spectrum of sinusoidal inputs

the autocorrelation function can be computed by

$$R(\tau) = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} u(t+\tau)u(t) = \sum_{k=1}^{m} C_k \cos(\omega_k \tau)$$

with $C_k = a_k^2/2$ for $k = 1, 2, \ldots, m$

if $\omega_m = \pi$, the coefficient C_m should be modified by

$$C_m = a_m^2 \sin^2 \phi_m$$

therefore, the spectrum is

$$S(\omega) = \sum_{k=1}^{m} (C_k/2)\left[\delta(\omega - \omega_k) + \delta(\omega + \omega_k)\right]$$

Autoregressive Moving Average sequence

let $e(t)$ be a pseudorandom sequence similar to white noise in the sense that

$$\frac{1}{N}\sum_{t=1}^{N} e(t)e(t+\tau) \to 0, \quad \text{as } N \to \infty$$

a general input $u(t)$ can be obtained by linear filtering

$$u(t) + c_1 u(t-1) + \cdots + c_p u(t-p) = e(t) + d_1 e(t-1) + \cdots + d_q e(t-p)$$

- u(t) is called *ARMA (autoregressive moving average)* process
- when all $c_i = 0$ it is called *MA (moving average)* process
- when all $d_i = 0$ it is called *AR (autoregressive)* process
- the user gets to choose c_i, d_i and the random generator of $e(t)$

the transfer function from $e(t)$ to $u(t)$ is

$$U(z) = \frac{D(z)}{C(z)} E(z)$$

where

$$\begin{aligned} C(z) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots c_p z^{-p} \\ D(z) &= 1 + d_1 z^{-1} + d_2 z^{-2} + \cdots d_q z^{-q} \end{aligned}$$

- the distribution of $e(t)$ is often chosen to be Gaussian
- c_i, d_i are chosen such that $C(z), D(z)$ have zeros outside the unit circle
- different choices of c_i, d_i lead to inputs with various spectral characteristics

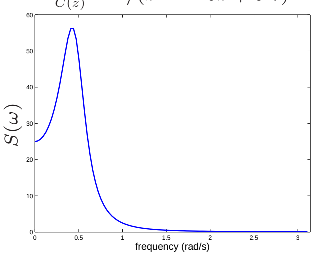
Spectrum of ARMA process

let $e(t)$ be a white noise with variance λ^2

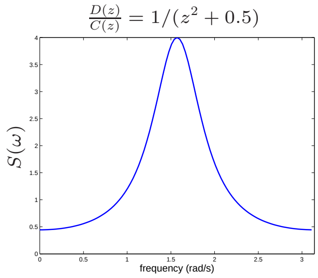
the spectral density of ARMA process is

$$S(\omega) = \lambda^2 \left| \frac{D(\omega)}{C(\omega)} \right|^2$$

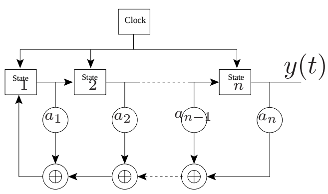
$\frac{D(z)}{C(z)} = 1/(z^2 - 1.5z + 0.7)$



$\frac{D(z)}{C(z)} = 1/(z^2 + 0.5)$



Pseudorandom Binary Sequence (PRBS)



$$x(t+1) = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ 1 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} x(t)$$

$$y(t) = \begin{bmatrix} 0 & \ldots & 0 & 1 \end{bmatrix} x(t)$$

state of PRBS satisfies a vector autoregressive equation

Characteristics of PRBS

- every initial state is allowed except the all-zero states
- the feedback coefficients $a_1, a_2, \ldots, a_n$ are either 0 or 1
- all additions are modulo-two operations
- the sequences are two-state signals (binary)
- there are possible $2^n - 1$ different state vectors x (all-zero state is excluded)
- a PRBS of period equal to $M = 2^n - 1$ is called **a maximum length PRBS** (ML PRBS)
- for *maximum length PRBS*, its characteristic resembles white random noise (pseudorandom)

Influence of the Feedback Path

let $n = 3$ and initialize x with $x(0) = (1, 0, 0)$

- with $a = (1, 1, 0)$, the state vectors $x(k), k = 1, 2, \ldots$ are

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \quad \begin{bmatrix}1\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\\1\end{bmatrix} \quad \begin{bmatrix}1\\0\\1\end{bmatrix} \quad \begin{bmatrix}1\\1\\0\end{bmatrix}$$

  the sequence has period equal to 3

- with $a = (1, 0, 1)$, the state vectors $x(k), k = 1, 2, \ldots$ are

$$\begin{bmatrix}1\\0\\0\end{bmatrix} \quad \begin{bmatrix}1\\1\\0\end{bmatrix} \quad \begin{bmatrix}1\\1\\1\end{bmatrix} \quad \begin{bmatrix}0\\1\\1\end{bmatrix} \quad \begin{bmatrix}1\\0\\1\end{bmatrix} \quad \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \begin{bmatrix}0\\0\\1\end{bmatrix} \quad \begin{bmatrix}1\\0\\0\end{bmatrix}$$

  the sequence has period equal to 7 (the maximum period, $2^3 - 1$)

Maximum length PRBS

denote q^{-1} the unit delay operator and let

$$A(q^{-1}) = 1 \oplus a_1 q^{-1} \oplus a_2 q^{-2} \oplus \cdots \oplus a_n q^{-n}$$

the PRBS $y(t)$ satisfies the homogeneous equation:

$$A(q^{-1}) y(t) = 0$$

this equation has only solutions of period $M = 2^n - 1$ *if and only if*

1. the binary polynomial $A(q^{-1})$ is irreducible, $i.e.$, there do not exist any two polynomials $A_1(q^{-1})$ and $A_2(q^{-1})$ such that

$$A(q^{-1}) = A_1(q^{-1}) A_2(q^{-1})$$

2. $A(q^{-1})$ is a factor of $1 \oplus q^{-M}$ but is not a factor of $1 \oplus q^{-p}$ for any $p < M$

Generating Maximum length PRBS

examples of polynomials $A(z)$ satisfying the previous two conditions on page 5-12

n	$A(z)$	
3	$1 \oplus z \oplus z^3$	$1 \oplus z^2 \oplus z^3$
4	$1 \oplus z \oplus z^4$	$1 \oplus z^3 \oplus z^4$
5	$1 \oplus z^2 \oplus z^5$	$1 \oplus z^3 \oplus z^5$
6	$1 \oplus z \oplus z^6$	$1 \oplus z^5 \oplus z^6$
7	$1 \oplus z \oplus z^7$	$1 \oplus z^3 \oplus z^7$
8	$1 \oplus z \oplus z^2 \oplus z^7 \oplus z^8$	$1 \oplus z \oplus z^6 \oplus z^7 \oplus z^8$
9	$1 \oplus z^4 \oplus z^9$	$1 \oplus z^5 \oplus z^9$
10	$1 \oplus z^3 \oplus z^{10}$	$1 \oplus z^7 \oplus z^{10}$

Properties of maximum length PRBS

let $y(t)$ be an ML PRBS of period $M = 2^n - 1$

- within one period $y(t)$ contains $(M+1)/2 = 2^{n-1}$ ones and $(M-1)/2 = 2^{n-1} - 1$ zeros
- For $k = 1, 2, \dots, M-1$,

$$y(t) \oplus y(t-k) = y(t-l)$$

  for some $l \in [1, M-1]$ that depends on k

moreover, for any binary variables x, y,

$$xy = \frac{1}{2}(x + y - (x \oplus y))$$

these properties will be used to compute the covariance function of maximum length PRBS

Covariance function of maximum length PRBS

the mean is given by counting the number of outcome 1 in $y(t)$:

$$m = \frac{1}{M}\sum_{t=1}^{M} y(t) = \frac{1}{M}\left(\frac{M+1}{2}\right) = \frac{1}{2} + \frac{1}{2M}$$

the mean is slightly greater than 0.5

using $y^2(t) = y(t)$, we have the covariance function at lag zero as

$$C(0) = \frac{1}{M}\sum_{t=1}^{M} y^2(t) - m^2 = m - m^2 = \frac{M^2 - 1}{4M^2}$$

the variance is therefore slightly less than $1/4$

Covariance function of maximum length PRBS

for $\tau = 1, 2, \ldots$,

$$\begin{aligned} C(\tau) &= (1/M)\sum_{t=1}^{M} y(t+\tau)y(t) - m^2 \\ &= \frac{1}{2M}\sum_{t=1}^{M}[y(t+\tau) + y(t) - (y(t+\tau) \oplus y(t))] - m^2 \\ &= m - \frac{1}{2M}\sum_{t=1}^{M} y(t+\tau-l) - m^2 = m/2 - m^2 \\ &= -\frac{M+1}{4M^2} \end{aligned}$$

Asymptotic behavior of the covariance function of PRBS

Define $\tilde{y}(t) = -1 + 2y(t)$ so that its outcome is either -1 or 1

if M is large enough,

$$\begin{aligned} \tilde{m} &= -1 + 2m = 1/M \approx 0 \\ \tilde{C}(0) &= 4C(0) = 1 - 1/M^2 \approx 1 \\ \tilde{C}(\tau) &= 4C(\tau) = -1/M - 1/M^2 \approx -1/M, \quad \tau = 1, 2, \ldots, M-1 \end{aligned}$$

with a large period length M

- the covariance function of PRBS has similar properties to a white noise
- however, their spectral density matrices can be drastically different

Spectral density of PRBS

the output of PRBS sequence is shifted to values $-a$ and a with period M

the autocorrelation function is also periodic and given by

$$R(\tau) = \begin{cases} a^2, & \tau = 0, \pm M, \pm 2M, \ldots \\ -\frac{a^2}{M}, & \text{otherwise} \end{cases}$$

since $R(\tau)$ is periodic with period M, it has a Fourier representation:

$$R(\tau) = \sum_{k=0}^{M-1} C_k e^{\mathrm{i}2\pi\tau k/M}, \quad \text{with Fourier coefficients } C_k$$

therefore, the spectrum of PRBS is an impulse train:

$$S(\omega) = \sum_{k=0}^{M-1} C_k \delta\left(\omega - \frac{2\pi k}{M}\right)$$

Spectral density of PRBS

hence, the Fourier coefficients

$$C_k = \frac{1}{M}\sum_{\tau=0}^{M-1} R(\tau)e^{-\mathrm{i}2\pi\tau k/M}$$

are also the spectral coefficients of $S(\omega)$

using the expression of $R(\tau)$, we have

$$C_0 = \frac{a^2}{M^2}, \quad C_k = \frac{a^2}{M^2}(M+1), \quad k = 1, 2, \ldots$$

therefore,

$$S(\omega) = \frac{a^2}{M^2}\left[\delta(\omega) + (M+1)\sum_{k=1}^{M-1}\delta(\omega - 2\pi k/M)\right]$$

It does not resemble spectral characteristic of a white noise (flat spectrum)

Comparison of the covariances between filtered inputs

- define $y_1(t)$ as the output of a filter:

$$y_1(t) - ay_1(t-1) = u_1(t),$$

  with white noise $u(t)$ of zero mean and variance λ^2

- define $y_2(t)$ be the output of the same filter:

$$y_2(t) - ay_2(t-1) = u_2(t),$$

  where $u_2(t)$ is a PRBS of period M and amplitude λ

what can we say about the covariances of $y_1(t)$ and $y_2(t)$?

Comparison of the correlations between filtered inputs

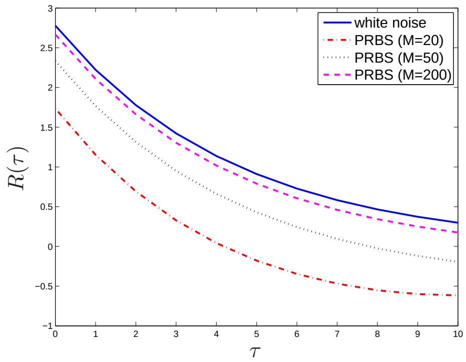
the correlation function of $y_1(t)$ is given by

$$R_1(\tau) = \left(\frac{\lambda^2}{1-a^2}\right)a^\tau, \quad \tau \geq 0$$

the correlation function of $y_2(t)$ can be calculated as

$$\begin{aligned} R_2(\tau) &= \int_{-\pi}^{\pi} S_{y_2}(\omega)e^{\mathrm{i}\omega\tau}d\omega \\ &= \int_{-\pi}^{\pi} S_{u_2}(\omega)\left|\frac{1}{1-ae^{\mathrm{i}\omega}}\right|^2 e^{\mathrm{i}\tau\omega}d\omega \\ &= \frac{\lambda^2}{M}\left[\frac{1}{(1-a)^2} + (M+1)\sum_{k=1}^{M-1}\frac{\cos(2\pi\tau k/M)}{1+a^2-2a\cos(2\pi k/M)}\right] \end{aligned}$$

Plots of the correlation functions



- the filter parameter is $a = 0.8$
- $R(\tau)$ of white noise and PRBS inputs are very close when M is large

Persistent excitation

a signal $u(t)$ is **persistently exciting** of order n if

1. the following limit exists:

$$R(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t+\tau)u(t)^T$$

2. the following matrix is positive definite

$$\mathbf{R}_n = \begin{bmatrix} R(0) & R(1) & \dots & R(n-1) \\ R(-1) & R(0) & \dots & R(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(1-n) & R(2-n) & \dots & R(0) \end{bmatrix}$$

(if $u(t)$ is from an ergodic stochastic process, then $\mathbf{R}(n)$ is the usual covariance matrix (assume zero mean))

Examining the order of persistent excitation

- **white noise input** of zero mean and variance λ^2

$$R(\tau) = \lambda^2\delta(\tau), \quad \Longrightarrow \quad \mathbf{R}_n = \lambda^2 I_n$$

  thus, white noise is persistently exciting of *all* orders

- **step input** of magnitude λ

$$R(\tau) = \lambda^2, \quad \forall\tau \quad \Longrightarrow \quad \mathbf{R}_n = \lambda^2\mathbf{1}_n$$

  a step function is persistently exciting of order 1

- **impulse input**: $u(t) = 1$ for $t = 0$ and 0 otherwise

$$R(\tau) = 0, \quad \forall\tau \quad \Longrightarrow \quad \mathbf{R}_n = 0$$

  an impulse is *not* persistently exciting of any order

Example 1: FIR models

recall the problem of estimating an FIR model where

$$h(k) = 0, \quad k \geq M$$

the coefficients $h(k)$ are the solution to the following equation

$$\begin{bmatrix} R_{yu}^T(0) \\ R_{yu}^T(1) \\ \vdots \\ R_{yu}^T(M-1) \end{bmatrix} = \begin{bmatrix} R_u(0) & R_u(1) & \cdots & R_u(M-1) \\ R_u(-1) & R_u(0) & \cdots & R_u(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_u(1-M) & R_u(2-M) & \cdots & R_u(0) \end{bmatrix} \begin{bmatrix} h^T(0) \\ h^T(1) \\ \vdots \\ h^T(M-1) \end{bmatrix}$$

- the equations has a unique solution iff $\mathbf{R}_M$ is nonsingular
- equivalent condition: u must be persistently exciting of order M
- need more p.e. if the model is more complex

Example 2: Estimating noisy linear models

consider a least-squares problem of estimating a first-order model

$$y(t) = ay(t-1) + bu(t) + e(t)$$

where $u(t)$ is an input signal, and $e(t)$ is an i.i.d. noise of zero mean

we can show that

- if $u(t)$ is a PRBS or step input, the consistent estimates are obtained, i.e.,

$$(\hat{a}, \hat{b}) \rightarrow (a, b), \quad \text{as } N \rightarrow \infty$$

- if $u(t)$ is an impulse, $\hat{a} \rightarrow a$ but $\hat{b}$ *does not* converge to b as N increases
- in loose terms, the impulse input does not provide enough information on $y(t)$ to estimate b

Properties of persistently exciting signals

assumptions:

- $u(t)$ is a multivariable ergodic process
- $S_u(\omega)$ is positive in at least n distinct frequencies within $(-\pi, \pi)$

we have the following two properties

Property 1 $u(t)$ is persistently exciting of order n

Property 2 if $H(z)$ is an asymptotically stable linear filter and $\det H(z)$ has no zero on the unit circle then the filtered signal $y(t) = H(q^{-1})u(t)$ is persistently exciting of order n

we can imply an ARMA process is persistently exciting of *any finite order*

Examining the order of PRBS

consider a PRBS of period M and magnitude $a, -a$

the matrix containing n-covariance sequenes (where $n \leq M$) is

$$\mathbf{R}_n = \begin{bmatrix} a^2 & -a^2/M & \dots & -a^2/M \\ -a^2/M & a^2 & \dots & -a^2/M \\ \vdots & \vdots & \ddots & \vdots \\ -a^2/M & -a^2/M & \dots & a^2 \end{bmatrix}$$

for any $x \in \mathbf{R}^n$,

$$\begin{aligned} x^T \mathbf{R}_n x &= x^T \left((a^2 + \frac{a^2}{M}) I - \frac{a^2}{M} \mathbf{1}\mathbf{1}^T \right) x \\ &\geq a^2 (1 + \frac{1}{M}) x^T x - \frac{a^2}{M} x^T x \mathbf{1}^T \mathbf{1} = a^2 \|x\|^2 \left(1 + \frac{(1-n)}{M} \right) \geq 0 \end{aligned}$$

a PRBS with period M is persistently exciting of order M

Examining the order of sum of sinusoids

consider the signal $u(t) = \sum_{k=1}^{m} a_k \sin(\omega_k t + \phi_k)$

where $0 \leq \omega_1 < \omega_2 < \ldots < \omega_m \leq \pi$

the spectral density of u is given by

$$S(\omega) = \sum_{k=1}^{m} \frac{C_k}{2} [\delta(\omega - \omega_k) + \delta(\omega + \omega_k)]$$

therefore $S(\omega)$ is nonzero (in the interval $(-\pi, \pi]$) in exactly n points where

$$n = \begin{cases} 2m, & 0 < \omega_1, \omega_m < \pi \\ 2m - 1, & 0 = \omega_1, \text{ or } \omega_m = \pi \\ 2m - 2, & 0 = \omega_1 \text{ and } \omega_m = \pi \end{cases}$$

it follows from Property 1 that $u(t)$ is persistently exciting of order n

Summary

- the choice of input is imposed by the type of identification method
- the input signal should be persistently exciting of a certain order to ensure that the system of a certain order can be identified
- some often used signals include PRBS and ARMA processes

References

Chapter 5 in

T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Exercises

5.1 Order of persistent excitation. Determine the order of persistent excitation of the following inputs.

(a) $u(t) = (-1)^t$, $t = 0, 1, 2, \ldots$
(b) $u(t) = 1 + (-1)^t$, $t = 0, 1, 2, \ldots$

Consider the identification of a moving-average process

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \cdots + b_n u(t-n) + \nu(t)$$

using correlation analysis. The parameters $b_1, b_2, \ldots, b_n$ are to be determined and $\nu(t)$ is zero-mean noise. Determine for what order n the parameters $b_1, \ldots, b_n$ can be uniquely estimated when using the following signals.

(a) $u(t) = (-1)^t$, $t = 0, 1, 2, \ldots$
(b) $u(t) = 1 + (-1)^t$, $t = 0, 1, 2, \ldots$
(c) $u(t) = \sin(\omega_1 t) + 3\sin(\omega_2 t)$, $0 < \omega_1 < \omega_2 < \pi$,
(d) $u(t)$ is a PRBS sequence of order 3,
(e) $u(t)$ is white noise.

5.2 Pseudo random binary sequence.

(a) Write a MATLAB function to generate maximum length PRBS sequences of order $n = 3$ to $n = 10$ by using the feedback coefficients given in the following table.

n	$A(z)$
3	$1 \oplus z \oplus z^3$
4	$1 \oplus z \oplus z^4$
5	$1 \oplus z^2 \oplus z^5$
6	$1 \oplus z \oplus z^6$
7	$1 \oplus z \oplus z^7$
8	$1 \oplus z \oplus z^2 \oplus z^7 \oplus z^8$
9	$1 \oplus z^4 \oplus z^9$
10	$1 \oplus z^3 \oplus z^{10}$

The inputs of the function are the number of state variables (n), an initial state ($x(0)$), and the length of PRBS sequence (N). Save the `m`-file as `prbs_yourname.m`.

(b) Provide an example of state vectors $x(k)$ for $k = 0, 1, \ldots, M$ to show that your code gives a maximum length PRBS.

(c) Generate a 256-point PRBS sequence of magnitude 1 and -1 using $n = 3$. Use `fft` or `periodogram` command to plot the empirical spectrum of PSBS signal. Compare the plot with the closed-form expression of the spectrum. For example, locate where the peaks occur.

Chapter 6

Linear least-squares

Linear least-squares method or linear regression is one of fundamental methods in statistics and engineering. The regression formulation is based on the assumption that a model is linear in parameters that are subject to be determined. The method arises from a background in solving a system of linear equations when the number of equations is greater than the number of unknown variables, often referred to as *over-determined linear equations*. When such case occurs, one typically is not able to solve the questions exactly, so we resort to solve the equations in the least-squares sense. That is, we allow to have residual errors from each of the equations, but aim to minimize the sum square of those error, explained as the 2-norm of residual vector, instead. The readers will find that the ingredients in this chapter require a background on linear algebra given in Chapter 3.

Learning objectives of this chapter are

- to understand a linear least-squares formulation and be able to formulate a regression model from applications,
- to explain the optimality condition of the solution and derive the closed-form solution of linear least-squares problem,
- to understand statistical properties of a least-squares estimate when data are generated and corrupted by noise in a particular setting,
- to understand how to find a numerical least-squares solution, even though we know that this is a mature technology, *i.e.*, a solution can be computed in a single command in any programming language.

6. Linear least-squares

- linear regression
- examples in engineering
- solving linear least-squares
- analysis of least-squares estimate
- computational aspects

Linear regression

- linear regression is the simplest type of *parametric* model
- it explains a relationship between variables y and x using a linear function:

$$y = Ax$$

  where $y \in \mathbf{R}^m$, $A \in \mathbf{R}^{m \times n}$, $x \in \mathbf{R}^n$
- y contains the measurement variables and is called the *regressed variable* or *regressand*
- each row vector a_k^T in matrix A is called *regressor*
- the matrix A is sometimes called *the design matrix*
- x is the *parameter vector*. Its element x_k is often called *regression coefficients*

Example 1: a polynomial trend

assume the model is the form of a polynomial of degree n

$$y(t) = a_0 + a_1 t + \cdots + a_r t^n$$

with unknown coefficients $a_0, \ldots, a_n$
this can be written in the form of linear regression as

$$\begin{bmatrix} y(t_1) \\ y(t_2) \\ \vdots \\ y(t_m) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \ldots & t_1^n \\ 1 & t_2 & \ldots & t_2^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_N & \ldots & t_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

given the measurements $y(t_i)$ for $t_1, t_2, \ldots, t_m$, we want to estimate the coefficents a_k

Example 2: truncated weighting function

a truncated weighting function model (or FIR model) is given by

$$y(t) = \sum_{k=0}^{M-1} h(k)u(t-k)$$

- an input u is known and applied to the system to measure the output y
- the relationship between y and u can be fit into a linear regression as

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(k) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} u(0) & u(-1) & \dots & u(-M+1) \\ u(1) & u(0) & \dots & u(-M+2) \\ \vdots & \vdots & \vdots & \vdots \\ u(k) & u(k-1) & \dots & u(k-M+1) \\ \vdots & \vdots & \vdots & \vdots \\ u(m) & u(N-1) & \dots & u(m-M+1) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{bmatrix}$$

Solving linear regressions

- the problem is to find an estimate of x from the measurements y and A
- if we choose the number of measurements, m to be equal to n, then x can be solved by
$$x = A^{-1}y,$$
provided that A is *invertible*
- in practice, in the presence of noise and disturbance, more data should be collected in order to get a better estimate
- this leads to overdetermined linear equations where an exact solution does not usually exist
- however, it can be solved by **linear least-squares** formulation

Definition of Linear least-squares

Overdetermined linear equations

$$Ax = y \quad A \text{ is } m \times n \text{ with } m > n$$

for most y cannot solve for x

Linear least-squares formulation

$$\text{minimize } \|Ax - y\|_2 = \left(\sum_{i=1}^{m} (\sum_{j=1}^{n} a_{ij}x_j - y_i)^2 \right)^{1/2}$$

- $r = Ax = y$ is called *the residual error*
- x with smallest residual norm $\|r\|$ is called *the least-squares solution*
- equivalent to minimizing $\|Ax - y\|^2$

Example: Data fitting

fit a function

$$y = g(t) = x_1 g_1(t) + x_2 g_2(t) + \ldots + x_n g_n(t)$$

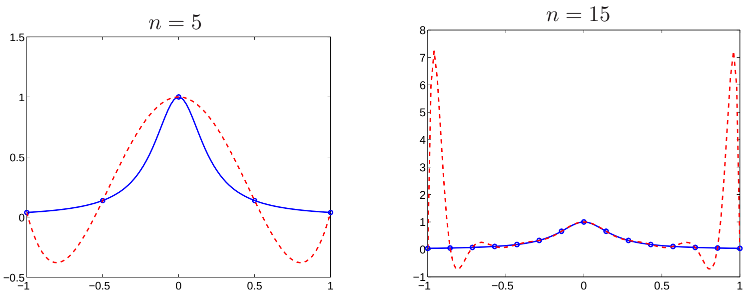
to data (t_1, y_1), (t_2, y_2), ..., (t_m, y_m), i.e., choose the coefficients x_k so that

$$g(t_1) \approx y_1, \quad g(t_2) \approx y_2, \quad , g(t_m) \approx y_m$$

- $g_i(t) : \mathbf{R} \rightarrow \mathbf{R}$ are given functions (*basis functions*)
- problem variables: the coefficients $x_1, x_2, \ldots, x_n$
- usually $m \gg n$, hence no exact solution with $g(t_i) = y_i$ for all i
- applications: developing simple, approximate model of observed data

Example: fit a polynomial to $f(t) = 1/(1+25t^2)$ on $[-1, 1]$

- pick $m = n$ points t_i in $[-1, 1]$ and calculate $y_i = 1/(1+25t_i^2)$
- interpolate by solving $Ax = y$

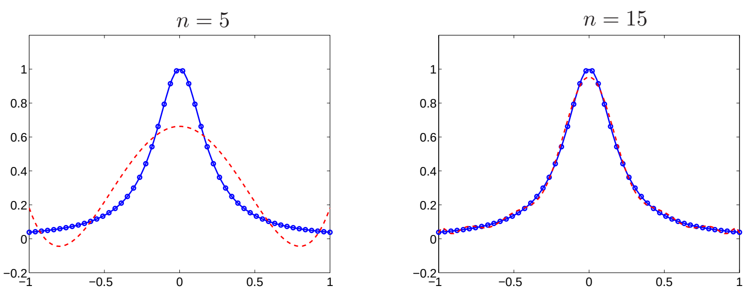


(blue solid line: f; red dashed line: polynomial g)

increase n does not improve the overall quality of the fit

same example by approximation

- pick $m = 50$ points t_i in $[-1, 1]$
- fit polynomial by minimizing $\|Ax - y\|$



(blue solid line: f; red dashed line: polynomial g)

much better fit overall

Some terminology

from the model $y = Ax + e$

variables y and A are commonly known as

y	A
endogenous variable	exogenous variable
dependent variable	independent variable
explained variable	explanatory variable
response variable	predictor
observable variable	regressor
	covariates
	manipulated variable

Closed-form of least-squares estimate

the zero gradient condition of LS objective is

$$\frac{d}{dx}\|Ax - y\|_2^2 = A^T(Ax - y) = 0$$

which is equivalent to the **normal equation**

$$A^T Ax = A^T y$$

if A is full rank:

- least-squares solution can be found by solving the normal equations
- n equations in n variables with a positive definite coefficient matrix
- the closed-form solution is $x = (A^T A)^{-1} A^T y$
- $(A^T A)^{-1} A^T$ is a left inverse of A

Properties of full rank matrices

suppose A is an $m \times n$ matrix; we always have

$$\mathbf{rank}(A) \leq \min(m, n)$$

if A is **full rank with** $m \geq n$

- $\mathbf{rank}(A) = n$ and $\mathcal{N}(A) = \{0\}$ $(Ax = 0 \Leftrightarrow x = 0)$
- $A^T A$ is positive definite: for any $x \neq 0$ then

$$\langle A^T Ax, x\rangle = \langle Ax, Ax\rangle = \|Ax\|^2 > 0$$

similarly, if A is **full rank** with $m \leq n$

- $\mathbf{rank}(A) = m$ and $\mathcal{N}(A^T) = \{0\}$
- AA^T is positive definite

Geometric interpretation of a LS problem

$$\text{minimize} \quad \|Ax - y\|^2$$

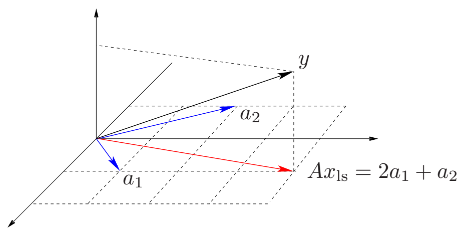
A is $m \times n$ with columns $a_1, a_2, \ldots a_m$

- $\|Ax - y\|$ is the distance of y to the vector

$$Ax = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

- solution x_{ls} gives the linear combination of the columns of A closest to y
- Ax_{ls} is the **projection** of y to the range of A

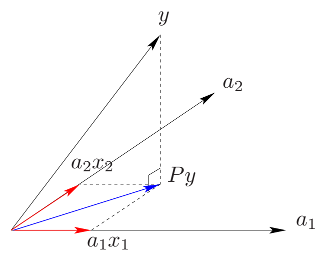
Example: $A = \begin{bmatrix} 1 & -1 \\ 1 & 2 \\ 0 & 0 \end{bmatrix}$, $y = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}$



least-squares solution x_{ls}

$$Ax_{\text{ls}} = \begin{bmatrix} 1 \\ 4 \\ 0 \end{bmatrix}, \qquad x_{\text{ls}} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Orthogonal projection



- Py is the orthogonal projection of y onto $\mathcal{R}(A)$ spanned by $a_1, \ldots, a_n$
- the projection satisfies the **orthogonality condition**

$$\langle Py - y, a_k \rangle = 0, \quad \forall k$$

(the optimal residual must be orthogonal to any vector in $\mathcal{R}(A)$)

- Py gives the best approximation; for any $\hat{y} \in \mathcal{R}(A)$ and $\hat{y} \neq Py$

$$\|y - Py\| < \|y - \hat{y}\|$$

- from the orthogonality condition and Py is a linear combination of $\{a_k\}$

$$\langle y, a_k \rangle = \langle Py, a_k \rangle = \langle \sum_{j=1}^{n} a_j x_j, a_k \rangle \quad \forall k$$

$$\begin{bmatrix} \langle y, a_1 \rangle \\ \langle y, a_2 \rangle \\ \vdots \\ \langle y, a_n \rangle \end{bmatrix} = \begin{bmatrix} \langle a_1, a_1 \rangle & \langle a_2, a_1 \rangle & \dots & \langle a_n, a_1 \rangle \\ \langle a_1, a_2 \rangle & \langle a_2, a_2 \rangle & \dots & \langle a_n, a_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle a_1, a_n \rangle & \langle a_2, a_n \rangle & \dots & \langle a_n, a_n \rangle \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- this also leads to the **normal equations**

$$A^T A x = A^T y$$

- $Ax_{\text{ls}} = Py$ with

$$P = A(A^T A)^{-1} A^T$$

if A has **full rank**

Definition: any orthogonal projection operator satisfies

- $P = P^T$
- $P^2 = P$ (Idempotent operator)

from its definition, any orthogonal projection operator obeys

- $\|Px\| \leq \|x\|$ for any x (contraction operator)
- $I - P \succeq 0$

Least-squares estimation

suppose y is generated under the dgp (data generating process)

$$y = Ax + e$$

- x is what we want to estimate or reconstruct
- y is our measurements
- e is an unknown *noise* or *measurement error*
- ith row of A characterizes $i$the sensor or ith measurement (and A is deterministic)

Least-squares estimation: choose an estimate $\hat{x}$ that minimizes

$$\|A\hat{x} - y\|$$

i.e., minimize the deviation between what we actually observed (y), and what we would observe if $x = \hat{x}$, and there were no noise ($e = 0$)

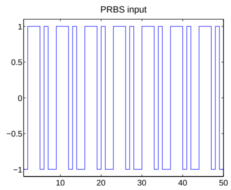
Example: first-order linear model

estimate the parameters a, b in a linear model

$$z(t) = az(t-1) + bu(t-1) + e(t)$$

from the measurement $z(t)$ and the input $u(t)$

- true parameters: $a = 0.8$, $b = 1$
- $u(t)$ is a PRBS sequence of magnitude $-1,1$ with period $M = 7$
- $e(t)$ is a zero mean white noise with variance 0.1



Estimation: choose $\hat{a}, \hat{b}$ that minimizes

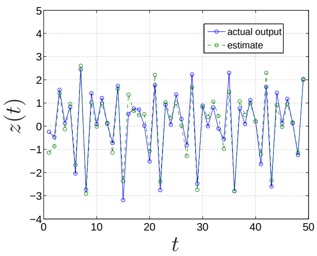
$$\sum_{t=1}^{N} \|z(t) - (\hat{a}z(t-1) + \hat{b}u(t-1))\|^2 = \|Ax - b\|^2$$

$$y = \begin{bmatrix} z(1) \\ \vdots \\ z(m) \end{bmatrix}, \quad A = \begin{bmatrix} z(0) & u(0) \\ \vdots & \vdots \\ z(m-1) & u(m-1) \end{bmatrix}, \quad x = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}$$

results:

from one realization of $e(t)$,

$$\hat{a} = 0.7485, \quad \hat{b} = 1.0768$$



Analysis of the LS estimate (static case)

assumptions:

- e is noise with zero mean and covariance matrix Σ
- the least-square estimate is given by

$$\hat{x} = \operatorname{argmin} \|Ax - y\|$$

- the information matrix A is *deterministic*

then the following properties hold:

- $\hat{x}$ is an unbiased estimate of x ($\mathbf{E}\hat{x} = x$, or $\hat{x} = x$ when $e = 0$)
- the covariance matrix of $\hat{x}$ is given by

$$\mathbf{cov}(\hat{x}) = \mathbf{E}(\hat{x} - \mathbf{E}\hat{x})(\hat{x} - \mathbf{E}\hat{x})^T = (A^TA)^{-1}A^T\Sigma A(A^TA)^{-1}$$

the expression of $\mathbf{cov}(\hat{x}) = (A^TA)^{-1}A^T\Sigma A(A^TA)^{-1}$ suggests that

- if A can be arbitrarily chosen, pick A that the covariance is small
- the covariance of the LS estimate depends on noise covariance

special case: noise covariance is diagonal

- $\Sigma = \mathbf{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ (heteroskedasticity): e_i has different variances
- $\Sigma = \sigma^2 I$ (homoskedasticity): e_i has uniform variance

for homoskedasticity case, the covariance of the LS estimate reduces to

$$\mathbf{cov}(\hat{x}) = \sigma^2(A^TA)^{-1}$$

BLUE property

under the dgp: $y = Ax + e$ and *homoskedasticity* of e, the LS estimator

$$\hat{x} = (A^TA)^{-1}A^Ty$$

is the **optimum unbiased linear least-mean-squares** estimator of x

assume $\hat{z} = By$ is any other linear estimator of x

- require $BA = I$ in order for $\hat{z}$ to be unbiased
- $\mathbf{cov}(\hat{z}) = BB^T$
- $\mathbf{cov}(\hat{x}) = BA(A^TA)^{-1}A^TB^T$ (apply $BA = I$)

Using $I - P \succeq 0$, we conclude that

$$\mathbf{cov}(\hat{z}) - \mathbf{cov}(\hat{x}) = B(I - A(A^TA)^{-1}A^T)B^T \succeq 0$$

suppose the covariance matrix of e is *not* I, says

$$\mathbf{E}ee^T = \Sigma$$

scale the equation $y = Ax + e$ by $\Sigma^{-1/2}$

$$\Sigma^{-1/2}y = \Sigma^{-1/2}Ax + \Sigma^{-1/2}e$$

the optimal unbiased linear least-mean-squares estimator of x is

$$\hat{x} = (A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}y$$

this is a special case of **weighted least-squares** problems

Weighted least-squares

given W a positive definite matrix and can be factorized as $W = L^T L$

a weighted least-squares problem is

$$\underset{x}{\text{minimize}} \quad \mathbf{tr}(Ax - y)^T W (Ax - y)$$

- equivalent formulation: $\text{minimize}_x \quad \|L(Ax - y)\|_F^2$
- can be solved from the modified normal equations

$$A^T W A x = A^T W y$$

- Ax_{wls} is the *orthogonal projection* on $\mathcal{R}(A)$ w.r.t the new inner product

$$\langle x, y \rangle_W = \langle Wx, y \rangle$$

Analysis of the LS estimate (dynamic case)

suppose we apply the LS method to a dynamical system

$$y(t) = H(t)\theta + e(t)$$

- the observations $y(1), y(2), \ldots, y(N)$ are available
- θ is the dynamical model parameter

typically, $H(t)$ contains the past outputs and inputs

$$y(1), \ldots, y(t-1), u(1), \ldots u(t-1)$$

(hence $H(t)$ is *no longer* deterministic)

and $e(t)$ is white noise with covariance Σ

the LS estimate $\hat{\theta}_N$ (depending on N) given by

$$\hat{\theta}_N = \left[\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right]^{-1} \left[\frac{1}{N}\sum_{t=1}^{N} H(t)^T y(t)\right]$$

has the following properties (under some assumptions):

- $\hat{\theta}_N$ is consistent, $i.e.$, it converges to the true parameter in probability

$$\operatorname{plim} \hat{\theta}_N = \theta \quad \Longleftrightarrow \quad \lim_{N\to\infty} P(|\hat{\theta}_N - \theta| > \epsilon) = 0$$

- $\sqrt{N}(\hat{\theta} - \theta)$ is asymptotically Gaussian distributed $\mathcal{N}(0, P)$ where

$$P = \Sigma_x^{-1} \Sigma_{ux} \Sigma_x^{-1}$$

Σ_x involves $\mathbf{E}[H(t)^T H(t)]$ and Σ_{ux} involes $\mathbf{E}[H(t)e(t)e(t)^T H(t)^T]$

the consistency results of LS estimate are based on *some assumptions*

$$\begin{aligned}\hat{\theta}_N - \theta &= \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)^{-1} \left\{\frac{1}{N}\sum_{t=1}^{N} H(t)^T y(t) - \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)\theta\right\} \\ &= \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)^{-1} \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T e(t)\right)\end{aligned}$$

hence, $\hat{\theta}_N$ is consistent if

- $\mathbf{E}[H(t)^T H(t)]$ is nonsingular
  satisfied in most cases, except u is not persistently exciting of order n
- $\mathbf{E}[H(t)^T e(t)] = 0$
  not satisfied in most cases, except $e(t)$ is white noise

Solving LS via Cholesky factorization

every positive definite $B \in \mathbf{S}^n$ can be factored as

$$B = LL^T$$

where L is lower triangular with positive diagonal elements

Fact: for $B \succ 0$, a linear equation

$$Bx = b$$

can be solved in $(1/3)n^3$ flops

solve the least-squares problem from the normal equations

$$A^T A x = A^T y$$

we have $A^T A \succ 0$ when A is full rank

Solving LS via QR factorization

- full QR factorization:
  $$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$
  with $[Q_1 \;\; Q_2] \in \mathbf{R}^{m \times m}$ orthogonal, $R_1 \in \mathbf{R}^{n \times n}$ upper triangular, invertible
- multiplication by orthogonal matrix doesn't change the norm, so

$$\begin{aligned}\|Ax - y\|^2 &= \left\| \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - y \right\|^2 \\ &= \left\| \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^T \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^T y \right\|^2\end{aligned}$$

$$= \left\| \begin{bmatrix} R_1 x - Q_1^T y \\ -Q_2^T y \end{bmatrix} \right\|^2$$
$$= \|R_1 x - Q_1^T y\|^2 + \|Q_2^T y\|^2$$

- this can be minimized by the choice $x_{\rm ls} = R_1^{-1} Q_1^T y$ (which makes the first term zero)
- residual with optimal x is

$$A x_{\rm ls} - y = -Q_2 Q_2^T y$$

- $Q_1 Q_1^T$ gives projection on $\mathcal{R}(A)$
- $Q_2 Q_2^T$ gives projection on $\mathcal{R}(A)^\perp$

Summary

- the linear least-squares method can be applied to models that are linear in the parameters
- a LS solution is unique if there is no colinearity (A is full rank)
- the method is mature, can be solve efficiently and is available in many softwares
- LS estimate has the BLUE property under the assumption that the noise in data generating process is homoskedastic
- LS estimate is consistent if the additive noise is uncorrelated with the regressors and the system is persistently excited

References

L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Second edition, 1999

Chapter 4 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 2-3 in
T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000

W.H. Greene, *Econometric Analysis*, Prentice Hall, 2008

Linear least-squares and *The solution of a least-squares problem*, EE103, Lieven Vandenberghe, UCLA, `http://www.ee.ucla.edu/~vandenbe/ee103.html`

Lectures on
Least-squares and *Least-squares applications*, EE263, Stephen Boyd, Stanford, `http://www.stanford.edu/class/ee263/letures.html`

Exercises

6.1 Least-squares fitting to a linear model. Consider a linear model with additive noise

$$y(t) = a + bt + e(t)$$

where a, b are constant and $e(t)$ is white noise with zero mean and unit variance. Suppose our goal is to estimate b only. Of course, one approach is to form a linear least-squares problem to estimate both a and b. This means we use the model

$$\mathcal{M}_1 : \quad y(t) = a + bt + \epsilon_1(t);$$

where $\epsilon_1(t)$ is the residual error, for the estimation problem. Alternatively, we can also work with the *difference* data. If we define $z(t) = y(t) - y(t-1)$, we can use the model:

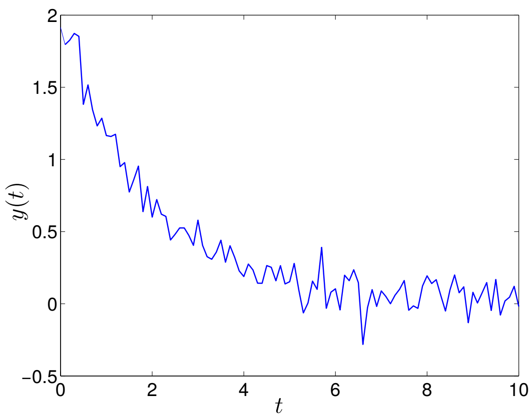
$$\mathcal{M}_2 : \quad z(t) = b + \epsilon_2(t);$$

for estimating b as well.

For each model, formulate the problem of estimating b into a linear least-squares problem. Check whether the estimate is unbiased, *i.e.*, $\mathbf{E}\hat{b} = b$. Derive and compare the variances of the estimate in the two cases. Assume that the data are collected at times $t = 1, 2, \ldots, N$.

Useful formula: $\sum_{i=1}^{n} i = n(n+1)/2$, $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.

6.2 Estimation of time series data. Consider a time series shown in the following figure.



One can see the time series has a decaying trend, so it would be interesting to consider an exponential model:

$$y(t) = e^{-at} y(0),$$

where $y(0)$ and a (time constant) are parameters to be determined.

(a) Formulate an estimation problem for this model and validate the result with the data given in `data-time-series.mat`. The file contains two realizations of this data set. You can use variable `y` for estimation and use variable `z` for validation.

(b) Propose other types of models (at least another two) that could fit to this data set. State clearly a main difference between the proposed models and the exponential model. Give a formulation for your estimation problem and compute the numerical values of the estimates from all the models you consider. Compare the estimation results with the exponential model (on the validation data set). A criterion you use to make a comparison must be quantitative and justifiable.

6.3 Multi 2-norm objectives. Consider the problem of minimizing the sum of two objectives.

$$\text{minimize} \quad \|Ax - b\|_2^2 + \rho\|Cx - d\|_2^2$$

with variable $x \in \mathbf{R}^n$ and $A \in \mathbf{R}^{m\times n}, b \in \mathbf{R}^m, C \in \mathbf{R}^{p\times n}$ and $d \in \mathbf{R}^p$ are given matrices. The parameter ρ is a given positive scalar. Show that this problem can be (easily) formulated in to a single 2-norm objective:

$$\text{minimize} \quad \|\mathcal{A}x - \mathbf{b}\|_2^2.$$

Derive what $\mathcal{A}$ and $\mathbf{b}$ are.

6.4 Estimation of scalar AR processes. An autoregressive (AR) processes of order p is described by

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_p y(t-p) + c + \nu(t), \tag{6.1}$$

where $\nu(t)$ is white noise. It represents a pure time series model where no input signal is assumed to be present. The parameters $a_1, a_2, \ldots, a_p$ are AR coefficients and c is a constant that describes a drift term in the model. Suppose a set of measurements $y(1), y(2), \ldots, y(N)$ is available and we wish to fit an AR model to these data. Formulate a least-squares problem to estimate $a_1, a_2, \ldots, a_p$ and c.

(a) A general least-squares formulation is to minimize $\|Ax - b\|$. Explain what A and b are, in this problem.

(b) Fit an AR model of order 3 to the Nikkei stock prices collected daily during Feb 2011 - Feb 2012. Use `nikkei_feb11_feb12` to find $y(1), y(2), \ldots, y(N)$. Plot a graph to compare the real data $y(t)$ and the estimate $\hat{y}(t)$ computed from the estimated model. Attach your MATLAB codes in the work sheet.

(c) Give the estimate values of a_1, a_2, a_3 and c. Discuss the results you found. How does the stock price from the past dates influence the current price ?

6.5 Navigation from range measurements. Let $(x, y) \in \mathbf{R}^2$ be the unknown coordinate of a point in the plane that we would like to track. Let $(p_i, q_i) \in \mathbf{R}^2$ be the known coordinates of a beacon for $i = 1, 2, \ldots, n$. Each of these beacons measures the distance between (x, y) and the ith beacon which is given by

$$d(x, y) = \|(p, q) - (x, y)\|_2. \tag{6.2}$$

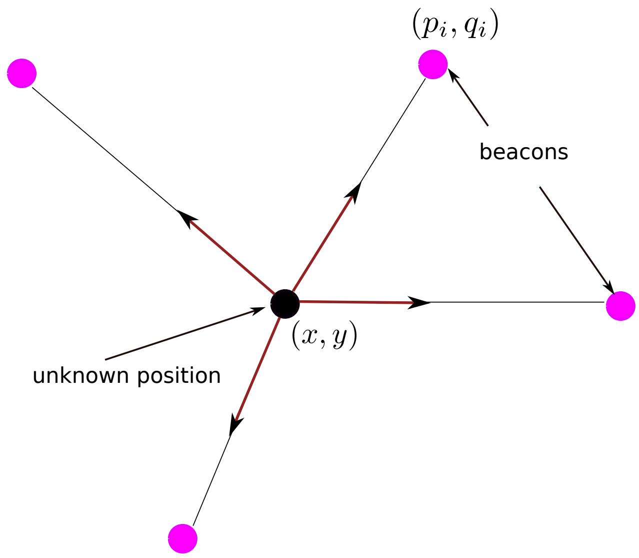
Our goal is to make use of the distance measurements from the n beacons to estimate the unknown position using the least-squares method. Let $(x_0, y_0) \in \mathbf{R}^2$ be a point assumed to be *known* and *close* to (x, y). Therefore, the distance in (6.2) can be approximated using the first-order Taylor expansion about (x_0, y_0) as

$$d(x, y) = d(x_0 + \delta y, x_0 + \delta y) \approx d(x_0, y_0) + Dd(x_0, y_0) \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$

where $Dd(x, y)$ is the first derivative of the distance function and $(\delta x, \delta y) = (x, y) - (x_0, y_0)$. If we use the approximate model and formulate as

$$\delta d \triangleq d(x, y) - d(x_0, y_0) = A \begin{bmatrix} \delta x \\ \delta y \end{bmatrix}$$

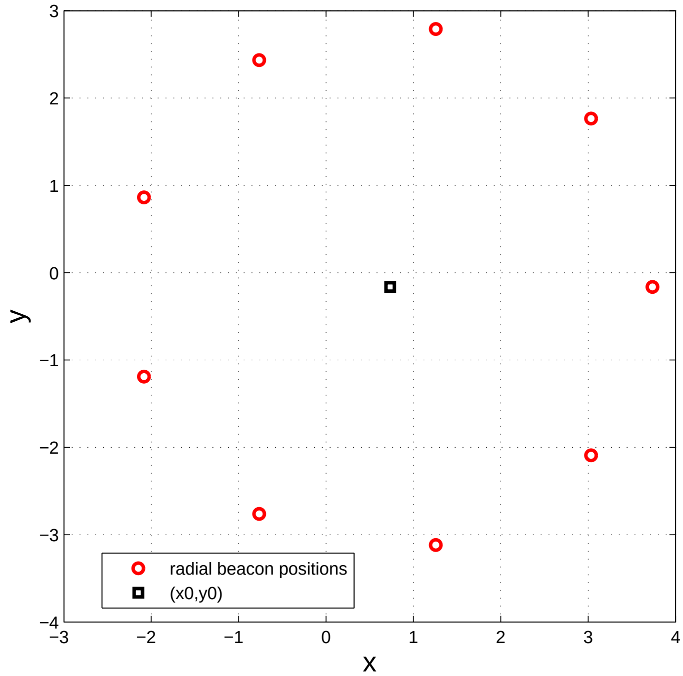
then the problem we're looking at is to choose $(\delta x, \delta y)$ so that δd is minimized.



(a) Suppose we have n measurements of distances from n beacons: $d_i(x, y) = \|(p_i, q_i) - (x, y)\|_2$ for $i = 1, \ldots, n$. Show that the estimation of (x, y) can be cast as a linear least-squares problem: minimize $\|b - Au\|_2$ with the variable $u = (\delta x, \delta y)$. Write down what b and A are.

(b) Find the condition for the uniqueness of the least-squares estimate. Describe the conditions geometrically (*i.e.*, does it depend on the number or locations of beacons?).

(c) Suppose we have n beacons and we place them in a symmetric layout around (x_0, y_0) as shown in the figure.



These positions can be computed in `beacon_pos.m` where the input is the number of beacons (n), the known coordinate (x_0, y_0) and the radius from (x_0, y_0) to each beacon, R. Choose a range of n (up to 10) and check the condition you derived in part b). If $z_0 = (x_0, y_0) = (0.4062, 0.2728)$ and set $R = 3$, compute the estimate of (x, y) versus n and write down in the table. Write down your codes and save it as `navigation.m`

	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$
$\hat{x}$								
$\hat{y}$								

Chapter 7

Significance tests for linear regression

When considering a linear least-squares problem, one usually makes an assumption that the measurement is generated from the so called *data generating process*, $y = X\beta + e$, where e is assumed to be noise having a certain distribution. Solving a linear least-squares problem does not require any statistical assumption about e, but those properties of e allow us to conclude about properties of the least-squares estimator, that is yet, for sure also a random entity. In Chapter 6, we have seen that the least-squares estimator is unbiased and its covariance matrix depends on the regressor matrix, X. This information should be used to remind us that whenever an estimator is calculated from a data set, it is never equal to the true value of the parameter (unless the data is generated from a noise-free model), but we should use statistical properties to explain about confident interval of such calculated value. One typical question in regression problem is to explore which explanatory variable is significant through the value of the corresponding regression coefficient. This leads to the test whether β_i is close to zero or not, so that we can infer about the significance of the ith variable to the model. In applications, if a variable is not significant, then it can be removed to obtain a more parsimonious model.

Learning objectives of this chapter are

- to review the fundamental concept of hypothesis test in statistics,
- to understand a significance test of regression coefficients and be able to perform such test on real data.

7. Significance tests for linear regression

- reviews on hypothesis testing
- regression coefficient test

Hypothesis tests

elements of statistical tests

- null hypothesis, alternative hypothesis
- test statistics
- rejection region
- type of errors: type I and type II errors
- confidence intervals, p-values

examples of hypothesis tests:

- hypothesis tests for the mean, and for comparing the means
- hypothesis tests for the variance, and for comparing variances

Testing procedures

a test consists of

- providing a statement of the hypotheses (H_0 (null) and H_1 (alternative))
- giving a rule that dictates if H_0 should be rejected or not

the decision rule involves a test statistic calculated on observed data

the Neyman-Pearson methodology partitions the sample space into two regions

the set of values of the test statistic for which:

the null hypothesis is rejected	**rejection region**
we fail to reject the null hypothesis	**acceptance region**

Test errors

since a test statistic is random, the same test can lead to different conclusions

- **type I error**: the test leads to *reject* H_0 when it is *true*
- **type II error:** the test *fails* to reject H_0 when it is *false*; sometimes called false alarm

probabilities of the errors:

- let β be the probability of type II error
- the **size** of a test is the probability of a type I error and denoted by α
- the **power** of a test is the probability of rejecting a false H_0 or $(1-\beta)$

α is known as **significance level** and typically controlled by an analyst

for a given α, we would like β to be as small as possible

Some common tests

- normal test
- t-test
- F-test
- Chi-square test

e.g. a test is called a t-test if the test statistic follows t-distribution

two approaches of hypothesis test

- critical value approach
- p-value approach

Critical value approach

Definition: the critical value (associated with a significance level α) is the value of the known distribution of the test statistic such that the probability of type I error is α

steps involved this test

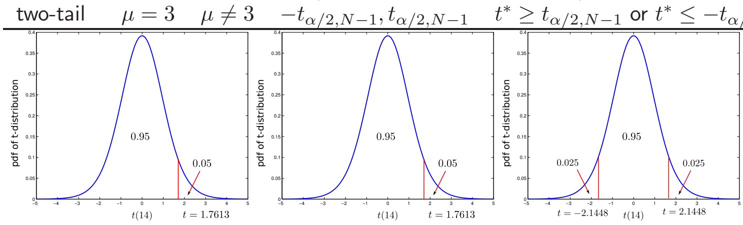
1. define the null and alternative hypotheses.
2. assume the null hypothesis is true and calculate the value of the test statistic
3. set a small significance level (typically $\alpha = 0.01, 0.05$, or 0.10) and determine the corresponding critical value
4. compare the test statistic to the critical value

condition	decision
the test statistic is more extreme than the critical value	reject H_0
the test statistic is less extreme than the critical value	accept H_0

example: hypothesis test on the population mean

- samples $N = 15$, $\alpha = 0.05$
- the test statistic is $t^* = \frac{\bar{x}-\mu}{s/\sqrt{N}}$ and has t-distribution with $N-1$ df

test	H_0	H_1	critical value	reject H_0 if
right-tail	$\mu = 3$	$\mu > 3$	$t_{\alpha,N-1}$	$t^* \geq t_{\alpha,N-1}$
left-tail	$\mu = 3$	$\mu < 3$	$-t_{\alpha,N-1}$	$t^* \leq -t_{\alpha,N-1}$
two-tail	$\mu = 3$	$\mu \neq 3$	$-t_{\alpha/2,N-1}, t_{\alpha/2,N-1}$	$t^* \geq t_{\alpha/2,N-1}$ or $t^* \leq -t_{\alpha/2,N-1}$



p-value approach

Definition: the p-value is the probability of observing a more extreme test statistic in the direction of H_1 than the one observed, by assuming that H_0 were true

steps involved this test

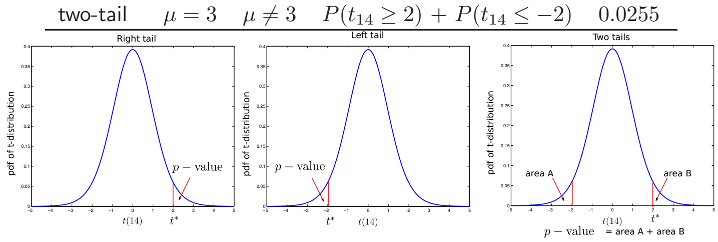
1. define the null and alternative hypotheses.
2. assume the null hypothesis is true and calculate the value of the test statistic
3. calculate the p-value using the known distribution of the test statistic
4. set a significance level α (small value such as $0.01, 0.05$)
5. compare the p-value to α

condition	decision
p-value $\leq \alpha$	reject H_0
p-value $\geq \alpha$	accept H_0

example: hypothesis test on the population mean (same as on page 7-7)

- samples $N = 15$, $\alpha = 0.01$ (have only a 1% chance of making a Type I error)
- suppose the test statistic (calculated from data) is $t^* = 2$

test	H_0	H_1	p-value expression	p-value
right-tail	$\mu = 3$	$\mu > 3$	$P(t_{14} \geq 2)$	0.0127
left-tail	$\mu = 3$	$\mu < 3$	$P(t_{14} \leq -2)$	0.0127
two-tail	$\mu = 3$	$\mu \neq 3$	$P(t_{14} \geq 2) + P(t_{14} \leq -2)$	0.0255



right-tail/left-tail tests: reject H_0, two-tail test: accept H_0

the two approaches assume H_0 were true and determine

p-value	critical value
the probability of observing a more extreme test statistic in the direction of the alternative hypothesis than the one observed	whether or not the observed test statistic is more extreme than would be expected (called critical value)

the null hypothesis is rejected if

p-value	critical value
$p - \text{value} \leq \alpha$	test statistic $\geq$ critical value

Significance tests for linear regression

- reviews on hypothesis testing
- **regression coefficient test**

Recap of linear regression

a linear regression model is

$$y = X\beta + u, \quad X \in \mathbf{R}^{N \times n}$$

homoskedasticity assumption: u_i has the same variance for all i, given by σ^2

- prediction (fitted) error: $\hat{u} := \hat{y} - y = X\hat{\beta} - y$
- residual sum of squares: $\mathrm{RSS} = \|\hat{u}\|_2^2$
- a consistent estimate of σ^2: $s^2 = \mathrm{RSS}/(N-n)$
- $(N-n)s^2 \sim \chi^2(N-n)$
- square root of s^2 is called **standard error of the regression**
- $\mathbf{Avar}(\hat{\beta}) = s^2(X^TX)^{-1}$ (estimated asymptotic covariance)

Common tests for linear regression

- testing a hypothesis about a coefficient

$$H_0 : \beta_k = 0 \quad \text{VS} \quad H_1 : \beta_k \neq 0$$

  we can use both t and F statistics

- testing using the fit of the regression

$$H_0 : \text{reduced model} \quad \text{VS} \quad H_1 : \text{full model}$$

  if H_0 were true, the reduced model ($\beta_k = 0$) would lead to smaller prediction error than that of the full model ($\beta_k \neq 0$)

Testing a hypothesis about a coefficient

statistics for testing hypotheses:

$$H_0 : \beta_k = 0 \quad \text{VS} \quad H_1 : \beta_k \neq 0$$

- $\frac{\hat{\beta}_k}{\sqrt{s^2((X^TX)^{-1})_{kk}}} \sim t_{N-n}$
- $\frac{(\hat{\beta}_k)^2}{\sqrt{s^2((X^TX)^{-1})_{kk}}} \sim F_{1,N-n}$

the above statistics are Wald statistics (see derivations in Greene book)

- the term $\sqrt{s^2((X^TX)^{-1})_{kk}}$ is referred to **standard error of the coefficient**
- the expression of SE can be simplified or derived in many ways (please check)
- e.g. R use t-statistic (two-tail test)

Testing using the fit of the regression

hypotheses are based on the fitting quality of reduced/full models

$$H_0 : \text{reduced model} \quad \text{VS} \quad H_1 : \text{full model}$$

reduced model: $\beta_k = 0$ and full model: $\beta_k \neq 0$

the F-statistic used in this test

$$\frac{(\text{RSS}_R - \text{RSS}_F)}{\text{RSS}_F/(N-n)} \sim F(1, N-n)$$

- RSS_R and RSS_F are the residual sum squares of reduced and full models
- RSS_R cannot be smaller than RSS_F, so if H_0 were true, then the F statistic would be zero
- e.g. `fitlm` in MATLAB use this F statistic, or in ANOVA table

MATLAB example

perform t-test using $\alpha = 0.05$ and the true parameter is $\beta = (1, 0, -1, 0.5)$

realization 1: $N = 100$

```
>> [btrue b SE pvalue2side]  =
    1.0000    1.0172    0.1087    0.0000
         0    0.1675    0.0906    0.0675
   -1.0000   -1.0701    0.1046    0.0000
    0.5000    0.5328    0.1007    0.0000
```

- $\hat{\beta}$ is close to β
- it's not clear if $\hat{\beta}_2$ is zero but the test decides $\hat{\beta}_2 = 0$
- note that all coefficients have pretty much the same SE

realization 2: $N = 10$

```
>> [btrue b SE pvalue2side]  =
    1.0000    1.0077    0.2894    0.0131
         0    0.1282    0.4342    0.7778
   -1.0000   -1.5866    0.2989    0.0018
    0.5000    0.2145    0.2402    0.4062
```

realization 3: $N = 10$

```
>> [btrue b SE pvalue2side]  =
    1.0000    0.8008    0.3743    0.0762
         0   -0.5641    0.5442    0.3399
   -1.0000   -1.1915    0.5117    0.0588
    0.5000    0.6932    0.4985    0.2137
```

- some of $\hat{\beta}$ is close to the true value but some is not
- the test 2 decides $\hat{\beta}_2$ and $\hat{\beta}_4$ are zero while the test 3 decides all β are zero
- the sample size N affects type II error (fails to reject H_0) and we get different results from different data sets

Summary

- common tests are available in many statistical softwares, e.g, `minitab`, `lm` in R, `fitlm` in MATLAB,
- one should use with care and interpret results correctly
- an estimator is random; one cannot trust its value calculated based on a data set
- examining statistical properties of an estimator is preferred

References

W.H. Greene, *Econometric Analysis*, Prentice Hall, 2008

Review of Basic Statistics (online course)

https://onlinecourses.science.psu.edu/statprogram

Stat 501 (online course)

https://onlinecourses.science.psu.edu/stat501

Chapter 8

Variants of least-squares

A linear least-squares problem is regarded as an unconstrained optimization with a quadratic cost objective. Most problems in engineering have some physical constraints in the parameters (or design variables) to be estimated. For example, price, length or width should be nonnegative quanties. Adding this prior condition to the linear least-squares problem results in a constrained quadratic optimization problem and the solution may or may not be obtained in a closed-form, as opposed to the unconstrained least-squares solution. Another important extension of the problem to put a penalty on the parameters in various forms, known as a *regularization method* in statistics. This prior comes from our assumption on the background of the application. For instance, in many applications, one prefers to obtain a parsimonious model, which means there are only a few number of nonzero parameters in the model. In such case, we tend to promote most parameters to be zero using a ℓ_1 penalty function added to the cost function of the least-squares problem. If we have an assumption that the model parameters should not be large (in the sum-square average, or a 2-norm sense) then we can add a ℓ_2 penalty function instead. These two instances can also be extended to an estimation of *groups* of model where we are interested in a common feature or the differences among those models. Lastly, we may encounter a least-squares problem when some of the problem parameters (regressor matrix, or output) are uncertain but we have some information about this uncertainty, either in deterministic or stochastic sense. This requires a reformulation in a robust sense, called *robust least-squares*, meaning that we aim to guarantee that the solution is optimal even if we have uncertainty in the problem parameters.

Learning objectives of this chapter are

- to formulate physical conditions of parameters as mathematical constraints in the least-squares problem and solve for numerical solutions using existing optimization methods,
- to understand the regularization methods and explain their connections with statistical estimation methods,
- to be able to numerically solve two basic methods: ℓ_1 and ℓ_2-regularized least-squares problems and understand the solution behaviors from those methods,
- to understand how to reformulate a robust least-squares problem when the information about model uncertainty is given.

8. Variations on least-squares

- least-squares with constraints
- ℓ_2 regularization
- ℓ_1 regularization
- generalizations of ℓ_1-regularized LS
- robust least-squares

Least-squares with constraints

$$\begin{array}{ll} \text{minimize} & \|Ax - y\| \\ \text{subject to} & x \in \mathcal{C} \end{array}$$

$\mathcal{C}$ is a convex set (many applications fall into this case)

- used to rule out certain unacceptable approximations of y
- arise as prior knowledge of the vector x to be estimated
- same as determining the projection of y on a set more complicated than a subspace
- form a convex optimization problem with no analytical solution (typically)

nonnegativity constraints on variables

$$\mathcal{C} = \{ x \mid x \succeq 0 \}$$

- parameter x known to be nonnegative, e.g., powers, rates, etc.
- finding the projection of y onto the *cone* generated by the columns of A

variable bounds

$$\mathcal{C} = \{ x \mid l \preceq x \preceq u \}$$

- vector x known to lie in an interval $[l, u]$
- finding the projection of y onto the image of a box under the linear mapping induced by A

probability distribution

$$\mathcal{C} = \{ \, x \mid x \succeq 0, \quad \mathbf{1}^T x = 1 \, \}$$

- arise in estimation of proportions which are nonnegative and sum to one
- approximating y by a convex combination of the columns of A

norm ball constraint

$$\mathcal{C} = \{ \, x \mid \|x - x_0\| \leq d \, \}$$

where x_0 and d are problem parameters

- x_0 is a prior guess of what x should be
- d is the maximum plausible deviation from our prior guess
- the constraints $\|x - x_0\| \leq d$ can denote a **trust region**. (the linear relation $y = Ax$ is an approximation and only valid when x is near x_0)

ℓ_2-regularized least-squares

adding the 2-norm penalty to the objective function

$$\underset{x}{\text{minimize}} \quad \|Ax - y\|_2^2 + \gamma \|x\|_2^2$$

- seek for an approximate solution of $Ax \approx y$ with small norm
- also called **Tikhonov regularized least-squares** or **ridge regression**
- $\gamma > 0$ controls the trade off between the fitting error and the size of x
- has the analytical solution for any $\gamma > 0$:

$$x = (A^T A + \gamma I)^{-1} A^T y$$

  (no restrictions on shape, rank of A)
- interpreted as a MAP estimation with the log-prior of the Gaussian

ℓ_1-regularized least-squares

Idea: adding $|x|$ to a minimization problem introduces a sparse solution

consider a scalar problem:

$$\underset{x}{\text{minimize}} \quad f(x) = (1/2)(x - a)^2 + \gamma |x|$$

to derive the optimal solution, we consider the two cases:

- if $x \geq 0$ then $f(x) = (1/2)(x - (a - \gamma))^2$

$$x^\star = a - \gamma, \quad \text{provided that } a \geq \gamma$$

- if $x \leq 0$ then $f(x) = (1/2)(x - (a + \gamma))^2$

$$x^\star = a + \gamma, \quad \text{provided that } a \leq -\gamma$$

when $|a| \leq \gamma$ then $x^\star$ must be zero the optimal solution to minimization of $f(x) = (1/2)(x-a)^2 + \gamma|x|$ is

$$x^\star = \begin{cases} (|a|-\gamma)\mathbf{sign}(a), & |a| > \gamma \\ 0, & |a| \le \gamma \end{cases}$$

meaning: if γ is large enough, x^* will be zero

generalization to vector case: $x \in \mathbf{R}^n$

$$\underset{x}{\text{minimize}} \quad f(x) = (1/2)\|x-a\|^2 + \gamma\|x\|_1$$

the optimal solution has the same form

$$x^\star = \begin{cases} (|a|-\gamma)\mathbf{sign}(a), & |a| > \gamma \\ 0, & |a| \le \gamma \end{cases}$$

where all operations are done in *elementwise*

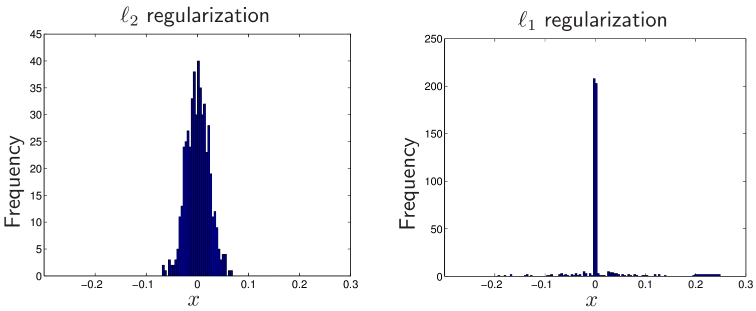
ℓ_1-regularized least-squares

adding the ℓ_1-norm penalty to the least-square problem

$$\underset{x}{\text{minimize}} \quad (1/2)\|Ax-y\|_2^2 + \gamma\|x\|_1 \tag{1}$$

- a convex heuristic method for finding a sparse x that gives $Ax \approx y$
- also called **Lasso** or **basis pursuit**
- a nondifferentiable problem due to $\|\cdot\|_1$ term
- no analytical solution, but can be solved efficiently
- interpreted as a MAP estimation with the log-prior of the Laplacian distribution

example $A \in \mathbf{R}^{m\times n}, b \in \mathbf{R}^m$ with $m = 100, n = 500, \gamma = 0.2$



- solution of ℓ_2 regularization is more widely spread
- solution of ℓ_1 regularization is concentrated at zero

Similar form of ℓ_1-regularized LS

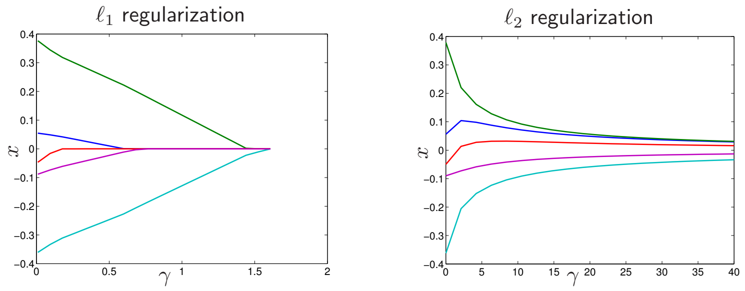
the ℓ_1-norm is an inequality constraint:

$$\underset{x}{\text{minimize}} \quad \|Ax - y\|_2 \quad \text{subject to} \quad \|x\|_1 \le t \tag{1}$$

- t is specified by the user
- t serves as a budget of the sum of absolute values of x
- the ℓ_1-regularized LS (1) is the Lagrangian form of this problem
- for each t where $\|x\|_1 \le t$ is active, there is a corresponding value of γ that yields the same solution from (1)

Solution paths of regularized LS

solve the regularized LS when $n = 5$ and vary γ (penalty parameter)



- for lasso, many entries of x are exactly zero as γ varies
- for ridge, many entries of x are nonzero but converging to small values

Generalizations of ℓ_1-regularized LS

many variants are proposed for acheiving particular structures in solutions

- elastic net: for highly correlated variables and lasso doesn't perform well
- group lasso: for acheiving sparsity in group
- fused lasso: for neighboring variables to be similar

Elastic net

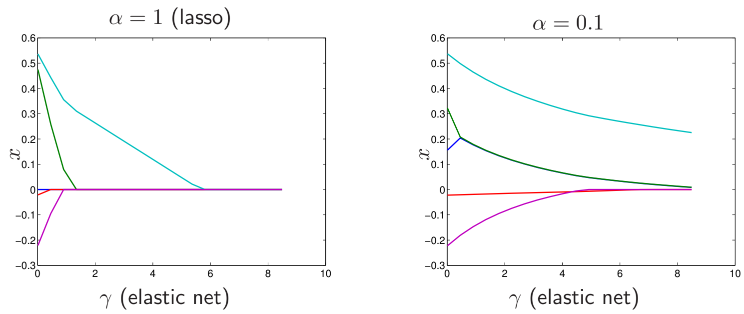
a combination between the ℓ_1 and ℓ_2 regularizations

$$\underset{x}{\text{minimize}} \quad (1/2)\|Ax - y\|_2^2 + \gamma \left\{(1/2)(1-\alpha)\|x\|_2^2 + \alpha\|x\|_1\right\}$$

where $\alpha \in [0,1]$ and γ are parameters

- when $\alpha = 1$ it's lasso and when $\alpha = 0$ it's a ridge regression
- used when we expect groups of very correlated variables (e.g. microarray, genes)
- strictly convex problem for any $\alpha < 1$ and $\lambda > 0$ (unique solution)

generate $A \in \mathbf{R}^{20\times 5}$ where a_1 and a_2 are highly correlated



- if $a_1 = a_2$, the ridge estimate of x_1 and x_2 will be equal (not obvious)
- the blue and green lines correspond to the variables x_1 and x_2
- the lasso does not reflect the relative importance of the two variables
- the elastic net selects the estimates of x_1 and x_2 together

Group lasso

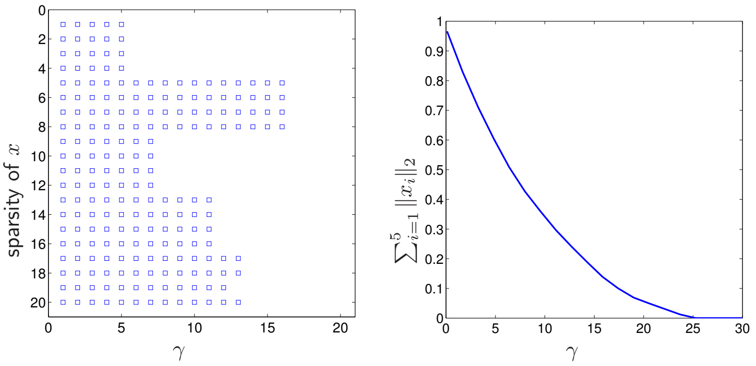
to have all entries in x within a *group* become zero simultaneously

let $x = (x_1, x_2, \ldots, x_K)$ where $x_j \in \mathbf{R}^n$

$$\text{minimize} \quad (1/2)\|Ax - y\|_2^2 + \gamma \sum_{j=1}^{K} \|x_j\|_2$$

- the sum of ℓ_2 norm is a generalization of ℓ_1-like penalty
- as γ is large enough, either x_j is entirely zero or all its element is nonzero
- when $n = 1$, group lasso reduces to the lasso
- a nondifferentiable convex problem but can be solved efficiently

generate the problem with $x = (x_1, x_2, \ldots, x_5)$ where $x_i \in \mathbf{R}^4$



- as γ increases, some of partition x_i becomes entirely zero
- as the sum of 2-norm is zero, the entire vector x is zero

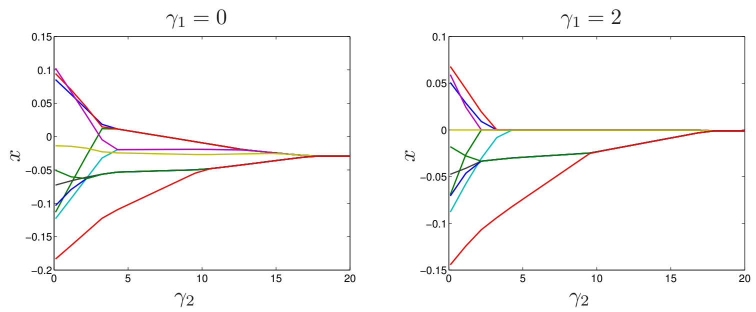
Fused lasso

to have neighboring variables similar and sparse

$$\underset{x \in \mathbf{R}^n}{\text{minimize}} \quad (1/2)\|Ax - y\|_2^2 + \gamma_1 \|x\|_1 + \gamma_2 \sum_{j=2}^{n} |x_j - x_{j-1}|$$

- the ℓ_1 penalty serves to shrink x_i toward zero
- the second penalty is ℓ_1-type encouraging some pairs of consecutive entries to be similar
- also known as **total variation denoising** in signal processing
- γ_1 controls the sparsity of x and γ_2 controls the similarity of neighboring entries
- a nondifferentiable convex problem but can be solved efficiently

generate $A \in \mathbf{R}^{100 \times 10}$ and vary γ_2 with two values of γ_1



- as γ_2, consecutive entries of x tend to be equal
- for a higher value of γ_1, some of the entries of x become zero

Robust least-squares

consider the LS problem

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2$$

but A may have variation or some uncertainty

we can treat the uncertainty in A in different ways

- A is deterministic but belongs to a set
- A is stochastic

Worst-case robust least-squares

describe the uncertainty by a set of possible values for A:

$$A \in \mathcal{A} \subseteq \mathbf{R}^{m \times n}$$

the problem is to minimize the worst-case error:

$$\underset{x}{\text{minimize}} \quad \sup_{A} \ \{\|Ax - y\|_2 \mid A \in \mathcal{A}\}$$

- always a convex problem
- its tractablity depends on the description of $\mathcal{A}$

Stochastic robust least-squares

when A is a random variable, so we can describe A as

$$A = \bar{A} + U,$$

where $\bar{A}$ is the average value of A and U is a random matrix

use the expected value of $\|Ax - y\|$ as the objective:

$$\underset{x}{\text{minimize}} \quad \mathbf{E}\|Ax - y\|_2^2$$

expanding the objective gives

$$\begin{aligned} \mathbf{E}\|Ax - y\|_2^2 &= (\bar{A}x - y)^T(\bar{A}x - y) + \mathbf{E}x^T U^T U x \\ &= \|\bar{A}x - y\|_2^2 + x^T P x \end{aligned}$$

where $P = \mathbf{E}U^T U$

this problem is equivalent to

$$\underset{x}{\text{minimize}} \quad \|\bar{A}x - y\|_2^2 + \|P^{1/2}x\|_2^2$$

with solution

$$x = (\bar{A}^T\bar{A} + P)^{-1}\bar{A}^T y$$

- a form of a regularized least-squares
- balance making $\bar{A}x - b$ small with the desire for a small x (so that the variation in Ax is small)
- Tikhonov regularization is a special case of robust least-squares: when U has zero mean and uncorrelated variables, $i.e.$, $\mathbf{E}U^TU = \delta I$

Summary

- variants of least-squares problems are regarded as optimization problems with quadratic cost objective
- most of them are convex programs and can be solved by many existing algorithms
- regularized least-squares are proposed to promote a certain structure in the solutions

References

Chapter 4 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 2-3 in
T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000

Chapter 6 in
S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge press, 2004

Exercises

8.1 Estimation of stable vector autoregressive processes. Consider an n-dimensional autoregressive model of order p,

$$y(t) = A_1 y(t-1) + A_2 y(t-2) + \cdots + A_p y(t-p) + \nu(t), \qquad (8.1)$$

where $A_k \in \mathbf{R}^{n \times n}$, for $k = 1, \ldots, p$, and $\nu(t)$ is zero-mean noise. This is a more general model than a scalar AR process described in equation (6.1) where we build a model for a group of variables $y_1(t), y_2(t), \ldots, y_n(t)$. In this exercise, we will formulate a least-squares problem to estimate $A_1, A_2, \ldots, A_p$ with conditions on these parameters.

(a) Given the measurements $y(1), y(2), \ldots, y(N)$, we find $A_1, \ldots, A_p$ such that

$$\sum_{k=p+1}^{N} \|y(k) - (A_1 y(k-1) + A_2 y(k-2) + \cdots + A_p y(k-p))\|_F^2,$$

is minimized. Show that the problem can be expressed as

$$\text{minimize} \;\; \|Y - AH\|_F, \qquad (8.2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $A = \begin{bmatrix} A_1 & A_2 & \cdots A_p \end{bmatrix}$. Determine Y and H.

(b) The problem (8.2) is an unconstrained optimization problem with variable A. Derive the zero-gradient condition and find its closed-form solution. Explain how you would solve for a numerical solution in MATLAB.

(c) In addition, we are interested in a solution of $A_1, A_2, \ldots, A_p$ that satisfies

$$\begin{aligned} (A_k)_{12} &= (A_k)_{21} = 0, \quad k = 1, 2, \ldots, p \\ (A_k)_{13} &= (A_k)_{31} = 0, \quad k = 1, 2, \ldots, p \end{aligned}$$

(A_k is a matrix of size $n \times n$; $(A_k)_{ij}$ means the (i, j) entry of A_k.) These conditions have a statistical interpretation. It means the components y_2 and y_1 have no interaction to each other, as well as the components y_1 and y_3. Show that the least-squares formulation (8.2) including the above constraints is a convex problem.

(d) Load `data-vec-ar` which contains $y(1), y(2), \ldots, y(N)$ in a variable `y` having size $n \times N$. Write MATLAB codes in CVX to solve the problem in part c). Use an AR model of order 3. Plot a graph of the first components of $y(t)$ and $\hat{y}(t)$ computed from the estimate of AR coefficients. Provide the estimate values of $A_1, A_2, \ldots, A_p$.

(e) Show that the model (8.1) can be represented in a state-space form:

$$x(t+1) = \mathcal{A}x(t) + \mathcal{B}u(t)$$

where $x(t) = (y(t-1), y(t-2), \ldots, y(t-p))$ and $u(t)$ is an input of the system. Determine $\mathcal{A}$ (the *dynamic matrix*). We will neglect how to derive $\mathcal{B}$ (the *input* matrix) for now. It is known that (8.1) is a stationary process if $\mathcal{A}$ is stable; all eigenvalues of $\mathcal{A}$ lie inside the unit disk, *i.e.*,

$$|\lambda(\mathcal{A})| < 1.$$

Repeat the part d) with a new data set in `data-vec-ar-short` where we have only a few samples of $y(t)$. Check whether the estimated model is stable.

(f) The previous part demonstrates a common problem that may occur in practice when we have a short sample size. A least-squares estimate does not necessarily yield a stable model. In this problem, we will develop a stability criterion that will be included as a constraint in the optimization problem.

- Show that if
$$\mathcal{A}^T\mathcal{A} \prec I$$
then all the eigenvalues of $\mathcal{A}$ lie inside the unit disk. *Hint.* Consider an eigenvalue problem; $\mathcal{A}\phi = \lambda\phi$.
- Show that the above condition can be expressed as a linear matrix inequality (LMI), which is linear in the optimization variables $A_1, A_2, \ldots, A_p$. *Hint.* Apply a Schur complement.

(g) Write a constrained optimization formulation of the least-squares problem of estimating $A_1, A_2, \ldots, A_p$, including the constraints in part c) and the stability constraint in f). Write MATLAB codes in CVX to solve the problem by using the data from `data-vec-ar-short.mat`. Verify if the resulting model is stable and provide the estimate values of $A_1, A_2, \ldots, A_p$.

8.2 Robust Least-squares. In this problem, we solve a least-squares problem
$$\text{minimize} \ \ \|Ax - b\|_2.$$
However, the matrix A has some uncertainty, and we model it as a random variable. The measurement vector b and the mean of A are given by
$$b = \begin{bmatrix} 2 \\ -3 \\ -1 \\ 1 \\ 3 \\ -5 \\ 5 \\ 3 \end{bmatrix}, \quad \overline{A} = \mathbf{E}[A] = \begin{bmatrix} 4 & 3 & 1 & 2 \\ 5 & -1 & -5 & 3 \\ 0 & -3 & -3 & 2 \\ 0 & -1 & -2 & -1 \\ -2 & -4 & 4 & 1 \\ 4 & -4 & -5 & -2 \\ -1 & 5 & -5 & 3 \\ -4 & 5 & -4 & -3 \end{bmatrix}$$
and its variance is given by
$$\mathbf{E}[(a_{ki} - \overline{a}_{ki})(a_{kj} - \overline{a}_{kj})] = \begin{cases} 4, & i = j \\ -1, & |i-j| = 1 \end{cases}, \quad k = 1, 2, \ldots, m$$
(We denote a_{ij} and $\overline{a}_{ij}$ the (i,j)th entries of A and $\overline{A}$, respectively.)

(a) Solve the robust least-squares problem
$$\text{minimize} \ \ \mathbf{E}\|Ax - b\|_2^2.$$
Explain how you would evaluate the cost objective. Give a numerical solution to this problem, and denoted it by x_{rls}.

(b) Compare the estimate from part a) with the least-squares problem that use the nominal value of A.
$$\text{minimize} \ \ \|\bar{A}x - b\|_2.$$
Denote the solution to this problem as x_{no}. Compare the fitting error $\|\bar{A}x - b\|$ between x_{rls} and x_{no}. Which estimate should yield the smallest error ? and why ?

(c) Discuss in which scenario the robust least-squares estimate will outperform the nominal least-squares. Provide a *numerical example* to show this.

8.3 Least-squares with uncertainty. Consider the least-squares problem: $\text{minimize} \|Ax - y\|_2$ but A has an uncertainty according to

$$A = \bar{A} + U$$

where $\bar{A}$ is the mean of A (deterministic matrix) and U is a zero-mean random matrix. The components of U, u_{ij}'s are i.i.d. Laplacian random variable with density function

$$f(u) = \frac{1}{2\alpha} e^{-\frac{|u|}{\alpha}}, \quad -\infty < u < \infty.$$

(a) Derive the robust least-squares estimate, $\hat{x}_{\text{rls}}$, which minimizes $\mathbf{E}\|Ax - y\|_2^2$.

(b) Use the data in `data-robust-LS-laplacian.mat` which contains $y, \bar{A}$, A (the uncertain matrix where we're not supposed to know) and x (the true value). Compute the least-squares estimate, $\hat{x}_{\text{ls}}$ and the robust least-squares estimate, $\hat{x}_{\text{rls}}$ using $\alpha = 1/2$. Write down the numerical values of these estimates.

(c) You can vary the parameter α and discuss how it affects $\hat{x}_{\text{rls}}$. What does it mean when α is very large?

Chapter 9

Instrumental variable methods

Any estimator including the least-squares estimate is a random variable. One desirable property of an estimator is the consistency, *i.e.*, whether the estimate converges to the true value in probabilistic sense, when data samples are large enough. We will see that it requires some restrict conditions on correlation between the regressor matrix and noise in the generating process, for a least-squares estimator to be consistent. These conditions are not satisfied in practice, when dynamical models are estimated. For this reason, the method of instrumental variable is introduced as a remedy for this issue.

Learning objectives of this chapter are

- to understand the concept of instruments and how to choose one to satisfy the condition for a consistent estimate,
- to be able to numerically solve for an instrumental estimate in a given problem.

9. Instrumental variable methods (IVM)

- review on the least-squares method
- description of IV methods
- choice of instruments
- extended IV methods

Revisit the LS method

using linear regression in dynamic models (SISO)

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \nu(t)$$

where $\nu(t)$ denotes the equation error

$$A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}, \quad B(q^{-1}) = b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}$$

we can write the dynamic as

$$y(t) = H(t)\theta + \nu(t)$$

where

$$\begin{aligned} H(t) &= \begin{bmatrix} -y(t-1) & \cdots & -y(t-n_a) & u(t-1) & \cdots & u(t-n_b) \end{bmatrix} \\ \theta &= \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} \end{bmatrix} \end{aligned}$$

the least-squares solution is the value of $\hat{\theta}$ that minimizes

$$\frac{1}{N}\sum_{t=1}^{N} \|\nu(t)\|^2$$

and is given by

$$\hat{\theta}_{\rm ls} = \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)^{-1} \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T y(t)\right)$$

to examine if $\hat{\theta}$ is consistent ($\hat{\theta} \to \theta$ as $N \to \infty$), note that

$$\begin{aligned} \hat{\theta}_{\rm ls} - \theta &= \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)^{-1} \left\{\frac{1}{N}\sum_{t=1}^{N} H(t)^T y(t) - \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)\theta\right\} \\ &= \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T H(t)\right)^{-1} \left(\frac{1}{N}\sum_{t=1}^{N} H(t)^T \nu(t)\right) \end{aligned}$$

hence, $\hat{\theta}_{\rm ls}$ is consistent if

- $\mathbf{E}[H(t)^T H(t)]$ is nonsingular
  satisfied in most cases, except u is not persistently exciting of order n
- $\mathbf{E}[H(t)^T \nu(t)] = 0$
  not satisfied in most cases, except $\nu(t)$ is white noise

summary:

- LS method for dynamical models is still certainly simple to use
- consistency is not readily obtained since the information matrix (H) is no longer deterministic
- it gives consistent estimates under restrictive conditions

to obtain consistency of the estimates, we modify the normal equation so that the output and the disturbance become uncorrelated

Solutions:

- PEM (Prediction error methods)
  - model the noise
  - applicable to general model structures
  - generally very good properties of the estimates
  - computationally quite demanding
- IVM (Instrumental variable methods)
  - do not model the noise
  - retain the simple LS structure
  - simple and computationally efficient approach
  - consistent for correlated noise
  - less robust and statistically less effective than PEM

Description of IVM

define $Z(t) \in \mathbf{R}^{n_\theta}$ with entries uncorrelated with $\nu(t)$

$$\frac{1}{N}\sum_{t=1}^{N} Z(t)^T \nu(t) = \frac{1}{N}\sum_{t=1}^{N} Z^T(t)[y(t) - H(t)\theta] = 0$$

The basic IV estimate of θ is given by

$$\hat{\theta} = \left(\frac{1}{N}\sum_{t=1}^{N} Z(t)^T H(t)\right)^{-1} \left(\frac{1}{N}\sum_{t=1}^{N} Z(t)^T y(t)\right)$$

provided that the inverse exists

- $Z(t)$ is called **the instrument** and is up to user's choice
- if $Z(t) = H(t)$, the IV estimate reduces to the LS estimate

Choice of instruments

the instruments $Z(t)$ have to be chosen such that

- $Z(t)$ is uncorrelated with noise $\nu(t)$

$$\mathbf{E}Z(t)^T\nu(t) = 0$$

- the matrix

$$\frac{1}{N}\sum_{t=1}^{N} Z(t)^T H(t) \to \mathbf{E}Z(t)^T H(t)$$

  has full rank

in other words, $Z(t)$ and $H(t)$ are correlated

one possibility is to choose

$$Z(t) = \begin{bmatrix} -\eta(t-1) & \dots & -\eta(t-n_a) & u(t-1) & \dots & u(t-n_b) \end{bmatrix}$$

where the signal $\eta(t)$ is obtained by filtering the input,

$$C(q^{-1})\eta(t) = D(q^{-1})u(t)$$

Special choices:

- let C, D be *a prior* estimates of A and B
- simple choice: pick $C(q^{-1}) = 1$, $D(q^{-1}) = -q^{-n_b}$

$$Z(t) = \begin{bmatrix} u(t-1) & \dots & u(t-n_a-n_b) \end{bmatrix}$$

  (with a reordering of $Z(t)$)

note that $u(t)$ and the noise $\nu(t)$ are assumed to be independent

Example via Yule-Walker equations

consider a scalar ARMA process:

$$A(q^{-1})y(t) = C(q^{-1})e(t)$$
$$y(t) + a_1 y(t-1) + \ldots + a_p y(t-p) = e(t) + c_1 e(t-1) + \ldots + c_r e(t-r)$$

where $e(t)$ is white noise with zero mean and variance λ^2

define $R_k = \mathbf{E}y(t)y(t-k)^T$, we obtain

$$R_k + a_1 R_{k-1} + \ldots + a_p R_{k-p} = 0, \quad k = r+1, r+2, \ldots$$

where we have used $\mathbf{E}C(q^{-1})e(t)y(t-k)^T = 0, \quad k > r$

this is referred to as **Yule-Walker equations** enumerate from $k = r+1, \ldots, r+m$, where $m \geq p$,

the Yule-Walker equations can be fit into a matrix form

$$\begin{bmatrix} R_r & R_{r-1} & \ldots & R_{r+1-p} \\ R_{r+1} & R_r & \ldots & R_{r+2-p} \\ \vdots & \vdots & \vdots & \\ R_{r+m-1} & R_{r+m-2} & \ldots & R_{r+m-p} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} R_{r+1} \\ R_{r+2} \\ \vdots \\ R_{r+m} \end{bmatrix} \triangleq \mathbf{R}\theta = -r$$

$\mathbf{R}$ and r are typically replaced by their sample esimates:

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{t=1}^{N} \begin{bmatrix} y(t-r-1) \\ \vdots \\ y(t-r-m) \end{bmatrix} \begin{bmatrix} y(t-1) & \ldots & y(t-p) \end{bmatrix}$$

$$\hat{r} = \frac{1}{N} \sum_{t=1}^{N} \begin{bmatrix} y(t-r-1) \\ \vdots \\ y(t-r-m) \end{bmatrix} y(t)$$

hence $\hat{\mathbf{R}}\hat{\theta} = -\hat{r}$ is equivalent to

$$\frac{1}{N} \sum_{t=1}^{N} \underbrace{\begin{bmatrix} y(t-r-1) \\ \vdots \\ y(t-r-m) \end{bmatrix}}_{Z(t)^T} \underbrace{\begin{bmatrix} -y(t-1) & \ldots & -y(t-p) \end{bmatrix}}_{H(t)}$$

$$= \frac{1}{N} \sum_{t=1}^{N} \begin{bmatrix} y(t-r-1) \\ \vdots \\ y(t-r-m) \end{bmatrix} y(t)$$

this is the relationship in basic IVM

$$\frac{1}{N} \sum_{t=1}^{N} Z(t)^T H(t)\theta = \frac{1}{N} \sum_{t=1}^{N} Z(t)^T y(t)$$

where we use the delayed output as an instrument

$$Z(t) = \begin{bmatrix} -y(t-r-1) & y(t-r-2) & \ldots & y(t-r-m) \end{bmatrix}^T$$

Extended IV methods

The *extended* IV method is to generalize the basic IV in two directions:

- allow $Z(t)$ to have more elements than θ ($n_z \geq n_\theta$)
- use prefiltered data

and the extended IV estimate of θ is obtained by

$$\min_\theta \left\| \sum_{t=1}^{N} Z(t)^T F(q^{-1})(y(t) - H(t)\theta) \right\|_W^2$$

where $\|x\|_W^2 = x^T W x$ and $W \succ 0$ is given

when $F(q^{-1}) = I, n_z = n_\theta, W = I$, we obtain the basic IV estimate

Define

$$A_N = \frac{1}{N}\sum_{t=1}^{N} Z(t)^T F(q^{-1})H(t)$$
$$b_N = \frac{1}{N}\sum_{t=1}^{N} Z(t)^T F(q^{-1})y(t)$$

then $\hat{\theta}$ is obtained by

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|b_N - A_N\theta\|_W^2$$

this is a weighted least-squares problem

the solution is given by

$$\hat{\theta} = (A_N^T W A_N)^{-1} A_N^T W b_N$$

note that this expression is only of theoretical interest

Theoretical analysis

Assumptions:

1. the system is strictly causal and asymptotically stable
2. the input $u(t)$ is persistently exciting of a sufficiently high order
3. the disturbance $\nu(t)$ is a stationary stochastic process with rational spectral density,
$$\nu(t) = G(q^{-1})e(t), \quad \mathbf{E}e(t)^2 = \lambda^2$$
4. the input and the disturbance are independent
5. the model and the true system have the same transfer function if and only if $\hat{\theta} = \theta$ (uniqueness)
6. the instruments and the disturbances are uncorrelated

from the system description

$$y(t) = H(t)\theta + \nu(t)$$

we have

$$\begin{aligned} b_N &= \frac{1}{N}\sum_{t=1}^{N} Z(t)^T F(q^{-1})y(t) \\ &= \frac{1}{N}\sum_{t=1}^{N} Z(t)^T F(q^{-1})H(t)\theta + \frac{1}{N}\sum_{t=1}^{N} Z(t)^T F(q^{-1})\nu(t) \\ &\triangleq A_N\theta + q_N \end{aligned}$$

thus,

$$\hat{\theta} - \theta = (A_N^T W A_N)^{-1} A_N^T W b_N - \theta = (A_N^T W A_N)^{-1} A_N^T W q_N$$

as $N \to \infty$,

$$(A_N^T W A_N)^{-1} A_N^T W q_N \to (A^T W A)^{-1} A^T W q$$

where

$$\begin{aligned} A &\triangleq \lim_{N\to} A_N = \mathbf{E}[Z(t)^T F(q^{-1}) H(t)] \\ q &\triangleq \lim_{N\to} q_N = \mathbf{E}[Z(t)^T F(q^{-1}) \nu(t)] \end{aligned}$$

hence, the IV estimate is consistent ($\lim_{N\to\infty} \hat{\theta} = \theta$) if

- A has full rank
- $\mathbf{E}[Z(t)^T F(q^{-1}) \nu(t)] = 0$

Numerical example

the true system is given by

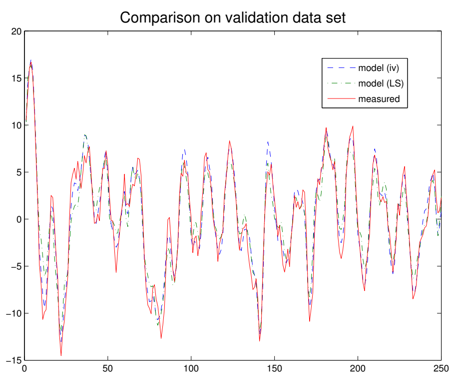
$$(1 - 1.5q^{-1} + 0.7q^{-2})y(t) = (1.0q^{-1} + 0.5q^{-2})u(t) + (1 - 1.0q^{-1} + 0.2q^{-2})e(t)$$

- ARMAX model
- $u(t)$ is from an ARMA process, independent of $e(t)$
- $e(t)$ is white noise withzero mean and variance 1
- $N = 250$ (number of data points)

estimation

- use ARX model and assume $n_a = 2, n_b = 2$
- compare the LS method with IVM

fit $\triangleq 100(1 - \|y - \hat{y}\|/\|y - \bar{y}\|)$



LS fit $= 66.97\%$, IV fit $= 77.50\%$

Example of MATLAB codes

```
%% Generate the data
close all; clear all;
N = 250; Ts = 1;
a = [1 -1.5 0.7]; b = [0 1 .5]; c = [1 -1 0.2];
Au = [1 -0.1 -0.12]; Bu = [0 1 0.2]; Mu = idpoly(Au,Bu,Ts);
u = sim(Mu,randn(2*N,1));   % u is ARMA process
noise_var = 1; e = randn(2*N,1);
M = idpoly(a,b,c,1,1,noise_var,Ts);
y = sim(M,[u e]);
uv = u(N+1:end); ev = e(N+1:end); yv = y(N+1:end);
u = u(1:N); e = e(1:N); y = y(1:N);
DATe = iddata(y,u,Ts); DATv = iddata(yv,uv,Ts);

%% Identification
na = 2; nb = 2; nc = 2;
theta_iv = iv4(DATe,[na nb 1]);    % ARX using iv4
theta_ls = arx(DATe,[na nb 1]);    % ARX using LS
```



```
%% Compare the measured output and the model output
[yhat2,fit2] = compare(DATv,theta_iv);
[yhat4,fit4] = compare(DATv,theta_ls);

figure;t = 1:N;
plot(t,yhat2{1}.y(t),'--',t,yhat4{1}.y(t),'-.',t,yv(t));
legend('model (iv)','model (LS)','measured')
title('Comparison on validation data set','FontSize',16);
```


References

Chapter 8 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Lecture on
Instrumental variable methods, System Identification (1TT875), Uppsala University, http://www.it.uu.se/edu/course/homepage/systemid/vt05

Exercises

9.1 IVM when the system does not belong to the model structure. Consider the system

$$y(t) = u(t-1) + u(t-2) + \nu(t) \tag{9.1}$$

where the input $u(t)$ and $\nu(t)$ are mutually independent white noise sequences of zero means and variances σ^2 and λ^2, respectively. Assume that the system identified using the model structure

$$y(t) + ay(t-1) = bu(t-1) + \varepsilon(t)$$

(a) Derive the correlation function between y and u, $R_{yu}(\tau)$ of the system (9.1) for $\tau = 0, 1, \ldots$.

(b) Derive the asymptotic (for $N \to \infty$) expression of the basic IV estimate based on the instrument

$$Z(t) = \begin{bmatrix} u(t-1) & u(t-2) \end{bmatrix}$$

of the parameters a and b.

(c) Derive the asymptotic expression of the LS estimate of a and b.

(d) Examine the stability properties of the models so obtained.

9.2 Consistency of IV estimate. Consider a scalar ARMAX model.

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = u(t-1) + \nu(t) \tag{9.2}$$

where $u(t)$ and $\nu(t)$ are independent zero-mean white noises with variances σ^2 and λ^2 respectively. The parameters a_1 and a_2 are such that the system is stable, *i.e.*, the two roots of $s^2 + a_1 s + a_2 = 0$ lie inside the unit circle. Assume the model in system identification is of the form

$$y(t) + cy(t-1) = bu(t-1) + \epsilon(t)$$

where $\epsilon(t)$ is the deviation between the model and the true system. The measurements $y(1), y(2), \ldots, y(N)$ are available in `data-consistency-iv2.mat`

(a) Derive the asymptotic expression of the basic IV estimate of c and b based on the instrument

$$Z(t) = \begin{bmatrix} u(t-1) & u(t-2) \end{bmatrix}.$$

(b) Derive the asymptotic expression of the LS estimate of c and b.

(c) Which method give a consistent estimate of b ? *i.e.*, Does $\hat{b} \to 1$ as $N \to \infty$?

(d) Do the estimates from both methods yield a stable model ?

(e) Verify the results in part c) and d) with simulation.

Chapter 10

Prediction Error Methods

The previous chapters on linear least-squares and instrumental variable method have made an important assumption on the model structure. They are feasible when the residual error between the measurement and the model output is *linear* in the model parameters. Such assumptions are not always satisfied in many model classes, for example, autoregressive moving average (ARMA) time series model. In this chapter, we introduce a prediction error method (PEM) that is applicable to a general model structure described in Chapter 4. The user defines a prediction model to compute the output at time t based on the data available up to time $t-1$. The principle of PEM is then to choose a cost objective as a loss function of prediction error and find the model parameters such that the loss function is minimized.

Learning objectives of this chapter are

- to understand basic elements of PEM, which are a model structure, a prediction model, and a criterion of the predictor,
- to derive an optimal prediction model for a given model structure,
- to apply numerical methods to solve a PEM estimator.

10. Prediction Error Methods (PEM)

- description
- optimal prediction
- Kalman filter
- statistical results
- computational aspects

Description

idea: determine the model parameter θ such that

$$e(t,\theta) = y(t) - \hat{y}(t|t-1;\theta)$$

is small

- $\hat{y}(t|t-1;\theta)$ is a prediction of $y(t)$ given the data up to and including time $t-1$ and based on θ

general linear predictor:

$$\hat{y}(t|t-1;\theta) = L(q^{-1};\theta)y(t) + M(q^{-1};\theta)u(t)$$

where L and M must contain one pure delay, $i.e.$,

$$L(0;\theta) = 0, M(0;\theta) = 0$$

Elements of PEM

one has to make the following choices, in order to define the method

- **Choice of model structure:** the parametrization of $G(q^{-1};\theta), H(q^{-1};\theta)$ and $\Lambda(\theta)$ as a function of θ
- **Choice of predictor:** the choice of filters L, M once the model is specified
- **Choice of criterion:** define a scalar-valued function of $e(t,\theta)$ that will assess the performance of the predictor

we commonly consider the **optimal mean square predictor**

the filters L and M are chosen such that the prediction error has small variance

Loss function

let N be the number of data points

sample covariance matrix:

$$R(\theta) = \frac{1}{N}\sum_{t=1}^{N} e(t,\theta)e^T(t,\theta)$$

$R(\theta)$ is a positive semidefinite matrix (and typically pdf when N is large)

loss function: scalar-valued function defined on positive matrices R

$$f(R(\theta))$$

f must be *monotonically increasing*, $i.e.$, let $X \succ 0$ and for any $\Delta X \succeq 0$

$$f(X + \Delta X) \geq f(X)$$

Example 1 $f(X) = \mathbf{tr}(WX)$ where $W \succ 0$ is a weighting matrix

$$f(X + \Delta X) = \mathbf{tr}(WX) + \mathbf{tr}(W\Delta X) \geq f(X)$$

($\mathbf{tr}(W\Delta X) \geq 0$ because if $A \succeq 0, B \succeq 0$, then $\mathbf{tr}(AB) \geq 0$)

Example 2 $f(X) = \det X$

$$\begin{aligned} f(X+\Delta X) - f(X) &= \det(X^{1/2}(I + X^{-1/2}\Delta X X^{-1/2})X^{1/2}) - \det X \\ &= \det X[\det(I + X^{-1/2}\Delta X X^{-1/2}) - 1] \\ &= \det X \left[\prod_{k=1}^{n}(1 + \lambda_k(X^{-1/2}\Delta X X^{-1/2})) - 1\right] \geq 0 \end{aligned}$$

the last inequalty follows from $X^{-1/2}\Delta X X^{-1/2} \succeq 0$, so $\lambda_k \geq 0$ for all k

both examples satisfy $f(X + \Delta X) = f(X) \iff \Delta X = 0$

Procedures in PEM

1. choose a model structure of the form

$$y(t) = G(q^{-1};\theta)u(t) + H(q^{-1};\theta)\nu(t), \quad \mathbf{E}\nu(t)\nu(t)^T = \Lambda(\theta)$$

2. choose a predictor of the form

$$\hat{y}(t|t-1;\theta) = L(q^{-1};\theta)y(t) + M(q^{-1};\theta)u(t)$$

3. select a criterion function $f(R(\theta))$
4. determine $\hat{\theta}$ that minimizes the loss function f

Least-squares method as a PEM

use linear regression in the dynamics of the form

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

we can write $y(t) = H(t)\theta + \varepsilon(t)$ where

$$\begin{aligned} H(t) &= \begin{bmatrix} -y(t-1) & \dots & -y(t-p) & u(t-1) & \dots & u(t-r) \end{bmatrix} \\ \theta &= \begin{bmatrix} a_1 & \dots a_p & b_1 & \dots & b_r \end{bmatrix}^T \end{aligned}$$

$\hat{\theta}$ that minimizes $(1/N)\sum_{t=1}^{N} \varepsilon^2(t)$ will give a prediction of $y(t)$:

$$\hat{y}(t) = H(t)\hat{\theta} = (1 - \hat{A}(q^{-1}))y(t) + \hat{B}(q^{-1})u(t)$$

hence, the prediction is in the form of

$$\hat{y}(t) = L(q^{-1};\theta)y(t) + M(q^{-1};\theta)u(t)$$

where $L(q^{-1};\theta) = 1 - \hat{A}(q^{-1})$ and $M(q^{-1};\theta) = B(q^{-1})$

note that $L(0;\theta) = 0$ and $M(0;\theta) = 0$,

so $\hat{y}$ uses the data up to time $t-1$ as required

the loss function in this case is $\mathbf{tr}(R(\theta))$ (quadratic in the prediction error)

Optimal prediction

consider the general linear model

$$y(t) = G(q^{-1})u(t) + H(q^{-1})\nu(t), \quad \mathbf{E}\nu(t)\nu(s)^T = \Lambda\delta_{t,s}$$

(we drop argument θ in G, H, Λ for notational convenience)

assumptions:

- $G(0) = 0, H(0) = I$
- $H^{-1}(q^{-1})$ and $H^{-1}(q^{-1})G(q^{-1})$ are asymptotically stable
- $u(t)$ and $\nu(s)$ are uncorrelated for $t < s$

rewrite $y(t)$ as

$$\begin{aligned}
y(t) &= G(q^{-1})u(t) + [H(q^{-1}) - I]\nu(t) + \nu(t) \\
&= G(q^{-1})u(t) + [H(q^{-1}) - I]H^{-1}(q^{-1})[y(t) - G(q^{-1})u(t)] + \nu(t) \\
&= \left\{H^{-1}(q^{-1})G(q^{-1})u(t) + [I - H^{-1}(q^{-1})]y(t)\right\} + \nu(t) \\
&\triangleq z(t) + \nu(t)
\end{aligned}$$

- $G(0) = 0$ and $H(0) = I$ imply $z(t)$ contains $u(s), y(s)$ up to time $t-1$
- hence, $z(t)$ and $\nu(t)$ are uncorrelated

let $\hat{y}(t)$ be an arbitrary predictor of $y(t)$

$$\begin{aligned}
\mathbf{E}[y(t) - \hat{y}(t)][y(t) - \hat{y}(t)]^T &= \mathbf{E}[z(t) + \nu(t) - \hat{y}(t)][z(t) + \nu(t) - \hat{y}(t)]^T \\
&= \mathbf{E}[z(t) - \hat{y}(t)][z(t) - \hat{y}(t)]^T + \Lambda \geq \Lambda
\end{aligned}$$

this gives a lower bound, Λ on the prediction error variance

the optimal predictor minimizes the prediction error variance

therefore, $\hat{y}(t) = z(t)$ and is given by

$$\hat{y}(t|t-1) = H^{-1}(q^{-1})G(q^{-1})u(t) + [I - H^{-1}(q^{-1})]y(t)$$

the corresponding prediction error can be written as

$$e(t) = y(t) - \hat{y}(t|t-1) = \nu(t) = H^{-1}(q^{-1})[y(t) - G(q^{-1})u(t)]$$

- from $G(0) = 0$ and $H(0) = I$, $\hat{y}(t)$ depends on past data up to time $t-1$
- these expressions suggest asymptotical stability assumptions in $H^{-1}G$ and H^{-1}

Optimal predictor for an ARMAX model

consider the model

$$y(t) + ay(t-1) = bu(t-1) + \nu(t) + c\nu(t-1)$$

where $\nu(t)$ is zero mean white noise with variance λ^2

for this particular case,

$$G(q^{-1}) = \frac{bq^{-1}}{1 + aq^{-1}}, \quad H(q^{-1}) = \frac{1 + cq^{-1}}{1 + aq^{-1}}$$

then the optimal predictor is given by

$$\hat{y}(t|t-1) = \frac{bq^{-1}}{1 + cq^{-1}}u(t) + \frac{(c-a)q^{-1}}{1 + cq^{-1}}y(t)$$

for computation, we use the recursion equation

$$\hat{y}(t|t-1) + c\hat{y}(t-1|t-2) = (c-a)y(t-1) + bu(t-1)$$

the prediction error is

$$e(t) = \frac{1+aq^{-1}}{1+cq^{-1}}y(t) - \frac{b}{1+cq^{-1}}u(t)$$

and it obeys

$$e(t) + ce(t-1) = y(t) + ay(t-1) - bu(t-1)$$

- the recursion equation requires an initial value, $i.e.$, $e(0)$
- setting $e(0) = 0$ is equivalent to $\hat{y}(0|-1) = y(0)$
- the transient is not significant for large t

Kalman Filter

for systems given in a state-space form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + \nu(t) \\ y(t) &= Cx(t) + \eta(t) \end{aligned}$$

- $\nu(t), \eta(t)$ are mutually uncorrelated white noise
- $\nu(t)$ and $\eta(t)$ have zero means and covariances R_1, R_2 resp.

the optimal *one-step* predictor of $y(t)$ is given by the **Kalman filter**

$$\begin{aligned} \hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + K[y(t) - C\hat{x}(t)] \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned}$$

where K is called the **steady-state Kalman gain**

the Kalman gain is given by

$$K = APC^T(CPC^T + R_2)^{-1}$$

where P is the positive solution to the **algebraic Riccati equation**:

$$P = APA^T + R_1 - APC^T(CPC^T + R_2)^{-1}CPA^T$$

- the predictor is *mean square optimal* if the disturbances are *Gaussian*
- for other distributions, the predictor is the **optimal linear predictor**

Example: Kalman filter of ARMAX model

consider the model

$$y(t) + ay(t-1) = bu(t-1) + \zeta(t) + c\zeta(t-1)$$

where $|c| < 1$ and $\zeta(t)$ is zero mean white noise with variance λ^2

this model can be written in state-space form as

$$\begin{aligned} x(t+1) &= \begin{bmatrix} -a & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 1 \\ c \end{bmatrix} \zeta(t+1) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \end{aligned}$$

with $\nu(t) \triangleq \begin{bmatrix} 1 \\ c \end{bmatrix} \zeta(t+1)$ and then $R_1 = \lambda^2 \begin{bmatrix} 1 & c \\ c & c^2 \end{bmatrix}, R_2 = 0$

solve the riccati equation and we can verify that P has the form

$$P = \lambda^2 \begin{bmatrix} 1+\alpha & c \\ c & c^2 \end{bmatrix}$$

where α satisfies

$$\alpha = (c-a)^2 + a^2\alpha - \frac{(c-a-a\alpha)^2}{1+\alpha}$$

there are two solutions, $\alpha = 0$ and $\alpha = c^2 - 1$

hence, we pick $\alpha = 0$ to make P positive definite

the Kalman gain is therefore

$$K = \begin{bmatrix} -a & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & c \\ c & c^2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left(\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & c \\ c & c^2 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)^{-1} = \begin{bmatrix} c-a \\ 0 \end{bmatrix}$$

the one-step optimal predictor of the output is

$$\begin{aligned} \hat{x}(t+1) &= \begin{bmatrix} -a & 1 \\ 0 & 0 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} c-a \\ 0 \end{bmatrix} \left(y(t) - \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}(t)\right) \\ &= \begin{bmatrix} -c & 1 \\ 0 & 0 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} b \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} c-a \\ 0 \end{bmatrix} y(t) \\ \hat{y}(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}(t) \end{aligned}$$

then it follows that

$$\begin{aligned} \hat{y}(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} q+c & -1 \\ 0 & q \end{bmatrix}^{-1} \begin{bmatrix} bu(t) + (c-a)y(t) \\ 0 \end{bmatrix} \\ &= \frac{1}{q+c}[bu(t) + (c-a)y(t)] \\ &= \frac{bq^{-1}}{1+cq^{-1}} u(t) + \frac{(c-a)q^{-1}}{1+cq^{-1}} y(t) \end{aligned}$$

same result as in page 10-13

Theoretical results

assumptions:

1. the data $\{u(t), y(t)\}$ are stationary processes
2. the input is persistently exciting
3. the Hessian $\nabla^2 f$ is nonsingular locally around the minimum points of $f(\theta)$
4. the filters $G(q^{-1}), H(q^{-1})$ are differentiable functions of θ

under these assumptions, the PEM estimate is **consistent**

$$\hat{\theta} \xrightarrow{p} \theta, \quad \text{as} \quad N \to \infty$$

Statistical efficiency

for Gaussian disturbances, the PEM method is **statistically efficient** if

- SISO: $f(\theta) = \mathbf{tr}(R(\theta))$
- MIMO:
  - $f(\theta) = \mathbf{tr}(WR(\theta))$ and $W = \Lambda^{-1}$ (the true covariance of noise)
  - $f(\theta) = \det(R(\theta))$

Computational aspects

I. Analytical solution exists

if the predictor is a linear function of the parameter

$$\hat{y}(t|t-1) = H(t)\theta$$

and the criterion function $f(\theta)$ is simple enough, $i.e.$,

$$f(\theta) = \mathbf{tr}(R(\theta)) = \frac{1}{N}\sum_{t=1}^{N} \|e(t,\theta)\|^2 = \frac{1}{N}\sum_{t=1}^{N} \|y(t) - H(t)\theta\|^2$$

it is clear that PEM is equivalent to the LS method

this holds for ARX or FIR models (but not for ARMAX and Output error models)

II. No analytical solution exists

it involves a nonlinear optimization for

- general criterion functions
- predictors that depend nonlinearly on the data

numerical algorithms: Newton-Ralphson, Gradient based methods, Grid search

typical issues in nonlinear minimization:

- solutions may consist of many local minima
- convergence rate and computational cost
- choice of initialization

Numerical example

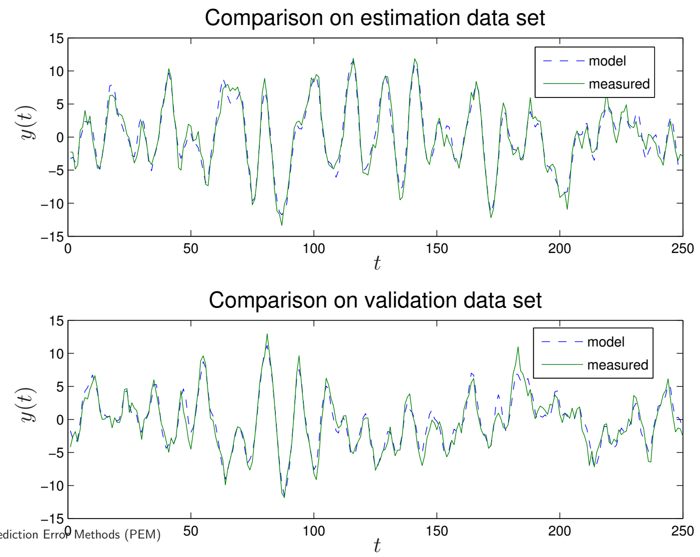
the true system is given by

$$(1 - 1.5q^{-1} + 0.7q^{-2})y(t) = (1.0q^{-1} + 0.5q^{-2})u(t) + (1 - 1.0q^{-1} + 0.2q^{-2})\nu(t)$$

- ARMAX model
- $u(t)$ is binary white noise, independent of $\nu(t)$
- $\nu(t)$ is white noise with zero mean and variance 1
- $N = 250$ (number of data points)

estimation

- assume the model structure and model order are known
- use `armax` command in MATLAB



Example of MATLAB codes

```
%% Generate the data
N = 250; Ts = 1; u_var = 1; noise_var = 1;
a = [1 -1.5 0.7]; b = [0 1 .5]; c = [1 -1 0.2];
u = sign(randn(2*N,1))*sqrt(u_var); v = randn(2*N,1);
M = idpoly(a,b,c,1,1,noise_var,Ts);
y = sim(M,[u v]);
uv = u(N+1:end); vv = v(N+1:end); yv = y(N+1:end);
u = u(1:N); v = v(1:N); y = y(1:N);
DATe = iddata(y,u,Ts); DATv = iddata(yv,uv,Ts);

%% Identification
na = 2; nb = 2; nc = 2;
theta_pem = armax(DATe,[na nb nc 1]);   % ARMAX using PEM

%% Compare the measured output and the model output
[yhat1,fit1] = compare(DATe,theta_pem);
[yhat2,fit2] = compare(DATv,theta_pem);
```



```
t = 1:N;
figure;
subplot(2,1,1);plot(t,yhat1{1}.y,'--',t,y);
legend('model','measured');
title('Comparison on estimation data set','FontSize',16);
ylabel('y');xlabel('t');
subplot(2,1,2);plot(t,yhat2{1}.y,'--',t,yv);legend('y2','y');
legend('model','measured');
title('Comparison on validation data set','FontSize',16);
ylabel('y');xlabel('t');
```


References

Chapter 7 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Lecture on
Prediction Error Methods, System Identification (1TT875), Uppsala University,
http://www.it.uu.se/edu/course/homepage/systemid/vt05

Exercises

10.1 Simulation and One-step Prediction. Consider an ARMAX model:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t-1) + b_2 u(t-2) + \nu(t) + c_1 \nu(t-1) + c_2 \nu(t-2)$$

where $u(t)$ is a known input and $\nu(t)$ is noise. Using the prediction error method to estimate $a_1, a_2, b_1, b_2, c_1, c_2$. The input and output measurements are available as variables `y` and `u` in `data-predicted-simulated.mat` Once an estimated model is available, one can use it to calculate the output of the system. There are two possibilities to implement this. The behavior of outputs calculated from both choices can be very different and will be illustrated in this exercise.

From the estimated ARMAX model, it follows that the relationship between input and output can be described from

$$y(t) = B(q, \hat{\theta})u(t) - A(q, \hat{\theta})y(t)$$

where $B(q, \hat{\theta}) = \hat{b}_1 q^{-1} + \hat{b}_2 q^{-2}$ and $A(q, \hat{\theta}) = \hat{a}_1 q^{-1} + \hat{a}_2 q^{-2}$. The one-step **predicted** output is calculated by

$$\hat{y}(t) = B(q, \hat{\theta})u(t) - A(q, \hat{\theta})y(t).$$

We have used the delayed outputs $\left(A(q, \hat{\theta})y(t) = \hat{a}_1 y(t-1) + \hat{a}_2 y(t-2)\right)$ or the past measurements to predict the output at time t. Alternatively, the past outputs can be replaced by the simulated values, and we call the output

$$\hat{y}(t) = B(q, \hat{\theta})u(t) - A(q, \hat{\theta})\hat{y}(t)$$

as **simulated** output. Suppose a new set of input and output measurements are available as `yv` and `uv` in `data-predicted-simulated.mat`. Compute the predicted output and simulated output and compare the plots of the errors. Which error has a shorter transient response ? Discuss the results.

Chapter 11

Statistical Estimation

We have seen linear least-squares, instrumental variable, and prediction error methods to estimate a model that best fits a data set. If we assume that the output measurement is generated from a data generating process, which is usually in the form of

$$y = f(x, \theta) + e$$

where $f(\cdot)$ is a true description (or true model) of data (could be, in general, nonlinear, and we never know this description), x is a possible explanatory variable for y, θ is the true parameter of f, and e is noise or uncertainty that makes our measurement ambiguous for model estimation. What these methods have in common is the fact that they do not make use of any *statistical property assumptions* about e in the *parameter estimation* process. They aim to minimize the residual errors in their own sense but a statistical distribution of the error is not applied as a prior information. In this chapter, estimation methods requires some statistical assumption about e. For example, we can assume that e is a normal variable and this certainly reflects in a change in the distribution of y as we can view y as a transformation of e. As a result, knowing some prior information about y should help improve estimation results. Statistical methods in this chapter includes mean-square estimation, maximum likelihood estimation, and maximum a posteriori estimation (MAP). These methods are used in various applications such as time series model estimation, or linear model with additive noise.

Learning objectives of this chapter are

- to understand the principle of mean-square estimation, maximum likelihood estimation and maximum a posteriori estimation,
- to apply the three methods to a linear model,
- to understand the importance of Cramer-Rao bound and how to use it to infer about the estimated covariance matrix.

11. Statistical Estimation

- conditional expectation
- mean square estimation (MSE)
- maximum likelihood estimation (ML)
- maximum a posteriori estimation (MAP)

Conditional expectation

let x, y be random variables with a joint density function $f(x, y)$
the conditional expectation of x given y is

$$\mathbf{E}[x|y] = \int x f(x|y) dx$$

where $f(x|y)$ is the conditional density: $f(x|y) = f(x, y)/f(y)$

Facts:

- $\mathbf{E}[x|y]$ is a function of y
- $\mathbf{E}[\mathbf{E}[x|y]] = \mathbf{E}[x]$
- for any scalar function $g(y)$ such that $\mathbf{E}[|g(y)|] < \infty$,

$$\mathbf{E}\left[(x - \mathbf{E}[x|y])g(y)\right] = 0$$

Mean square estimation

suppose x, y are random with a joint distribution

problem: find an estimate $h(y)$ that minimizes the mean square error:

$$\mathbf{E}\|x - h(y)\|^2$$

result: the optimal estimate in the mean square is *the conditional mean*:

$$h(y) = \mathbf{E}[x|y]$$

Proof. use the fact that $x - \mathbf{E}[x|y]$ is uncorrelated with any function of y

$$\begin{aligned}\mathbf{E}\|x - h(y)\|^2 &= \mathbf{E}\ \|x - \mathbf{E}[x|y] + \mathbf{E}[x|y] - h(y)\|^2 \\ &= \mathbf{E}\ \|x - \mathbf{E}[x|y]\|^2 + \mathbf{E}\ \|\mathbf{E}[x|y] - h(y)\|^2\end{aligned}$$

hence, the error is minimized only when $h(y) = \mathbf{E}[x|y]$

Gaussian case: x, y are jointly Gaussian: $(x, y) \sim \mathcal{N}(\mu, C)$ where

$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \quad C = \begin{bmatrix} C_x & C_{xy} \\ C_{xy}^T & C_y \end{bmatrix}$$

the conditional density function of x given y is also Gaussian with conditional mean

$$\mu_{x|y} = \mu_x + C_{xy} C_y^{-1}(y - \mu_y),$$

and conditional covariance matrix

$$C_{x|y} = C_x - C_{xy} C_y^{-1} C_{xy}^T$$

hence, for Gaussian distribution, the optimal mean square estimate is

$$\mathbf{E}[x|y] = \mu_x + C_{xy} C_y^{-1}(y - \mu_y),$$

the optimal estimate is **linear** in y

conclusions:

- $\mathbf{E}[x|y]$ is called the minimum mean square error (MMSE) estimator
- the MMSE estimator is typically nonlinear in y and is obtained from $f(x, y)$
- for Gaussian case, the MMSE estimator is **linear** in y
- the MMSE estimator must satisfy the **orthogonal principle**:

  $$[(x - \hat{x}_{\text{mmse}})g(y)] = 0$$

  where g is any function of y such that $\mathbf{E}[|g(y)|^2] < \infty$
- MMSE estimator can be difficult to evaluate, so one can consider a linear MMSE estimator

Linear MMSE estimator

the linear unbiased MMSE estimator takes the affine form:

$$h(y) = K\tilde{y} + \mathbf{E}[x], \quad \text{(with } \tilde{y} = y - \mathbf{E}[y])$$

important results: define $\tilde{x} = x - \mathbf{E}[x]$

- the linear MMSE estimator minimizes

  $$\mathbf{E}\|x - h(y)\|^2 = \mathbf{E}\|\tilde{x} - K\tilde{y}\|^2$$
- the linear MMSE estimator is

  $$h(y) = C_{xy} C_y^{-1}(y - \mathbf{E}[y]) + \mathbf{E}[x]$$
- the form of linear MMSE requires just covariance matrices of x, y
- it coincides with the optimal mean square estimate for Gaussian RVs

Wiener-Hopf equation

the optimal condition for linear MMSE estimator is

$$C_{xy} = KC_y$$

and is called the **Wiener-Hopf** equation

- obtained by differentiating the MSE w.r.t. K

$$\text{MSE} = \mathbf{E}\,\mathbf{tr}(\tilde{x} - K\tilde{y})(\tilde{x} - K\tilde{y})^T = \mathbf{tr}(C_x - C_{xy}K^T - KC_{yx} + KC_yK^T)$$

- also obtained from the condition

$$\mathbf{E}[(x - h(y))y^T] = 0 \quad \Rightarrow \quad \mathbf{E}[(\tilde{x} - K\tilde{y})\tilde{y}^T] = 0$$

(the optimal residual is uncorrelated with the observation y)

Minimizing the error covariance matrix

for any estimate $h(y)$, the covariance matrix of the corresponding error is

$$\mathbf{E}\left[(x - h(y))(x - h(y))^T\right]$$

the problem is to choose $h(y)$ to yield the minimum covariance matrix
(instead of minimizing the mean square norm)

we compare two matrices by

$$M \preceq N \quad \text{if} \quad M - N \preceq 0$$

or $M - N$ is nonpositive definite

now restrict to the linear case:

$$h(y) = Ky + c$$

the covariance matrix can be written as

$$(\mu_x - (K\mu_y + c))(\mu_x - (K\mu_y + c))^T + C_x - KC_{yx} - C_{xy}K^T + KC_yK^T$$

the objective is minimized with respect to c when

$$c = \mu_x - K\mu_y$$

(same as the best unbiased linear estimate of the mean square error)

the covariance matrix of the error is reduced to

$$f(K) = C_x - KC_{yx} - C_{xy}K^T + KC_yK^T$$

note that $f(K) \succeq 0$ because we can write $f(K)$ as

$$f(K) = \begin{bmatrix} -I & K \end{bmatrix} \begin{bmatrix} C_x & C_{xy} \\ C_{xy}^T & C_y \end{bmatrix} \begin{bmatrix} -I \\ K^T \end{bmatrix}$$

let K_0 be a solution to the Wiener-Hopf equation: $C_{xy} = K_0 C_y$

we can verify that

$$f(K) = f(K_0) + (K - K_0)C_y(K - K_0)^T$$

so $f(K)$ is minimized when $K = K_0$

the miminum covariance matrix is

$$f(K_0) = C_x - C_{xy}C_y^{-1}C_{xy}^T$$

for $C = \begin{bmatrix} C_x & C_{xy} \\ C_{xy}^T & C_y \end{bmatrix}$, note that

- the minimum covariance matrix is the Schur complement of C_x in C
- it is exactly a conditional covariance matrix for Gaussian variables

Maximum likelihood estimation

- $y = (y_1, \dots, y_m)$: the observations of random variables
- θ: unknown parameters to be estimated
- $f(y|\theta)$: the probability density function of y for a fixed θ

in ML estimation, we assume θ are **fixed** (and deterministic) parameters

to estimate θ from y, we maximize the density function for a given θ:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\ f(y|\theta)$$

- $f(y|\theta)$ is called the **likelihood function**
- θ is chosen so that the observed y becomes "as likely as possible"

Example 1: estimate the mean and covariance matrix of Gaussian RVs

- observe a sequence of *independent* random variables: $y_1, y_2, \dots, y_m$
- each y_k is an n-dimensional Gaussian: $y_k \sim \mathcal{N}(\mu, \Sigma)$, but μ, Σ are unknown
- the likelihood function of $y_1, \dots, y_N$ for given μ, Σ is

$$f(y_1, y_2, \dots, y_m|\mu, \Sigma) = \frac{1}{(2\pi)^{mn/2}} \cdot \frac{1}{|\Sigma|^{m/2}} \cdot \mathbf{exp} - \frac{1}{2}\sum_{k=1}^{m}(y_k - \mu)^T\Sigma^{-1}(y_k - \mu)$$

- to maximize f, it is convenient to consider the **log-likelihood function**: (up to a constant)

$$L(\mu, \Sigma) = \log f = \frac{m}{2}\log\det\Sigma^{-1} - \frac{1}{2}\sum_{k=1}^{m}(y_k - \mu)^T\Sigma^{-1}(y_k - \mu)$$

- the log-likelihood is concave in Σ^{-1}, μ, so the ML estimate satisfies the zero gradient conditions:

$$\frac{\partial L}{\partial \Sigma^{-1}} = \frac{m\Sigma}{2} - \frac{1}{2}\sum_{k=1}^{m}(y_k - \mu)(y_k - \mu)^T = 0$$
$$\frac{\partial L}{\partial \mu} = \sum_{k=1}^{m} \Sigma^{-1}(y_k - \mu) = 0$$

- we obtain the ML estimate of μ, Σ as

$$\hat{\mu}_{\mathrm{ml}} = \frac{1}{m}\sum_{k=1}^{m} y_k, \quad \hat{\Sigma}_{\mathrm{ml}} = \frac{1}{m}\sum_{k=1}^{m}(y_k - \hat{\mu}_{\mathrm{ml}})(y_k - \hat{\mu}_{\mathrm{ml}})^T$$

  - $\hat{\mu}_{\mathrm{ml}}$ is the sample mean
  - $\hat{\Sigma}_{\mathrm{ml}}$ is a (biased) sample covariance matrix

Example 2: linear measurements with i.i.d. noise

consider a linear measurement model

$$y = A\theta + v$$

$\theta \in \mathbf{R}^n$ is parameter to be estimated
$y \in \mathbf{R}^m$ is the measurement
$v \in \mathbf{R}^m$ is i.i.d. noise

(v_i are independent, identically distributed) with density f_v

the density function of $y - A\theta$ is therefore the same as v:

$$f(y|\theta) = \prod_{k=1}^{m} f_v(y_k - a_k^T\theta)$$

where a_k^T are the row vectors of A

the ML estimate of θ depends on the noise distribution f_v

suppose v_k is Gaussian with zero mean and variance σ

- the log-likelihood function is

$$L(\theta) = \log f = -(m/2)\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{m}(y_k - a_k^T\theta)^2$$

  (a_k^T are row vectors of A)

- therefore the ML estimate of θ is

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\ \|y - A\theta\|_2^2$$

- the solution of a least-squares problem

what about other distributions of v_k ?

Maximum a posteriori (MAP) estimation

assumptions:

- assume that θ is a *random variable*
- θ and y has a joint distribution $f(y, \theta)$

the MAP estimate of θ is given by

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, f_{\theta|y}(\theta|y)$$

- $f_{\theta|y}$ is called the **posterior** density of θ
- $f_{\theta|y}$ represents our knowledge of θ after we observe y
- MAP estimate is the value that maximizes the conditional density of θ, given the observed y

from Bayes rule, the MAP estimate is also obtained by

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \quad f_{y|\theta}(y|\theta) f_\theta(\theta)$$

taking logarithms, we can express $\hat{\theta}$ as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \quad \log f_{y|\theta}(y|\theta) + \log f_\theta(\theta)$$

- the only difference between ML and MAP estimate is the term $f_\theta(\theta)$
- f_θ is called the **prior** density, representing prior knowledge about θ
- $\log f_\theta(\theta)$ penalizes choices of θ that are unlikely to happen

under what condition on f_θ is the MAP estimate identical to the ML estimate ?

Example 3: linear measurement with IID noise

use the model in page 11-14 and assume θ has a prior density f_θ on $\mathbf{R}^n$

the MAP estimate can be found by solving

$$\text{maximize} \quad \log f_\theta(\theta) + \sum_{k=1}^{m} \log f_v(y_k - a_k^T \theta)$$

suppose $\theta \sim \mathcal{N}(0, \beta I)$ and $v_k \sim \mathcal{N}(0, \sigma)$, the MAP estimation is

$$\text{maximize} \quad -\frac{1}{\beta}\|\theta\|_2^2 - \frac{1}{\sigma^2}\|A\theta - y\|_2^2$$

conclusion: MAP estimate with a *Guassian prior* is the solution to a least-squares problem with ℓ_2 regularization

what if θ has a Laplacian distribution ?

Cramér-Rao inequality

for any **unbiased** estimator $\hat{\theta}$ with the covariance matrix of the error:

$$\mathbf{cov}(\hat{\theta}) = \mathbf{E}(\theta - \hat{\theta})(\theta - \hat{\theta})^T,$$

we always have a lower bound on $\mathbf{cov}(\hat{\theta})$:

$$\mathbf{cov}(\hat{\theta}) \succeq \left[\mathbf{E}(\nabla_\theta \log f(y|\theta))^T (\nabla_\theta \log f(y|\theta))\right]^{-1} = -\left[\mathbf{E}\nabla_\theta^2 \log f(y|\theta)\right]^{-1}$$

- $f(y|\theta)$ is the density function of observations y for a given θ
- the RHS is called the **Cramér-Rao** lower bound
- provide the minimal covariance matrix over all possible estimators $\hat{\theta}$
- $J \triangleq \mathbf{E}\nabla_\theta^2 \log f(y|\theta)$ is called the **Fisher information matrix**
- an estimator for which the C-R equality holds is called **efficient**

Proof of the Cramér-Rao inequality

since $f(y|\theta)$ is a density function and $\hat{\theta}$ is unbiased, we have

$$1 = \int f(y|\theta)dy, \quad \theta = \int \hat{\theta}(y) f(y|\theta) dy$$

differentiate the eqs w.r.t. θ and use $\nabla_\theta \log f(y|\theta) = \dfrac{\nabla_\theta f(y|\theta)}{f(y|\theta)}$

$$0 = \int \nabla_\theta \log f(y|\theta) f(y|\theta) dy, \quad I = \int \hat{\theta}(y) \nabla_\theta \log f(y|\theta) f(y|\theta) dy$$

these two identities can be expressed as

$$\mathbf{E}\left[(\hat{\theta}(y) - \theta)\nabla_\theta \log f(y|\theta)\right] = I$$

($\mathbf{E}$ is taken w.r.t y, and θ is fixed)

consider a positive semidefinite matrix

$$\mathbf{E}\begin{bmatrix} \hat{\theta}(y) - \theta \\ (\nabla_\theta \log f(y|\theta))^T \end{bmatrix} \begin{bmatrix} \hat{\theta}(y) - \theta \\ (\nabla_\theta \log f(y|\theta))^T \end{bmatrix}^T \succeq 0$$

expand the product into the form

$$\begin{bmatrix} A & I \\ I & D \end{bmatrix}$$

where $A = \mathbf{E}(\hat{\theta}(y) - \theta)(\hat{\theta}(y) - \theta)^T$ and

$$D = \mathbf{E}(\nabla_\theta \log f(y|\theta))^* (\nabla_\theta \log f(y|\theta))$$

the Schur complement of the $(1,1)$ block must be nonnegative:

$$A - ID^{-1}I \succeq 0$$

which implies the Cramér Rao inequality

now it remains to show that

$$\mathbf{E}(\nabla_\theta \log f(y|\theta))^T(\nabla_\theta \log f(y|\theta)) = -\mathbf{E}\nabla_\theta^2 \log f(y|\theta)$$

from the equation

$$0 = \int \nabla_\theta \log f(y|\theta) f(y|\theta) dy,$$

differentiating on both sides gives

$$0 = \int \nabla_\theta^2 \log f(y|\theta) f(y|\theta) dy + \int \nabla_\theta \log f(y|\theta)^T \nabla_\theta \log f(y|\theta) f(y|\theta) dy$$

or

$$-\mathbf{E}[\nabla_\theta^2 \log f(y|\theta)] = \mathbf{E}[\nabla_\theta \log f(y|\theta)^T \nabla_\theta \log f(y|\theta)]$$

Example of computing the Cramér Rao bound

revisit a linear model with correlated Gaussian noise:

$$y = A\theta + v, \quad v \sim \mathcal{N}(0, \Sigma), \quad \Sigma \text{ is known}$$

the density function $f(y|\theta)$ is given by $f_v(y - A\theta)$ which is Gaussian

$$\begin{aligned}
\log f(y|\theta) &= -\frac{1}{2}(y - A\theta)^T \Sigma^{-1}(y - A\theta) - \frac{m}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma \\
\nabla_\theta \log f(y|\theta) &= A^T\Sigma^{-1}(y - A\theta) \\
\nabla_\theta^2 \log f(y|\theta) &= -A^T\Sigma^{-1}A
\end{aligned}$$

hence, for any unbiased estimate $\hat{\theta}$,

$$\mathbf{cov}(\hat{\theta}) \succeq (A^T\Sigma^{-1}A)^{-1}$$

Linear models with additive noise

estimate parameters in a linear model with addtive noise:

$$y = A\theta + v, \quad v \sim \mathcal{N}(0, \Sigma), \quad \Sigma \text{ is known}$$

and we explore several estimates from the following approaches

- no use of noise information
  - least-squares estimate (LS)
- use information about the noise (Gaussian distribution, Σ)

assume θ is a fixed parameter	assume $\theta \sim \mathcal{N}(0, \Lambda)$
weighted least-squares (WLS)	minimum mean square (MMSE)
best linear unbiased (BLUE)	maximum a posteriori (MAP)
maximum likelihood (ML)	

Least-squares: $\hat{\theta}_{\rm ls} = (A^TA)^{-1}A^Ty$ and is unbiased

$$\mathbf{cov}(\hat{\theta}_{\rm ls}) = \mathbf{cov}((A^TA)^{-1}A^Tv) = (A^TA)^{-1}A^T\Sigma A(A^TA)^{-1}$$

we can verifty that $\mathbf{cov}(\hat{\theta}_{\rm ls}) \succeq (A^T\Sigma^{-1}A)^{-1}$

(the error covariance matrix is bigger than the CR bound)

however the bound is tight when the noise covariance is diagonal:

$$\Sigma = \sigma^2 I$$

(the noise v_k are uncorrelated)

Weighted least-squares: for a given weight matrix $W \succ 0$

$$\hat{\theta}_{\rm wls} = (A^TWA)^{-1}A^TWy, \quad \text{and is unbiased}$$

$$\begin{aligned}\mathbf{cov}(\hat{\theta}_{\rm wls}) &= \mathbf{cov}((A^TWA)^{-1}A^TWv) \\ &= (A^TWA)^{-1}A^TW\Sigma WA(A^TWA)^{-1}\end{aligned}$$

$\mathbf{cov}(\hat{\theta}_{\rm wls})$ attains the minimum (the CR bound) when $W = \Sigma^{-1}$

$$\hat{\theta}_{\rm wls} = (A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}y$$

interpretation:

- large Σ_{ii} means the ith measurement is highly uncertain
- should put less weight on the corresponding ith entry of the residual

Maximum likelihood

from $f(y|\theta) = f_v(y - A\theta)$,

$$\log f(y|\theta) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma - \frac{1}{2}(y - A\theta)^T\Sigma^{-1}(y - A\theta)$$

the zero gradient condition gives

$$\nabla_\theta \log f(y|\theta) = A^T\Sigma^{-1}(y - A\theta) = 0$$

$$\hat{\theta}_{\rm ml} = (A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}y$$

$\hat{\theta}_{\rm ml}$ is also efficient (achieves the minimum covariance matrix)

moreover, we can verify that

$$\hat{\theta}_{\rm ml} = \hat{\theta}_{\rm wls} = \hat{\theta}_{\rm blue}$$

minimum mean square estimate:

- θ is random and independent of v
- $\theta \sim \mathcal{N}(0, \Lambda)$

hence, θ and y are jointly Gaussian with zero mean and the covariance:

$$C = \begin{bmatrix} C_\theta & C_{\theta y} \\ C_{\theta y}^T & C_{yy} \end{bmatrix} = \begin{bmatrix} \Lambda & \Lambda A^T \\ A\Lambda & A\Lambda A^T + \Sigma \end{bmatrix}$$

$\hat{\theta}_{\mathrm{mmse}}$ is essentially the conditional mean (readily computed for Gaussian)

$$\hat{\theta}_{\mathrm{mmse}} = \mathbf{E}[\theta|y] = C_{\theta y} C_{yy}^{-1} y = \Lambda A^T (A\Lambda A^T + \Sigma)^{-1} y$$

alternatively, we claim that $\mathbf{E}[\theta|y]$ is linear in y (because θ, y are Gaussian)

$$\hat{\theta}_{\mathrm{mmse}} = \hat{\theta}_{\mathrm{lms}} = Ky$$

and K can be computed from the Wiener-Hopf equation

Maximum a posteriori:

- θ is random and independent of v
- $\theta \sim \mathcal{N}(0, \Lambda)$

the MAP estimate can be found by solving

$$\hat{\theta}_{\mathrm{map}} = \underset{\theta}{\mathrm{argmax}} \ \ \log f(\theta|y) = \underset{\theta}{\mathrm{argmax}} \ \ \log f(y|\theta) + \log f(\theta)$$

without having to solve this problem, it is immediate that

$$\hat{\theta}_{\mathrm{map}} = \hat{\theta}_{\mathrm{mmse}}$$

since for Gaussian density function, $\mathbf{E}[\theta|y]$ maximizes $f(\theta|y)$

nevertheless, we can write down the posteriori density function

$$\log f(y|\theta) = -\frac{1}{2}\log\det\Sigma - \frac{1}{2}(y - A\theta)^T \Sigma^{-1} (y - A\theta)$$
$$\log f(\theta) = -\frac{1}{2}\log\det\Lambda - \frac{1}{2}\theta^T \Lambda^{-1}\theta$$

(these terms are up to a constant)

the MAP estimate satisfies the zero gradient (w.r.t. θ) condition:

$$-A^T \Sigma^{-1}(y - A\theta) + \Lambda^{-1}\theta = 0$$

which gives

$$\hat{\theta}_{\mathrm{map}} = (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1} A^T \Sigma^{-1} y$$

- $\hat{\theta}_{\mathrm{map}}$ is clearly similar to $\hat{\theta}_{\mathrm{ml}}$ except the extra term Λ^{-1}
- when $\Lambda = \infty$ or *maximum ignorance*, it reduces to ML estimate

- from $\hat{\theta}_{\text{mmse}} = \hat{\theta}_{\text{map}}$, it is interesting to verify

$$\Lambda A^T (A\Lambda A^T + \Sigma)^{-1} y = (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1} A^T \Sigma^{-1} y$$

(see the proof next page - it can be skipped)

define $H = (A\Lambda A^T + \Sigma)^{-1} y$ and we have

$$A\Lambda A^T H + \Sigma H = y$$

we start with the expression of $\hat{\theta}_{\text{lms}}$

$$\begin{aligned}
\hat{\theta}_{\text{mmse}} &= \Lambda A^T (A\Lambda A^T + \Sigma)^{-1} y = \Lambda A^T H \\
A\hat{\theta}_{\text{mmse}} &= A\Lambda A^T H = y - \Sigma H \\
\Lambda A^T \Sigma^{-1} A \theta_{\text{mmse}} &= \Lambda A^T \Sigma^{-1} y - \Lambda A^T H \\
&= \Lambda A^T \Sigma^{-1} y - \hat{\theta}_{\text{mmse}} \\
(I + \Lambda A^T \Sigma^{-1} A)\hat{\theta}_{\text{mmse}} &= \Lambda A^T \Sigma^{-1} y \\
(\Lambda^{-1} + A^T \Sigma^{-1} A)\hat{\theta}_{\text{mmse}} &= A^T \Sigma^{-1} y \\
\hat{\theta}_{\text{mmse}} &= (\Lambda^{-1} + A^T \Sigma^{-1} A)^{-1} A^T \Sigma^{-1} y \triangleq \hat{\theta}_{\text{map}}
\end{aligned}$$

to compute the covariance matrix of the error, we use $\hat{\theta}_{\text{map}} = \mathbf{E}[\theta|y]$

$$\mathbf{cov}(\hat{\theta}_{\text{map}}) = \mathbf{E}\left[(\theta - \mathbf{E}[\theta|y])(\theta - \mathbf{E}[\theta|y])^T\right]$$

use the fact that the optimal residual is uncorrelated with y

$$\mathbf{cov}(\hat{\theta}_{\text{map}}) = \mathbf{E}\left[(\theta - \mathbf{E}[\theta|y])\theta^T\right]$$

next $\hat{\theta}_{\text{map}} = \mathbf{E}[\theta|y]$ is a linear function in y

$$\begin{aligned}
\mathbf{cov}(\hat{\theta}_{\text{map}}) = C_\theta - KC_{y\theta} &= \Lambda - (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1} A^T \Sigma^{-1} A \Lambda \\
&= (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1}\left[(A^T \Sigma^{-1} A + \Lambda^{-1})\Lambda - A^T \Sigma^{-1} A \Lambda\right] \\
&= (A^T \Sigma^{-1} A + \Lambda^{-1})^{-1} \preceq (A^T \Sigma^{-1} A)^{-1}
\end{aligned}$$

$\hat{\theta}_{\text{map}}$ yields a smaller covariance matrix than $\hat{\theta}_{\text{ml}}$ as it should be
(ML does not use a prior knowledge about θ)

Summary

- estimate methods in this section require statistical properties of random entities in the model
- minimum-mean-square estimate is the conditional mean and typically a nonlinear function in the measurement data
- a maximum-likelihood estimation is a nonlinear optimization problem; it can reduce to have a closed-form solution in some special case of noise distribution (e.g. Gaussian)
- a maximum a posteriori estimation takes model parameters as random variables; it requires a prior distribution of these parameters

References

Appendix B in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 2-3 in
T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, 2000

Chapter 9 in
A. V. Balakrishnan, *Introduction to Random Processes in Engineering*, John Wiley & Sons, Inc., 1995

Chapter 7 in
S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge press, 2004

Exercises

11.1 ML estimation for some common noise densities. Consider a linear measurement model

$$y = A\theta + v \tag{11.1}$$

where $y \in \mathbf{R}^m$ is the measurement, $\theta \in \mathbf{R}^n$ is the parameter to be estimated and $v \in \mathbf{R}^m$ is i.i.d noise (v_i are independent, identically distributed) with density f_v. In class, we learn that if we assume v_i is Gaussian with zero mean and variance σ, then the ML estimate of θ is given by

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \|A\theta - y\|_2^2$$

(The ML estimate is identical to the least-squares estimate.) In this problem, we explore the formulation of the ML problem if we assume v_i has other distributions. Formulate the ML estimation into an optimization problem or some constraints on A, θ, y for the following density functions.

(a) *Laplacian noise.* v_i are Laplacian with density function $p(z) = (1/2a)e^{-|z|/a}$ where $a > 0$.

(b) *Uniform noise.* v_i are uniformly distributed on $[-a, a]$ with density function $p(z) = 1/(2a)$ on $[-a, a]$.

(Your formulation should be simplified enough so that the resulting problem can be readily solved.)

11.2 MAP estimation of a linear model for some common noise densities. Suppose the measurement y and parameter x are related by

$$y = Ax + v \tag{11.2}$$

where $y \in \mathbf{R}^m$ and $x \in \mathbf{R}^n$, and v_i are i.i.d. with Gaussian distribution of zero mean and variance σ^2. Find the MAP estimates of x when

- x_i are independent Gaussian with zero mean and variance λ^2.
- x_i are independent Laplacian with the density function

$$p(x_i) = \frac{1}{2\lambda} e^{-|x_i|/\lambda}, \quad i = 1, 2, \ldots, n, \quad \lambda > 0.$$

(a) Show that each of the MAP estimates is a solution of a regularized least-squares problem.

(b) Use the data given in `data-map-linmodel.mat` with parameter $\sigma = 1, \lambda = 1$ to find the numerical values of the ML estimate, and the two MAP estimates. Compare the estimate result with the true value of x given in vector `x`.

(c) Plot three subgraphs; each of them illustrates a comparison of the estimation result between the true parameter x and each of the estimates. Use `stem` command to plot the values of x and $\hat{x}$. Discuss the results. Which estimate yields the smallest error ? Compute $\|x - \hat{x}\|$.

(d) Plot a histogram of the two MAP estimates by using `hist` command. What are the main features you observe from the histograms ? If a sparse estimate of x is favored, which estimation method would you choose ?

11.3 Nonlinear and linear estimators of the mean. Consider the observations

$$y(t) = x + v(t), \quad t = 1, 2, \ldots, N$$

where x and v are independent real-valued random variables, $v(t)$ is a white-noise Gaussian process with zero-mean and unit variance, and x takes the values of ± 1 with *equal* probability. Note that x takes one value for all t but its value is random. In this problem, we are given the observations $y(1), y(2), \ldots, y(N)$ and we would like to find the following estimates of x:

- The least-squares estimate, $\hat{x}_{\text{ls}}$.
- The least-mean-squares (or minimum-mean-square-error) estimate, $\hat{x}_{\text{lms}}$.
- The best linear unbiased estimate (BLUE), $\hat{x}_{\text{blue}}$.

and discuss about their properties. To simplify your analysis, you can write the process in the vector form as

$$y = x\mathbf{1} + v$$

where $\mathbf{1}$ is the all-one vector (with length N).

(a) Derive and give the closed-form expression of the least-squares estimate of x.

(b) Show that the least-mean-squares (lms) estimate of x is given by

$$\hat{x}_{\text{lms}} = \tanh\left(\sum_{t=1}^{N} y(t)\right).$$

Note that the minimum-mean-squared-error (MMSE) and the least-mean-squares estimates are the same. They can be used interchangeably. *Hint.* $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

(c) Derive the BLUE estimate of x.

(d) We provide 100 data sets of measurements y; each of which contains N time points. The observations y from different data set is generated from different values of x (but one data set corresponds to one value of x). Write MATLAB codes to compute the LS, LMS and BLUE estimates of x for each data set. You can load the data from `data-nonlinest.mat`. The variables are `y,x,N,SAMPLES` where

- `y` has size `N` $\times$ `SAMPLES` where $N = 5$ and `SAMPLES` is the number of data sets, which is 100.
- `x` has size $1 \times$ `SAMPLES`, the true values of x.

Save your MATLAB M-file as `yourname.m`

(e) From each data set compute the following quantities:

$$\sqrt{\sum_{t=1}^{N} (y(t) - \hat{x})^2}, \quad \text{and} \quad |x - \hat{x}|^2$$

for the three estimates. Plot two figures: (i) $\|y - \hat{x}\mathbf{1}\|$ versus data set index and the figure contains three plots from the three estimates, (ii) $\hat{x}$ versus data set index and the figure contains four plots from the three estimates and the true value of x. Save the two figures as `resid_normy.pdf` and `xhat.pdf`. From above quantities and the plots, *discuss and compare* the three estimates. Which one you are going to use ? Justify your answer.

11.4 Cramér-Rao bound. Consider a problem of estimating the mean a of the process

$$y(t) = a + \nu(t)$$

where $\nu \sim \mathcal{N}(0, a)$. In this problem, the noise variance is unknown and is assumed to be as high as the process mean.

(a) Determine $\hat{a}_{\mathrm{ls}}$, the least-squares estimate of a

(b) Determine $\hat{a}_{\mathrm{ml}}$, the maximum likelihood estimate of a. While the noise is Gaussian, why is $\hat{a}_{\mathrm{ml}}$ different from $\hat{a}_{\mathrm{ls}}$ in this case ?

(c) Are $\hat{a}_{\mathrm{ml}}$ and $\hat{a}_{\mathrm{ls}}$ consistent ?, *i.e.*, $\hat{a} \to a$ as $N \to \infty$?

(d) Compute the Cramér-Rao bound of any unbiased estimators of a

(e) Is $\hat{a}_{\mathrm{ls}}$ efficient ?, *i.e.*, the variance of $\hat{a}_{\mathrm{ls}}$ achieves the Cramér-Rao bound ?

Chapter 12

Subspace identification

Subspace identification is a method for estimating the system parameters in a state-space representation. It applies geometric tools in linear algebra, rather than other statistical estimation methods. To understand subspace identification, we consider the system of two cases: noiseless case (deterministic) and stochastic case without input. Each of these cases, the estimation consists of two main steps: to determine state sequences and to compute the system matrices.

To begin with, we review some linear algebra tools that are SVD factorization, orthogonal and oblique projections. The result of deterministic and stochastic cases are presented. Subsequently, we explain how to combine the results from both cases to estimate the system in a general setting.

Learning objectives of this chapter are

- to explain the oblique projection which is served as a basis for subspace identification,
- to explain the combination of deterministic and stochastic identification applied to state-space models,
- to apply existing numerical methods for subspace identification.

12. Subspace methods

- main idea
- notation
- geometric tools
- deterministic subspace identification
- stochastic subspace identification
- combination of deterministic-stochastic identifications
- MATLAB examples

Introduction

consider a stochastic discrete-time linear system

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad y(t) = Cx(t) + Du(t) + v(t)$$

where $x \in \mathbf{R}^n, u \in \mathbf{R}^m, y \in \mathbf{R}^l$ and $\mathbf{E}\begin{bmatrix} w(t) \\ v(t) \end{bmatrix}\begin{bmatrix} w(t) \\ v(t) \end{bmatrix}^T = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta(t,s)$

problem statement: given input/output data $(u(t), y(t))$ for $t = 0, \ldots, N$

- find an appropriate order n
- estimate the system matrices (A, B, C, D)
- estimate the noice covariances: Q, R, S

Basic idea

the algorithm involves two steps:

1. estimation of state sequence:
   - obtained from input-output data
   - based on linear algebra tools (QR, SVD)
2. least-squares estimation of state-space matrices (once states $\hat{x}$ are known)

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \underset{A,B,C,D}{\text{minimize}} \left\| \begin{bmatrix} \hat{x}(t+1) & \hat{x}(t+2) & \cdots & \hat{x}(t+j) \\ y(t) & y(t+1) & \cdots & y(t+j-1) \end{bmatrix} \right. \\ \left. - \begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} \hat{x}(t) & \hat{x}(t+1) & \cdots & \hat{x}(t+j-1) \\ u(t) & u(t+1) & \cdots & u(t+j-1) \end{bmatrix} \right\|_F^2$$

and $\hat{Q}, \hat{S}, \hat{R}$ are estimated from the least-squares residuals

Geometric tools

- notation and system related matrices
- row and column spaces
- orthogonal projections
- oblique projections

System related matrices

extended observability matrix

$$\Gamma_i = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} \in \mathbf{R}^{li \times n}, \quad i > n$$

extended controllability matrix

$$\Delta_i = \begin{bmatrix} A^{i-1}B & A^{i-2}B & \cdots & AB & B \end{bmatrix} \in \mathbf{R}^{n \times mi}$$

a block Toeplitz

$$H_i = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ CA^{i-2}B & CA^{i-3}B & CA^{i-4}B & & D \end{bmatrix} \in \mathbf{R}^{li \times mi}$$

Notation and indexing

we use subscript i for time indexing

$$X_i = \begin{bmatrix} x_i & x_{i+1} & \cdots & x_{i+j-2} & x_{i+j-1} \end{bmatrix} \in \mathbf{R}^{n \times j}, \quad \text{usually } j \text{ is large}$$

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \\ \hline u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right] = \left[\begin{array}{c} U_{0|i-1} \\ \hline U_{i|2i-1} \end{array}\right] = \left[\begin{array}{c} U_p \\ \hline U_f \end{array}\right]$$

- $U_{0|2i-1}$ has $2i$ blocks and j columns and usually j is large
- U_p contains the past inputs and U_f contains the future inputs

we can shift the index so that the top block contain the row of u_i

$$U_{0|2i-1} \triangleq \left[\begin{array}{ccccc} u_0 & u_1 & u_2 & \cdots & u_{j-1} \\ u_1 & u_2 & u_3 & \cdots & u_j \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{i-1} & u_i & u_{i+1} & \cdots & u_{i+j-2} \\ u_i & u_{i+1} & u_{i+2} & \cdots & u_{i+j-1} \\ \hline u_{i+1} & u_{i+2} & u_{i+3} & \cdots & u_{i+j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ u_{2i-1} & u_{2i} & u_{2i+1} & \cdots & u_{2i+j-2} \end{array}\right] = \left[\begin{array}{c} U_{0|i} \\ \hline U_{i+1|2i-1} \end{array}\right] = \left[\begin{array}{c} U_p^+ \\ \hline U_f^- \end{array}\right]$$

- $+/-$ can be used to shift the border between the past and the future block
- $U_p^+ = U_{0|i}$ and $U_f^- = U_{i+1|2i-1}$
- the output matrix $Y_{0|2i-1}$ is defined in the same way
- $U_{0|2i-1}$ and $Y_{0|2i-1}$ are **block Hankel** matrices (same block along anti-diagonal)

Row and Column spaces

let $A \in \mathbf{R}^{m\times n}$

row space	**column space**
$\mathbf{row}(A) = \{y \in \mathbf{R}^n \mid y = A^T x,\ x \in \mathbf{R}^m\}$	$\mathcal{R}(A) = \{y \in \mathbf{R}^m \mid y = Ax,\ x \in \mathbf{R}^n\}$
$z^T = u^T A$	$z = Au$
z^T is in $\mathbf{row}(A)$	z is in $\mathcal{R}(A)$
$Z = BA$	$Z = AB$
rows of Z are in $\mathbf{row}(A)$	columns of Z are in $\mathcal{R}(A)$

it's obvious from the definition that

$$\mathbf{row}(A) = \mathcal{R}(A^T)$$

Orthogonal projections

denote P the projections on the row or the column space of B

$\mathbf{row}(B)$		$\mathcal{R}(B)$	
$P(y^T)$	$= y^T B^T (BB^T)^{-1} B$	$P(y)$	$= B(B^T B)^{-1} B^T y$
B	$= \begin{bmatrix} L & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$	B	$= \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}$
$P(y^T)$	$= y^T Q_1 Q_1^T$	$P(y)$	$= Q_1 Q_1^T y$
$(I-P)(y^T)$	$= y^T Q_2 Q_2^T$	$(I-P)(y)$	$= Q_2 Q_2^T y$
A/B	$= AB^T (BB^T)^{-1} B$	A/B	$= B(B^T B)^{-1} B^T A$

- result for row space is obtained from column space by replacing B with B^T
- A/B is the projection of the $\mathbf{row}(A)$ onto $\mathbf{row}(B)$ (or projection of $\mathcal{R}(A)$ onto $\mathcal{R}(B)$)

Projection onto a row space

denote the projection matrices onto $\mathbf{row}(B)$ and $\mathbf{row}(B)^\perp$

$\mathbf{row}(B)$	$\mathbf{row}(B)^\perp$
$\Pi_B = B^T(BB^T)^{-1}B$	$\Pi_B^\perp = I - B^T(BB^T)^{-1}B$
$A/B = AB^T(BB^T)^{-1}B$	$A/B^\perp = A(I - B^T(BB^T)^{-1}B)$

get projections of $\mathbf{row}(A)$ onto $\mathbf{row}(B)$ or $\mathbf{row}(B)^\perp$ from LQ factorization

$$\begin{aligned}
\begin{bmatrix} B \\ A \end{bmatrix} &= \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = \begin{bmatrix} L_{11}Q_1^T \\ L_{21}Q_1^T + L_{22}Q_2^T \end{bmatrix} \\
A/B &= (L_{21}Q_1^T + L_{22}Q_2^T)Q_1Q_1^T = L_{21}Q_1^T \\
A/B^\perp &= (L_{21}Q_1^T + L_{22}Q_2^T)Q_2Q_2^T = L_{22}Q_2^T
\end{aligned}$$

Oblique projection

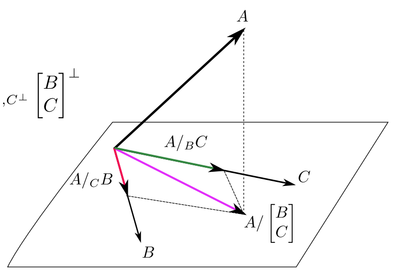
instead of an *orthogonal* decomposition $A = A\Pi_B + A\Pi_{B^\perp}$,
we represent $\mathbf{row}(A)$ as a linear combination of

the rows of two *non-orthogonal* matrices B and C and
of the orthogonal complement of B and C

$$A = L_B B + L_C C + L_{B^\perp, C^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp$$

$$L_C C \triangleq A/_B C$$

$$L_B B \triangleq A/_C B$$



$A/_B C$ is called the **oblique projection** of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$

the oblique projection can be interpreted as follows

1. project $\mathbf{row}(A)$ orthogonally into the *joint* row of B and C that is $A/\begin{bmatrix} B \\ C \end{bmatrix}$
2. decompose the result in part 1) along $\mathbf{row}(B)$, denoted as $L_B B$
3. decompose the result in part 1) along $\mathbf{row}(C)$, denoted as $L_C C$
4. the orthogonal complement of the result in part 1) is denoted as $L_{B^\perp, C^\perp} \begin{bmatrix} B \\ C \end{bmatrix}^\perp$

the **oblique projection** of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$ can be computed as

$$A/_B C = L_C C = L_{32} L_{22}^{-1} \begin{bmatrix} L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$$

where

$$\begin{bmatrix} B \\ C \\ A \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix}$$

the computation of the oblique projection can be derived as follows

- the projection of $\mathbf{row}(A)$ into the joint row space of B and C is

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{1}$$

- this can also written as linear combination of the rows of B and C

$$A/\begin{bmatrix} B \\ C \end{bmatrix} = L_B B + L_C C = \begin{bmatrix} L_B & L_C \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{2}$$

- equating (1) and (2) gives

$$\begin{bmatrix} L_B & L_C \end{bmatrix} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}^{-1} = \begin{bmatrix} L_{31} & L_{32} \end{bmatrix} \begin{bmatrix} L_{11}^{-1} & 0 \\ -L_{22}^{-1} L_{21} L_{11}^{-1} & L_{22}^{-1} \end{bmatrix}$$

the **oblique projection** of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$ is then

$$A/_B C = L_C C = L_{32} L_{22}^{-1} C = L_{32} L_{22}^{-1} (L_{21} Q_1^T + L_{22} Q_2^T) \tag{3}$$

(finished the proof)

useful properties: $B/_B C = 0$ and $C/_B C = C$

these can be viewed from constructing the matrices

$$\begin{bmatrix} B \\ C \\ B \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{11} & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_1^T \end{bmatrix}, \quad \begin{bmatrix} B \\ C \\ C \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{21} & L_{22} & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ 0 \end{bmatrix}$$

and apply the result of oblique projection in (3)

Equivalent form of oblique projection

the oblique projection of $\mathbf{row}(A)$ along $\mathbf{row}(B)$ into $\mathbf{row}(C)$ can also be defined as

$$A/_B C = A \begin{bmatrix} B^T & C^T \end{bmatrix} \left(\begin{bmatrix} BB^T & BC^T \\ CB^T & CC^T \end{bmatrix}^\dagger \right)_{\text{last } r \text{ columns}} \cdot C$$

where C has r rows

using the properties: $B/_B C = 0$ and $C/_B C = C$, we have

corollary: oblique projection can also be defined

$$A/_B C = (A/B^\perp) \cdot (C/B^\perp)^\dagger C$$

see detail in P.V. Overschee page 22

Subspace method

- main idea
- notation
- geometric tools
- **deterministic subspace identification**
- stochastic subspace identification
- combination of deterministic-stochastic identification
- MATLAB examples

Deterministic subspace identification

problem statement: estimate A, B, C, D in **noiseless** case from y, u

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

method outline:

1. calculate the state sequence (x)
2. compute the system matrices (A, B, C, D)

it is based on the input-output equation

$$\begin{aligned} Y_{0|i-1} &= \Gamma_i X_0 + H_i U_{0|i-1} \\ Y_{i|2i-1} &= \Gamma_i X_i + H_i U_{i|2i-1} \end{aligned}$$

Calculating the state sequence

derive future outputs

from state equations we have input/output equations

past: $Y_{0|i-1} = \Gamma_i X_0 + H_i U_{0|i-1}$, future: $Y_{i|2i-1} = \Gamma_i X_i + H_i U_{i|2i-1}$

from state equations, we can write X_i (future) as

$$\begin{aligned} X_i &= A^i X_0 + \Delta_i U_{0|i-1} = A^i(-\Gamma_i^\dagger H_i U_{0|i-1} + \Gamma_i^\dagger Y_{0|i-1}) + \Delta_i U_{0|i-1} \\ &= \begin{bmatrix} \Delta_i - A^i \Gamma_i^\dagger H_i & A^i \Gamma_i^\dagger \end{bmatrix} \begin{bmatrix} U_{0|i-1} \\ Y_{0|i-1} \end{bmatrix} \triangleq L_p W_p \end{aligned}$$

future states = in the row space of past inputs and past outputs

$$Y_{i|2i-1} = \Gamma_i L_p W_p + H_i U_{i|2i-1}$$

find oblique projection of future outputs: onto past data and along the future inputs

$$A/_B C = (A/B^{\perp})\cdot(C/B^{\perp})^{\dagger}C \quad \Longrightarrow \quad Y_f/_{U_f}W_p = (Y_{i|2i-1}/U^{\perp}_{i|2i-1})(W_p/U^{\perp}_{i|2i-1})^{\dagger}W_p$$

the oblique projection is defined as $\mathcal{O}_i$ and can be derived as

$$\begin{aligned} Y_{i|2i-1} &= \Gamma_i L_p W_p + H_i U_{i|2i-1} \\ Y_{i|2i-1}/U^{\perp}_{i|2i-1} &= \Gamma_i L_p W_p/U^{\perp}_{i|2i-1} + 0 \\ (Y_{i|2i-1}/U^{\perp}_{i|2i-1})(W_p/U^{\perp}_{i|2i-1})^{\dagger}W_p &= \Gamma_i L_p \underbrace{(W_p/U^{\perp}_{i|2i-1})(W_p/U^{\perp}_{i|2i-1})^{\dagger}W_p}_{W_p} \\ \mathcal{O}_i &= \Gamma_i L_p W_p = \Gamma_i X_i \end{aligned}$$

projection = extended observability matrix · future states

we have applied the result of $FF^{\dagger}W_p = W_p$ which is NOT obvious

see Overschee page 41 (up to some assumptions on excitation in u)

compute the states: from SVD factorization

since Γ_i has n columns and X_i has n rows, so $\mathbf{rank}(\mathcal{O}_i) = n$

$$\begin{aligned} \mathcal{O}_i &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T \\ &= U_1 \Sigma_n^{1/2} T \cdot T^{-1} \Sigma_n^{1/2} V_1^T, \quad \text{for some non-singular } T \end{aligned}$$

the extended observability is equal to

$$\Gamma_i = U_1 \Sigma_n^{1/2} T$$

the future states is equal to

$$X_i = \Gamma_i^{\dagger}\mathcal{O}_i = \Gamma_i^{\dagger} \cdot Y_{i|2i-1}/_{U_{i|2i-1}}W_p$$

future states = inverse of extended observability matrix · projection of future outputs

note that in Overschee use SVD of $W_1\mathcal{O}_i W_2$ for some weight matrices

Computing the system matrices

from the definition of $\mathcal{O}_i$, we can obtain

$$\mathcal{O}_{i-1} = \Gamma_{i-1} X_{i+1} \quad \Longrightarrow \quad X_{i+1} = \Gamma_{i-1}^{\dagger}\mathcal{O}_{i-1}$$

(X_i and X_{i+1} are calculated using only input-output data)

the system matrices can be solved from

$$\begin{bmatrix} X_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_i \\ U_{i|i} \end{bmatrix}$$

in a linear least-squares sense

- options to solve in a single or two steps (solve A, C first then B, D)
- for two-step approach, there are many options: using LS, total LS, stable A

Subspace method

- main idea
- notation
- geometric tools
- deterministic subspace identification
- **stochastic subspace identification**
- combination of deterministic-stochastic identification
- MATLAB examples

Stochastic subspace identification

problem statement: estimate A, C, Q, S, R from the system without input:

$$x(t+1) = Ax(t) + w(t), \quad y(t) = Cx(t) + v(t)$$

where Q, S, R are noise covariances (see page 12-2)

method outline:

1. calculate the state sequence (x) from input/output data
2. compute the system matrices (A, C, Q, S, R)

note that classical identification would use Kalman filter that requires *system matrices* to estimate the state sequence

Bank of non-steady state Kalman filter

if the system matrices *would be known*, $\hat{x}_{i+q}$ would be obtained as follows

$$\begin{array}{lcl} \hat{X}_0 & = & \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \\ P_0 & = & 0 \qquad\qquad\qquad\qquad\qquad \text{Kalman filter} \\ Y_p & & \begin{bmatrix} y_0 & \cdots & y_q & \cdots & y_{j-1} \\ \vdots & & \vdots & & \vdots \\ y_{i-1} & \cdots & y_{i+q-1} & \cdots & y_{i+j-2} \end{bmatrix} \quad \downarrow \\ \hat{X}_i & & \begin{bmatrix} \hat{x}_i & \cdots & \hat{x}_{i+q} & \cdots & \hat{x}_{i+j-1} \end{bmatrix} \end{array}$$

- start the filter at time q with the initial 0
- iterate the non-steady state Kalman filte over i time steps (vertical arrow down)
- note that to get $\hat{x}_{i+q}$ it uses only partial i outputs
- repeat for each of the j columns to obtain a *bank* of non-steady state KF

Calculation of a state sequence

project the future outputs: onto the past output space

$$\mathcal{O}_i \triangleq Y_{i|2i-1}/Y_{0|i-1} = Y_f/Y_p$$

it is shown in Overschee (THM 8, page 74) that

$$\mathcal{O}_i = \Gamma_i \hat{X}_i$$

(product of extended observability matrix and the vector of KF states)

define another projection and we then also obtain

$$\begin{aligned}\mathcal{O}_{i-1} &\triangleq Y_{i+1|2i-1}/Y_{0|i} = Y_f^-/Y_p^+ \\ &= \Gamma_{i-1}\hat{X}_{i+1}\end{aligned}$$

(proof on page 82 in Overschee)

compute the state: from SVD factorization

- the system order (n) is the rank of $\mathcal{O}_i$

$$\mathcal{O}_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T$$

- for some non-singular T, and from $\mathcal{O}_i = \Gamma_i \hat{X}_i$, we can obtain

$$\Gamma_i = U_1 \Sigma_n^{1/2} T, \quad \hat{X}_i = \Gamma_i^{\dagger} \mathcal{O}_i$$

- the shifted state $\hat{X}_{i+1}$ can be obtained as

$$\hat{X}_{i+1} = \Gamma_{i-1}^{\dagger} \mathcal{O}_{i-1} = (\underline{\Gamma_i})^{\dagger} \mathcal{O}_{i-1}$$

  where $\underline{\Gamma_i}$ denotes Γ_i without the last l rows

- $\hat{X}_i$ and $\hat{X}_{i+1}$ are obtained directly from output data (do not need to know system matrices)

Computing the system matrices

system matrices: once $\hat{X}_i$ and $\hat{X}_{i+1}$ are known, we form the equation

$$\underbrace{\begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix}}_{\text{known}} = \begin{bmatrix} A \\ C \end{bmatrix} \underbrace{\hat{X}_i}_{\text{known}} + \underbrace{\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}}_{\text{residual}}$$

- $Y_{i|i}$ is a block Hankel matrix with only one row of outputs
- the residuals (innovation) are uncorrelated with $\hat{X}_i$ (regressors) then solving this equation in the LS sense yields an asymptotically unbiased estimate:

$$\begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} = \begin{bmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \hat{X}_i^{\dagger}$$

noise covariances

- the estimated noise covariances are obtained from the residuals

$$\begin{bmatrix} \hat{Q}_i & \hat{S}_i \\ \hat{S}_i^T & \hat{R}_i \end{bmatrix} = (1/j) \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}^T$$

- the index i indicates that these are the *non-steady* state covariance of the non-steady state KF
- as $i \to \infty$, which is upon convergence of KF, we have convergence in Q, S, R

Subspace identification

- main idea
- notation
- geometric tools
- deterministic subspace identification
- stochastic subspace identification
- **combination of deterministic-stochastic identifications**
- MATLAB examples

Combined deterministic-stochastic identification

problem statement: estimate A, C, B, D, Q, S, R from the system:

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad y(t) = Cx(t) + Du(t) + v(t)$$

(system with **both** input and noise)

assumptions: (A, C) observable and see page 98 in Overschee

method outline:

1. calculate the state sequence (x) using oblique projection
2. compute the system matrices using least-squares

Calculating a state sequence

project future outputs: into the joint rows of past input/output along future inputs

define the two oblique projections

$$\mathcal{O}_i = Y_f /_{U_f} \begin{bmatrix} U_p \\ Y_p \end{bmatrix}, \quad \mathcal{O}_{i-1} = Y_f^- /_{U_f^-} \begin{bmatrix} U_p^+ \\ Y_p^+ \end{bmatrix}$$

important results: the oblique projections are the product of extended observability matrix and the KF sequences

$$\mathcal{O}_i = \Gamma_i \tilde{X}_i, \quad \mathcal{O}_{i-1} = \Gamma_{i-1} \tilde{X}_{i+1}$$

where $\tilde{X}_i$ is initialized by a particular $\hat{X}_0$ and run the same way as on page 12-24

(see detail and proof on page 108-109 in Overschee)

compute the state: from SVD factorization

- the system order (n) is the rank of $\mathcal{O}_i$

$$\mathcal{O}_i = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_n V_1^T$$

- for some non-singular T, and from $\mathcal{O}_i = \Gamma_i \hat{X}_i$, we can compute

$$\Gamma_i = U_1 \Sigma_n^{1/2} T, \quad \tilde{X}_i = \Gamma_i^\dagger \mathcal{O}_i$$

- the shifted state $\tilde{X}_{i+1}$ can be obtained as

$$\tilde{X}_{i+1} = \Gamma_{i-1}^\dagger \mathcal{O}_{i-1} = (\underline{\Gamma_i})^\dagger \mathcal{O}_{i-1}$$

  where $\underline{\Gamma_i}$ denotes Γ_i without the last l rows

- $\hat{X}_i$ (stochastic) and $\tilde{X}_i$ (combined) are different by the initial conditions

Computing the system matrices

system matrices: once $\tilde{X}_i$ and $\tilde{X}_{i+1}$ are known, we form the equation

$$\underbrace{\begin{bmatrix} \tilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix}}_{\text{known}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \underbrace{\begin{bmatrix} \tilde{X}_i \\ U_{i|i} \end{bmatrix}}_{\text{known}} + \underbrace{\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}}_{\text{residual}}$$

- solve for A, B, C, D in LS sense and the estimated covariances are

$$\begin{bmatrix} \hat{Q}_i & \hat{S}_i \\ \hat{S}_i^T & \hat{R}_i \end{bmatrix} = (1/j) \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}^T$$

(this approach is summarized in a combined algorithm 2 on page 124 of Overschee)

properties:

- $\tilde{X}_i$ and $\hat{X}_i$ are different by initial conditions but their difference goes to zero if either of the followings holds: (page 122 in Overschee)
  1. as $i \to \infty$
  2. the system if purely deterministic, $i.e.$, no noise in the state equation
  3. the deterministic input $u(t)$ is white noise
- the estimated system matrices are hence **biased** in many practical settings, $e.g.$, using steps, impulse input
- when at least one of the three conditions is satisfied, the estimate is asymptotically unbiased

Summary of combined identification

deterministic (no noise)	**stochastic (no input)**	**combined**
$\mathcal{O}_i = Y_f/_{U_f}\begin{bmatrix}U_p\\Y_p\end{bmatrix}$	$\mathcal{O}_i = Y_f/Y_p$	$\mathcal{O}_i = Y_f/_{U_f}\begin{bmatrix}U_p\\Y_p\end{bmatrix}$
$\mathcal{O}_i = \Gamma_i X_i$	$\mathcal{O}_i = \Gamma_i \hat{X}_i$	$\mathcal{O}_i = \Gamma_i \tilde{X}_i$
states are determined	state are estimated	state are estimated
	$\hat{X}_0 = 0$	$\tilde{X}_0 = X_0/_{U_f}U_p$

- without input, $\mathcal{O}_i$ is the projection of future outputs into past outputs
- with input, $\mathcal{O}_i$ should be explained jointly from past input/output data using the knowledge of inputs that will be presented to the system in the future
- with noise, the state estimates are initialized by the projection of the deterministic states

Complexity reduction

goal: to find as low-order model as possible that can predict the future

- reduce the complexity of the amount of information of the past that we need to keep track of to predict future
- thus we reduce the complexity of $\mathcal{O}_i$ (reduce the subspace dimension to n)

$$\underset{R}{\text{minimize}} \quad \|W_1(\mathcal{O}_i - \mathcal{R})W_2\|_F^2, \quad \text{subject to} \;\; \mathbf{rank}(\mathcal{R}) = n$$

  W_1, W_2 are chosen to determine which part of info in $\mathcal{O}_i$ is important to retain
- then the solution is
$$\mathcal{R} = W_1^{-1}U_1\Sigma_n V_1^T W_2^\dagger$$
  and in existing algorithms, $\mathcal{R}$ is used (instead of $\mathcal{O}_i$) to factorize for Γ_i

Algorithm variations

many algorithms in the literature start from SVD of $W_1\mathcal{O}_iW_2$

$$W_1\mathcal{O}_iW_2 = U_1\Sigma_n^{1/2}TT^{-1}\Sigma_n^{1/2}V_1^T$$

and can be arranged into two classes:

1. obtain the right factor of SVD as the state estimates $\tilde{X}_i$ to find the system matrices

2. obtain the left factor of SVD as Γ_i to determine A, C and B, D, Q, S, R subsequently

algorithms: `n4sid, CVA, MOESP` they all use different choices of W_1, W_2

Conclusions

- the subspace identification consists of two main steps:
  1. estimate the state sequence *without knowing the system matrices*
  2. determine the system matrices once the state estimates are obtained

- the state sequences are estimated based on the oblique projection of future input

- the projection can be shown to be related with the extended observability matrix and the state estimates, allowing us to retrieve the states via SVD factorization

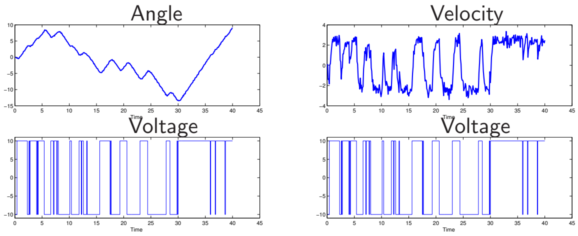
- once the states are estimated, the system matrices are obtained using LS

Example: DC motor

time response of the second-order DC motor system

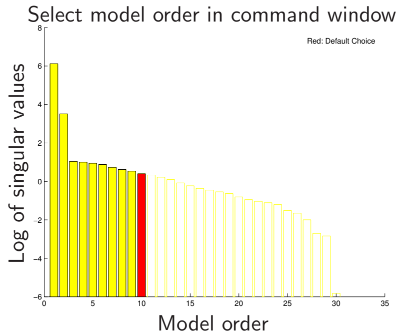
$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 1/\tau \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \beta/\tau \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ \gamma/\tau \end{bmatrix} T_l(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)\end{aligned}$$

where τ, β, γ are parameters to be estimated



use `n4sid` command in MATLAB

```
z = iddata(y,u,0.1);
m1 = n4sid(z,[1:10],'ssp','free','ts',0);
```



the software let the user choose the model order

select $n = 2$ and the result from free parametrization is

$$A = \begin{bmatrix} 0.010476 & -0.056076 \\ 0.76664 & -4.0871 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0015657 \\ -0.040694 \end{bmatrix}$$

$$C = \begin{bmatrix} 116.37 & 4.6234 \\ 4.766 & -24.799 \end{bmatrix}, \quad D = 0$$

the structure of A, B, C, D matrices can be specified

```
As = [0 1; 0 NaN]; Bs = [0; NaN];
Cs = [1 0; 0 1]; Ds = [0; 0];
Ks = [0 0;0 0]; X0s = [0;0];
```

where `NaN` is free parameter and we assign this structure to `ms` model

```
A = [0 1; 0 -1]; B = [0; 0.28];
C = eye(2); D = zeros(2,1);
ms = idss(A,B,C,D); % nominal model (or initial guess)
setstruc(ms,As,Bs,Cs,Ds,Ks,X0s);
set(ms,'Ts',0);  % Continuous model
```


the structured parametrization can be used with `pem` command

```
m2 = pem(z,ms,'display','on');
```

the estimate now has a desired structure

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -4.0131 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1.0023 \end{bmatrix}$$

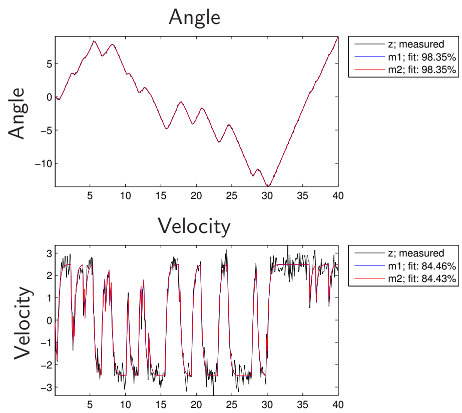
$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = 0$$

choosing model order is included in `pem` command as well

```
m3 = pem(z,'nx',1:5,'ssp','free');
```

`pem` use the `n4sid` estimate as an initial guess compare the fitting from the two models

```
compare(z,m1,m2);
```



References

Chapter 7 in
L. Ljung, *System Identification: Theory for the User*, 2nd edition, Prentice Hall, 1999

System Identification Toolbox demo
Building Structured and User-Defined Models Using System Identification Toolbox

P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems*, KLUWER Academic Publishers, 1996

K. De Cock and B. De Moor, *Subspace identification methods*, 2003

Chapter 13

Model selection and model validation

In model estimation process, a model structure and order are required from the user where a type of model structure is suitably chosen from a prior knowledge about the application of interest. For many dynamical systems, the model order can be approximated from some background of the application, *e.g.*, a DC motor with the motor angle as the output is of second-order. However, there are also some certain applications that the true model order is unknown, especially for time series models. A model class consists of many candidates of varied complexities. Therefore, when a model order is not known, one can consider varying the model complexity and estimate all the models in consideration, *e.g.*, use a state-space model of order n, and enumerate $n = 1, 2, \ldots, 10$. Intuitively, a model with a high complexity should capture the system dynamics better than a simple model. If we merely used a goodness of fit to select the best model then we would typically end up choosing a model of higher order. We will see later in practice, that as the model complexity varies, the goodness of fit is not always improved *significantly* and this is called the issue of *over-fitting.* This chapter considers another way to select a good model candidate that takes into account both the model fitting and model complexity, known as a criterion of *model selection*. Once we choose a model candidate, other considerations are needed to validate the model performance. For example, does the model contain a pole-zero cancellation? (indicating that the model order is still too high). Does the model capture sufficient dynamics from the input or the noise terms? This process is called a *model validation* and is performed last before we decide to use the chosen model.

Learning objectives of this chapter are

- to explain the bias-variance dilemma, and understand the causes of over-fitting problem,
- to apply typical model selection criterions which are AIC, BIC, and FPE,
- to perform typical model validation tests such as cross validation, and residual analysis.

13. Model Selection and Model Validation

- introduction
- model selection
- model validation

General aspects of the choice of model structure

1. type of model set
   - linear/nonlinear, state-spaces, black-box models
   - ARX, ARMAX, OE,...
2. size of the model set
   - degrees of polynomials $A(q^{-1}), B(q^{-1}), C(q^{-1})$, ...
   - dimension of state-space models
3. model parametrization

objective: obtain a good model at a low cost

1. **quality of the model:** defined by a measure of the goodness, e.g., the mean-squared error
   - MSE consists of a *bias* and a *variance* contribution
   - to reduce the bias, one has to use more flexible model structures (requiring more parameters)
   - the variance typically increases with the number of estimated parameters
   - the best model structure is therefore a trade-off between *flexibility* and *parsimony*
2. **price of the model:** an estimation method (which typically results in an optimization problem) highly depends on the model structures, which influences:
   - algorithm complexity
   - properties of the loss function
3. intended use of the model

Bias-Variance decomposition

assume that the observation Y obeys

$$Y = f(X) + \nu, \quad \mathbf{E}\nu = 0, \quad \mathbf{cov}(\nu) = \sigma^2$$

the mean-squared error of a regression fit $\hat{f}(X)$ at $X = x_0$ is

$$\begin{aligned} \text{MSE} &= \mathbf{E}[(Y - \hat{f}(x_0))^2 | X = x_0] \\ &= \sigma^2 + [\mathbf{E}\hat{f}(x_0) - f(x_0)]^2 + \mathbf{E}[\hat{f}(x_0) - \mathbf{E}\hat{f}(x_0)]^2 \\ &= \sigma^2 + \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)) \end{aligned}$$

- this relation is known as **bias-variance decomposition**
- no matter how well we estimate $f(x_0)$, σ^2 represents *irreducible error*
- typically, the more complex we make model $\hat{f}$, the lower the bias, but the higher the variance

Example

consider a stable first-order AR process

$$y(t) + ay(t-1) = \nu(t)$$

where $\nu(t)$ is white noise with zero mean and variance λ^2

consider the following two models:

$$\begin{aligned} \mathcal{M}_1 &: \; y(t) + a_1 y(t-1) = e(t) \\ \mathcal{M}_2 &: \; y(t) + c_1 y(t-1) + c_2 y(t-2) = e(t) \end{aligned}$$

let $\hat{a}_1, \hat{c}_1, \hat{c}_2$ be the LS estimates of each model

we can show that

$$\mathbf{var}(a_1) < \mathbf{var}(c_1)$$

(the simpler model has less variance)

apply a linear regression to the dynamical models

$$y(t) = H(t)\theta + \nu(t)$$

it asymptotically holds that

$$\mathbf{cov}(\hat{\theta}) = \lambda^2[\mathbf{E}H(t)^T H(t)]^{-1}$$

for model $\mathcal{M}_1$, we have $H(t) = -y(t-1)$, so

$$\mathbf{cov}(\hat{a}_1) = \lambda^2 / R_y(0)$$

for model $\mathcal{M}_2$, we have $H(t) = -\begin{bmatrix} y(t-1) & y(t-2) \end{bmatrix}$ and

$$\mathbf{cov}\begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \end{pmatrix} = \lambda^2 \begin{bmatrix} R_y(0) & R_y(1) \\ R_y(1) & R_y(0) \end{bmatrix}^{-1}$$

to compute $R_y(\tau)$, we use the relationship

$$R_y(\tau) = (-a)^\tau R_y(0),$$

where $R_y(0)$ is solved from a Riccati equation and the solution is

$$R_y(0) = \frac{\lambda^2}{1-a^2}$$

apply this result, we can show that

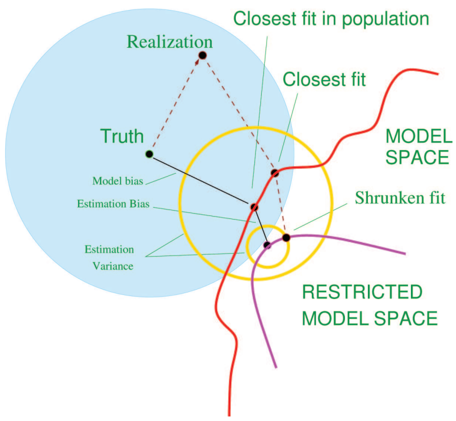
$$\mathbf{cov}(\hat{c}_1) = \frac{\lambda^2 R_y(0)}{R_y(0)^2 - R_y(1)^2} = \frac{\lambda^2}{R_y(0)(1-a^2)}$$

while

$$\mathbf{cov}(\hat{a}_1) = \frac{\lambda^2}{R_y(0)}$$

since $|a| < 1$, we can claim that $\mathbf{cov}(\hat{a}_1) < \mathbf{cov}(\hat{c}_1)$

Schematic of the behavior of bias and variance



(T. Hastie *et.al.* *The Elements of Statistical Learning*, Springer, 2010 page 225)

Model selection

- simple approach: enumerate a number of different models and to compare the resulting models
- what to compare ? how well the model is capable of reproducing these data
- how to compare ? comparing models on fresh data set: cross-validation
- model selection criterions
  - Akaike Information Criterion (AIC)
  - Baysian Information Criterion (BIC)
  - Minimum Description Length (MDL)

Overfitting

we will explain by an example of AR model with white noise ν

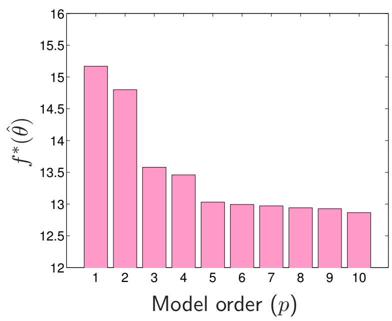
$$y(t) + a_1 y(t-1) + \dots a_p y(t-p) = \nu(t)$$

- true AR model has order $p = 5$
- the parameters to be estimated are $\theta = (a_1, a_2, \dots, a_p)$ with p unknown
- question: how to choose a proper value of p ?
- define a quadratic loss function

$$f(\theta) = \sum_{t=p+1}^{N} |y(t) - (a_1 y(t-1) + \dots + a_p y(t-p))|^2$$

and obtain $\hat{\theta}$ by using the LS method:

$$\hat{\theta} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_p) = \operatorname*{argmin}_{\theta} f(\theta)$$



- the minimized loss is a decreasing function of the model structure
- f begins to decreases as the model picks up the relevant features
- as p increases, the model tends to *over fit* the data
- in practice, we look for the "knee" in the curve (around $p = 5$)

Parsimony Principle

idea: among competing models which all explain the data well, the model with the smallest number of parameters should be chosen

In the previous example on page 13-11, how to determine model order p ?

- a trade-off curve between the loss function and the model order
- model selection criterions

a model selction criterion consists of two parts:

Loss function + Model complexity

- the first term is to assess the quality of the model, e.g., quadratic loss, likelihood function
- the second term is to penalize the model order and grows as the number of parameters increases

Examples of model selection criterions

Akaike Information Criterion (AIC)

$$\text{AIC} = -2\mathcal{L} + 2d$$

Bayesian Information Criterion (BIC)

$$\text{BIC} = -2\mathcal{L} + d\log N$$

Akaike's Final Prediction-Error Criterion (FPE)

$$\text{FPE} = \left(\frac{1}{N}\sum_{t=1}^{N} e^2(t,\theta)\right)\frac{1+d/N}{1-d/N}$$

- $\mathcal{L}$ is the loglikelihood function
- d is the number of effective parameters
- $e(t,\theta)$ is the prediction error

Some known properties:

- BIC tends to penalize complex models more heavily (due to the term $\log N$)
- BIC is asymptotically consistent

  (the probability that BIC will select the correct model approaches one as the sample size $N \to \infty$)
- AIC and FPE tends to choose models which are too complex as $N \to \infty$

AIC and BIC for Gaussian innovation

the ML method can be interpreted as PEM if the noise is *Gaussian*

in this case, the loglikelihood function (up to a constant) is

$$\mathcal{L} = \log L(\theta) = -\frac{N}{2}\log\det R(\theta)$$

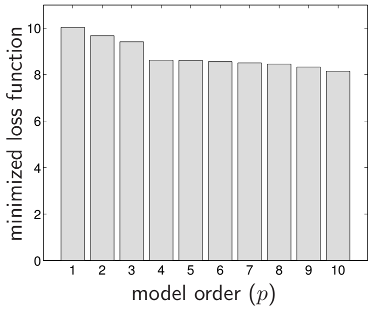
where $R(\theta) = \frac{1}{N}\sum_{t=1}^{N} e(t,\theta)e(t,\theta)^T$ is the sample covariance matrix

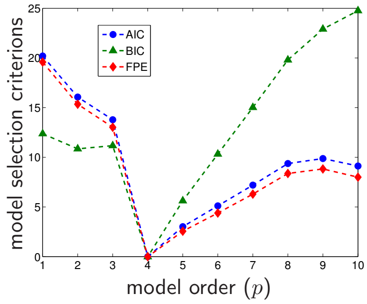
for scalar case, substituting $\mathcal{L}$ in AIC and BIC gives

$$\text{AIC} = -2\mathcal{L} + 2d = N\log\left(\frac{1}{N}\sum_{t=1}^{N} e^2(t,\theta)\right) + 2d$$

$$\text{BIC} = -2\mathcal{L} + d\log N = N\log\left(\frac{1}{N}\sum_{t=1}^{N} e^2(t,\theta)\right) + d\log N$$

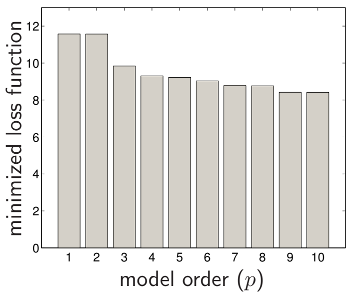
Example

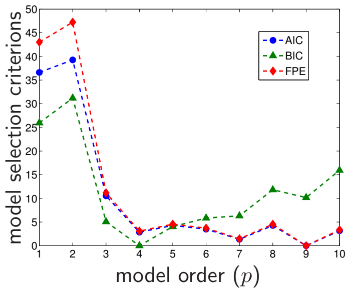




- the true system is AR model of order 4 with white noise of variance 1
- generate data of 100 points and estimate θ using LS

another realization





- AIC an FPE pick model of order 9 (too high)
- BIC still choose the correct model order (4)
- the estimates from AIC and FPE are not consistent
- BIC yields estimates that are consistent

Model validation

the parameter estimation procedure picks out the *best* model

a problem of model validation is to verify whether *this best* model is "good enough"

general aspects of model validation

- validation with respect to the purpose of the modeling
- feasibility of physical parameters
- consistency of model input-output behavior
- model reduction
- parameter confidence intervals
- simulation and prediction

Comparing model structures

use k-*step ahead model predictions* as a basis of the comparisons

$\hat{y}_k(t|m)$ denotes the k-step predictor based on model m and

$$u(t-1), \ldots u(1), y(t-k), \ldots, y(1)$$

for a linear model $y = \hat{G}u + \hat{H}\nu$, common choices are

- $\hat{y}_1(t|m)$ is the standard mean square optimal predictor

$$\hat{y}_1(t|m) = \hat{y}(t|t-1) = \hat{H}^{-1}(q^{-1})\hat{G}(q^{-1})u(t) + (1 - \hat{H}^{-1}(q^{-1}))y(t)$$

- $\hat{y}_\infty(t|m)$ is based on past inputs only (referred to as a pure *simulation*)

$$\hat{y}_\infty(t|m) = \hat{G}(q^{-1})u(t)$$

the models can be compared via a scalar measure of goodness:

$$J(m) = \frac{1}{N}\sum_{t=1}^{N} \|y(t) - \hat{y}_k(t|m)\|^2$$

the normalized measure, R, is given by detrending y and computing

$$R^2 = 1 - \frac{J(m)}{(1/N)\sum_{t=1}^{N}\|y(t)\|^2}$$

R represents part of the output variation that is explained by the model

- $J(m)$ depends on the realization of the data used in the comparison
- it is natural to consider the expected value of this measure:

$$\bar{J}(m) = \mathbf{E}J(m)$$

  which gives a quality measure for the given model

Cross validation

- a model structure that is "too rich" to describe the system will also partly model the disturbances that are present in the actual data set
- this is called an "overfit" of the data
- using a fresh dataset that was not included in the identification experiment for model validation is called "cross validation"
- cross validation is a nice and simple way to compare models and to detect "overfitted" models
- cross validation requires a large amount of data, the validation data cannot be used in the identification

K-fold cross-validation

- a simple and widely used method for estimating prediction error
- used when data are often scarce, then we split the data into K equal-sized parts
- for the kth part, we fit the model to the other $K-1$ parts of the data
- then compute $J(m)$ on the kth part of the data
- repeat this step for $k = 1, 2, \ldots, K$
- the cross-validation estimate of $J(m)$ is

$$\mathsf{CV}(m) = \frac{1}{K}\sum_{i=1}^{K} J_k(m)$$

- if $K = N$, it is known as *leave-one-out* cross-validation

Residual Analysis

the prediction error evaluated at $\hat{\theta}$ is called *the residuals*

$$e(t) = e(t, \hat{\theta}) = y(t) - \hat{y}(t; \hat{\theta})$$

- represents part of the data that the model could not reproduce
- if $\hat{\theta}$ is the true value, then $e(t)$ is white

a pragmatic view starting point is to use the basis statistics:

$$S_1 = \max_t |e(t)|, \quad S_2 = \frac{1}{N}\sum_{t=1}^{N} e^2(t)$$

to asses the quality of the model

the use of these statistics has an implicit invariance assumption

the residuals do not depend on the particular input

- the covariance between the residuals and past inputs

$$R_{eu}(\tau) = \frac{1}{N}\sum_{t=\tau}^{N} e(t)u(t-\tau)$$

  should be small if the model has picked up the essential part of the dynamics from u to y

- it also indicates that the residual is invariant to various inputs
- if

$$R_e(\tau) = \frac{1}{N}\sum_{t=\tau}^{N} e(t)e(t-\tau)$$

  is not small for $\tau \neq 0$, then part of $e(t)$ could have been predicted from past data

- this means $y(t)$ could have been better predicted

Whiteness test

if the model is accurately describing the observed data,

then the residuals $e(t)$ should be *white*, $i.e.$,

its covariance function $R_e(\tau)$ is zero except at $\tau = 0$

a way to validate the model is to test the hypotheses

$$\begin{aligned} H_0 &: \quad e(t) \text{ is a white sequence} \\ H_1 &: \quad e(t) \text{ is not a white sequence} \end{aligned}$$

this can be done via **autocorrelation test**

Autocorrelation test

the autocovariance of the residuals is estimated as:

$$\hat{R}_e(\tau) = \frac{1}{N}\sum_{t=\tau}^{N} e(t)e(t-\tau)$$

if H_0 holds, then the squared covariance estimate is asymptotically χ^2 distributed:

$$N\frac{\sum_{k=1}^{m}\hat{R}_e^2(k)}{\hat{R}_e^2(0)} \to \chi^2(m)$$

furthermore, the normalized autocovariance estimate is asymptotically Gaussian distributed

$$\sqrt{N}\frac{\hat{R}_e(\tau)}{\hat{R}_e(0)} \to \mathcal{N}(0,1)$$

a typical way of using the first test statistics for validation is as follows

let x denote a random variable which is χ^2-distributed with m degrees of freedom

furthermore, define $\chi^2_\alpha(m)$ by

$$\alpha = P(x > \chi^2_\alpha(m))$$

for some given α (typically between 0.01 and 0.1)

then if

$$N\frac{\sum_{k=1}^{m}\hat{R}_e^2(k)}{\hat{R}_e^2(0)} > \chi^2_\alpha(m) \quad \text{reject} \quad H_0$$

$$N\frac{\sum_{k=1}^{m}\hat{R}_e^2(k)}{\hat{R}_e^2(0)} < \chi^2_\alpha(m) \quad \text{accept} \quad H_0$$

(m is often chosen from 5 up to $N/4$)

Cross Correlation test

the input and the residuals should be uncorrelated (no unmodeled dynamics)

$$R_{eu}(\tau) = \mathbf{E}e(t)u(t-\tau) = 0$$

- if the model is not an accurate representation of the system, one can expect $R_{eu}(\tau)$ for $\tau \geq 0$ is far from zero
- indication of possible feedback in the input
- if $R_{eu}(\tau) \neq 0$ for $\tau < 0$ then there is output feedback in the input
- use the normalized test quantity

$$x_\tau = \frac{\hat{R}_{eu}(\tau)^2}{\hat{R}_e(\tau)\hat{R}_u(0)}$$

  for checking wheter the input and the residual are uncorrelated

for this purpose, introduce

$$\hat{R}_u = \frac{1}{N}\sum_{t=m+1}^{N} \begin{bmatrix} u(t-1) \\ \vdots \\ u(t-m) \end{bmatrix} \begin{bmatrix} u(t-1) & \cdots & u(t-m) \end{bmatrix}$$

$$r = \frac{1}{N}\sum_{t=1}^{N} \begin{bmatrix} u(t-\tau-1) \\ \vdots \\ u(t-\tau-m) \end{bmatrix} e(t)$$

where τ is a given integer and assume that $u(t) = 0$ for $t \leq 0$

then we have

$$Nr^T[\hat{R}_e(0)\hat{R}_u]^{-1}r \longrightarrow \chi^2(m)$$

which can be used to design a hypothesis test

Numerical examples

the system that we will identify is given by

$$(1 - 1.5q^{-1} + 0.7q^{-2})y(t) = (1.0q^{-1} + 0.5q^{-2})u(t) + (1 - 1.0q^{-1} + 0.2q^{-2})\nu(t)$$

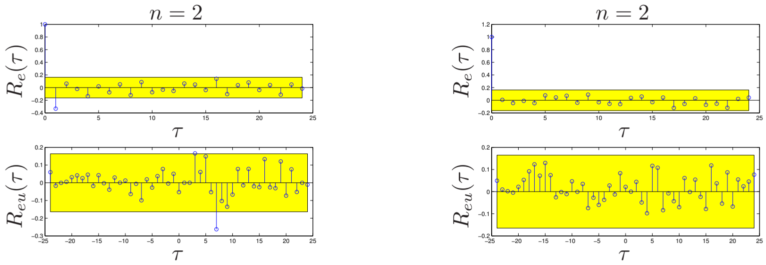
- $u(t)$ is binary white noise, independent of the white noise $\nu(t)$
- generate two sets of data, one for estimation and one for validation
- each data set contains data points of $N = 250$

Estimation:

- fitting ARX model of order n using the LS method
- fitting ARMAX model of order n using PEM

and vary n from 1 to 6

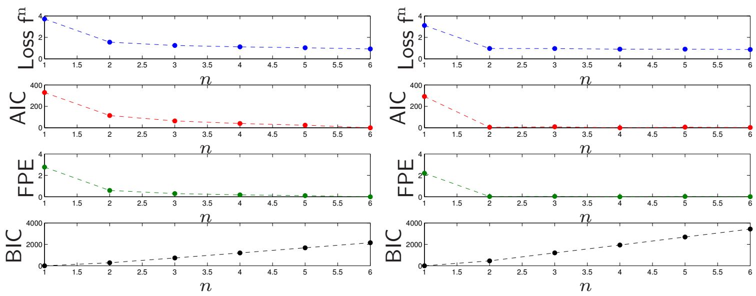
Example of residual analysis



(*Left.* LS method *Right.* PEM)

- the significant correlation of e shows that e cannot be seen as white noise, or the noise model H is not adequate
- the significant correlation between e and u shows the dynamics G is not adequate

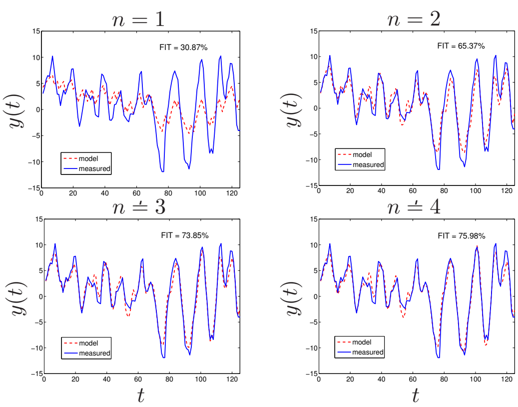
Example of model selection scores



(*Left.* LS method *Right.* PEM)

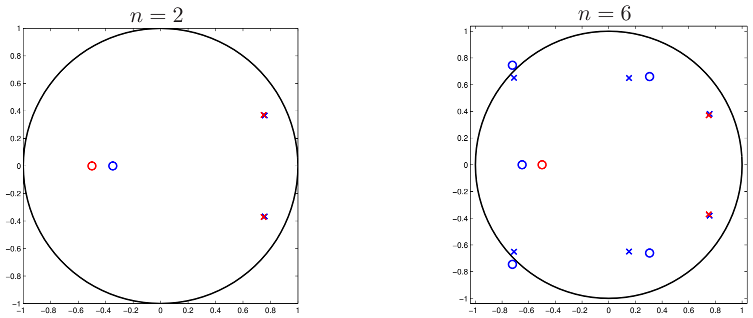
- AIC and FPE mostly pick higher models ($n = 4, 6$)
- BIC picks the simplest model
- all these scores decrease significantly at $n = 2$

Example of output prediction



(estimated by the LS method and validated on a new data set)

Example of zero-pole location



- estimated by PEM, ○: zeros, ×: poles
- red: true system, blue: estimated models
- chance of zero-pole cancellation at higher order

Example of MATLAB commands

let `theta` be an `idobject` obtained by using System Identification toolbox

- `armax`: estimate ARMAX models using PEM
- `iv4`: estimate ARX models using IVM
- `arx`: estimate ARX models using the LS method
- `resid`: residual analysis
- `compare`: compare the prediction with the measurement
- `zpplot`: plots of zeros and poles
- `theta.EstimationInfo.LossFcn`: value of loss function
- `theta.EstimationInfo.FPE`: value of FPE

References

Chapter 11 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 16 in
L. Ljung, *System Identification: Theory for the User*, 2nd edition, Prentice Hall, 1999

Lecture on
Model Structure Determination and Model Validation, System Identification (1TT875), Uppsala University,
`http://www.it.uu.se/edu/course/homepage/systemid/vt05`

Chapter 7 in T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second edition, Springer, 2009

Exercises

13.1 Bias and Variance of constrained LS estimate. Consider a linear model

$$y = Ax + \nu,$$

where $y \in \mathbf{R}^m$ is the measurement data, $x \in \mathbf{R}^n$ is the parameter to be estimated, $A \in \mathbf{R}^{m \times n}$ is the information matrix which is given, and ν_i is i.i.d. noise with zero mean and unit variance. In this problem, we investigate some properties of two estimates. Let $x_{\rm ls}$ be the least-squares estimate of x. Suppose, we have some *prior* information that the first component of x should be zero ($x_1 = 0$). Then we can let z be the least-squares estimate of x under a condition that $z_1 = 0$, *i.e.*,

$$z \;=\; \operatorname{argmin} \|Az - y\|_2 \quad \text{subject to} \quad z_1 = 0 \tag{13.1}$$

In what follows, we will explore statistical properties of $x_{\rm ls}$ and z.

(a) Find the closed-form expression of the mean and covariance of $x_{\rm ls}$.

(b) Explain how you can find the closed-form expression of z. *Hint.* Write Az as a linear combination of column vectors of A.

(c) Derive the closed-form expression of the mean and covariance of z.

(d) Are the two estimates unbiased ? Can you compare the biases of x and z ? Which one is smaller (in a norm sense) ? Does the result make sense to you ?

(e) Let $\mathbf{cov}(x_{\rm ls})$ and $\mathbf{cov}(z)$ be the covariance matrices of $x_{\rm ls}$ and z, respectively. Compare $\mathbf{cov}(x_{\rm ls})$ and $\mathbf{cov}(z)$. Which estimate has a smaller covariance ? Explain if the result you found in part (d) and (e) agree with the concept of bias and variance. Recall that we say $A \succeq B$ (in a matrix sense) if and only if $A - B$ is positive semidefinite. *Hint.* You will find the following result on the inverse of block matrix useful.

Suppose

$$X = \begin{bmatrix} A & B^T \\ B & D \end{bmatrix} \succ 0.$$

Then it can be shown that

$$\begin{bmatrix} A & B^T \\ B & D \end{bmatrix}^{-1} - \begin{bmatrix} 0 & 0 \\ 0 & D^{-1} \end{bmatrix} = \begin{bmatrix} I \\ -D^{-1}B \end{bmatrix} S^{-1} \begin{bmatrix} I & -B^T D^{-1} \end{bmatrix} \succeq 0,$$

where $S = A - B^T D^{-1} B$ is the Schur complement of A in X.

(f) *(3 points.)* From the closed-form expressions of mean and covariance of $x_{\rm ls}$ and z in part (a) and (c), we will verify if their empirical estimates are close to the true values. In the data file, `data-cmp-var-bias-sparseLS.mat`, we have provided A and 100 instances of y and the true value of x, in variables `A`, `y`, `x0`, respectively. To compute the empirical mean and covariance of $x_{\rm ls}$ and z, check out the `mean` and `cov` commands. Write down $\mathbf{E}[x_{\rm ls}], \mathbf{E}[z], \mathbf{cov}(x_{\rm ls})$, and $\mathbf{cov}(z)$ (theoretical values) and write down the empirical values of these four quantities. Discuss the results.

(g) *(3 points.)* Compare the biases of $x_{\rm ls}$ and z and compare $\mathbf{cov}(x_{\rm ls})$ and $\mathbf{cov}(z)$ (check if $\mathbf{cov}(x_{\rm ls}) - \mathbf{cov}(z) \succeq 0$.) Does the result agree with your argument in part (d) and (e) ?

13.2 Selecting the model order using BIC. Consider a scalar autoregressive model described by

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_p y(t-p) + \nu(t),$$

where $\nu(t)$ is i.i.d. noise with zero mean and variance σ^2. The measurements $y(1), y(2), \dots, y(N)$ are available in `data-modelsel-ar`. In this problem, we will estimate ten AR models with order $p = 1, 2, \dots, 10$, and we apply the Bayes information criterion (BIC) to select the best model. The BIC score is defined by

$$\text{BIC} = -2\mathcal{L} + d \log N$$

where $\mathcal{L}$ is the *log-likelihood* function of a model, d is the number of parameters in the model, and N is the number of data points used in the estimation.

(a) If the model order is given (p is known), derive a (conditional) maximum likelihood formulation for estimating $a_1, a_2, \dots, a_p$ and σ^2 (we assume $y(1), y(2), \dots, y(p)$ are known). Explain how you can compute the maximum likelihood estimate.

(b) Explain how you can compute each term in the BIC score. Vary p from 1 to 10 and complete the following table.

order	$\mathcal{L}$	$d \log N$	BIC
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

(c) Explain which term in the BIC score i) indicates the goodness of fit of the model ii) refers to the model complexity. Discuss how the goodness of fit and the complexity of the model change as p increases.

(d) Which model is selected by the BIC score ? Write down the estimate of $a_1, a_2, \dots, a_p$.

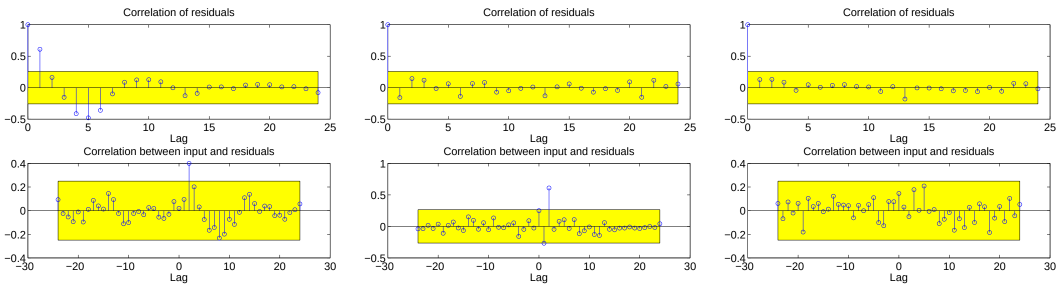
13.3 Residual test for ARX estimation. Consider a model of a mass-spring-damper system in discrete-time be given by

$$\begin{aligned} x(t) &= \begin{bmatrix} 1 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} x(t-1) + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u(t-1) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) + \nu(t), \end{aligned} \tag{13.2}$$

where $\nu(t)$ is measurement noise. However, we assume we do not know the true model of this system given in (13.2). In the experiment, the input signal $u(t)$ is a PRBS signal. By using the least-squares method, we estimate three ARX models which are the transfer functions from u to y:

$$G_1(q) = \frac{0.28q^{-2}}{1 - 1.33q^{-1} + 0.62q^{-2}}, \quad G_2(q) = \frac{0.074q^{-1}}{1 - 1.45q^{-1} + 0.71}, \quad G_3(q) = \frac{0.10}{1 - 0.89q^{-1}}.$$

We have validated these models on a new data set. For each model, the correlation function of residuals (top), and the cross correlation between input and residuals (bottom) are plotted in the figures below. Without a knowledge of the true system described in (13.2), can you match



(a) (b) (c)

each of the three estimated models to one of the three plots of residual test ? Which model is likely to agree with the result from the residual test in (a), (b), and (c) ? Justify your answer.

Chapter 14

Recursive identification

The procedures of model selection and validation described in chapter 13 provide us a model ready to be used for a particular purpose: controller design, forecasting, or others. All the estimation methods have one property in common that as the sample size of data in the model training process is increased, the estimation result is improved in many senses, for instance, the mean-square error is reduced or the estimator becomes consistent. In practice, acquiring a huge amount of data is not typical in some applications due to several factors: cost of operations or sensor problems, etc. Therefore, we generally obtain a model with a certain performance, provided all the data we have for training at the moment. However, we can collect more data as time progresses. Hence, it is generally interesting to cooperate new arrival measurement to improve the model estimation process in an adaptive manner. In control application, an adaptive estimation is also useful when the physical system is time-varying. For these reasons, this chapter describes basic recursive estimation methods where the principle underlying these is that the adaptive rule of parameter is derived from the offline expression of optimal estimator, or from the optimality condition, but with the consideration of new measurement is added to data set.

Learning objectives of this chapter are:

- to understand the effect of forgetting factors,
- to apply the recursive least-squares and recursive instrumental variable methods with forgetting factor,
- to explain the recursive prediction error method.

14. Recursive Identification Methods

- introduction
- recursive least-squares method
- recursive instrumental variable method
- recursive prediction error method

Introduction

features of recursive (online) identification

- $\hat{\theta}(t)$ is computed by some 'simple modification' of $\hat{\theta}(t-1)$
- used in central part of adaptive systems
- not all data are stored, so a small requirement on memory
- easily modified into real-time algorithms
- used in fault detection, to find out if the system has changed significantly

How to estimate time-varying parameters

- update the model regularly
- make use of previous calculations in an efficient manner
- the basic procedure is to modify the corresponding off-line method

Desirable properties of recursive algorithms

- fast convergence
- consistent estimates (time-invariant case)
- good tracking (time-varying case)
- computationally simple

Trade-offs

- convergence vs tracking
- computational complexity vs accuracy

Recursive least-squares method (RLS)

Recursive estimation of a constant: Consider the model

$$y(t) = b + \nu(t), \quad \nu(t) \text{ is a disturbance of variance } \lambda^2$$

the least-squares estimate of b is the arithmetic mean:

$$\hat{\theta}(t) = \frac{1}{t}\sum_{k=1}^{t} y(k)$$

this expression can be reformulated as

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{1}{t}[y(t) - \hat{\theta}(t-1)]$$

- the current estimate is equal to the previous estimate plus a correction
- the correction term is the deviation of the predicted value from what is actually observed

RLS algorithm for a general linear model

$$y(t) = H(t)\theta + \nu(t)$$

The recursive least-squares algorithm is given by

$$\begin{aligned} e(t) &= y(t) - H(t)\hat{\theta}(t-1) \\ P(t) &= P(t-1) - P(t-1)H^T(t)[I + H(t)P(t-1)H(t)^T]^{-1}H(t)P(t-1) \\ K(t) &= P(t)H(t)^T \quad = \quad P(t-1)H(t)^T[I + H(t)P(t-1)H(t)^T]^{-1} \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)e(t) \end{aligned}$$

- interprete $e(t)$ as a prediction error and $K(t)$ as a gain factor
- the update rule in $P(t)$ has an efficient matrix inversion for scalar case

Proof of the update formula the least-square estimate is given by

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

denote $P(t)$ as

$$P(t) = \left(\sum_{k=1}^{t} H(k)^T H(k)\right)^{-1} \quad \Longrightarrow \quad P^{-1}(t) = P^{-1}(t-1) + H(t)^T H(t)$$

then it follows that

$$\begin{aligned} \hat{\theta}(t) &= P(t)\left[\sum_{k=1}^{t-1} H(k)^T y(k) + H(t)^T y(t)\right] \\ &= P(t)\left[P^{-1}(t-1)\hat{\theta}(t-1) + H(t)^T y(t)\right] \end{aligned}$$

$$\begin{aligned}\hat{\theta}(t) &= P(t)\left[(P^{-1}(t) - H(t)^T H(t))\hat{\theta}(t-1) + H(t)^T y(t)\right] \\ &= \hat{\theta}(t-1) + P(t)H(t)^T\left[y(t) - H(t)\hat{\theta}(t-1)\right]\end{aligned}$$

to obtain the update rule for $P(t)$, we apply the matrix inversion lemma:

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

to

$$P^{-1}(t) = P^{-1}(t-1) + H(t)^T H(t)$$

where we use

$$A = P^{-1}(t-1), \quad B = H(t)^T, \quad C = I \quad D = H(t)$$

Initial conditions

- $\hat{\theta}(0)$ is the initial parameter estimate
- $P(0)$ is an estimate of the covariance matrix of the initial parameter
- if $P(0)$ is small then $K(t)$ will be small and $\hat{\theta}(t)$ will not change much
- if $P(0)$ is large, $\hat{\theta}(t)$ will quickly jump away from $\hat{\theta}(0)$
- it is common in practice to choose

  $$\hat{\theta}(0) = 0, \quad P(0) = \rho I$$

  where ρ is a constant
- using a large ρ is good if the initial estimate $\hat{\theta}(0)$ is uncertain

Effect of the initial values

we simulate the following system

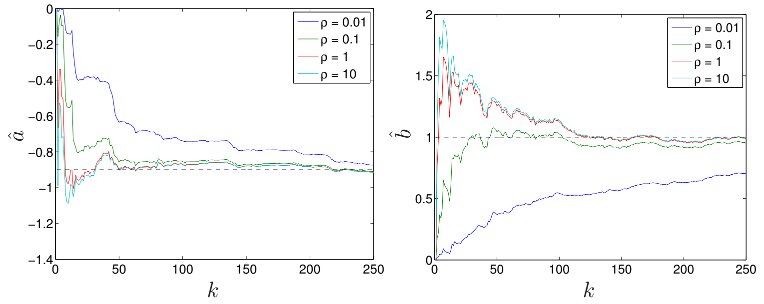
$$y(t) - 0.9y(t-1) = 1.0u(t-1) + \nu(t)$$

- $u(t)$ is binary white noise
- $\nu(t)$ is white noise of zero mean and variance 1
- identify the system using RLS with 250 points of data
- the parameters are initialized by

  $$\hat{\theta}(0) = 0, \quad P(0) = \rho\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

  for $\rho = 0.01, 0.1, 1, 10$

the graphs show the influence of the initial values



- large and moderate values of ρ ($i.e.$, $\rho = 1, 10$) lead to similar results
- for large ρ, little confidence is given to $\hat{\theta}(0)$, so quick transient response
- a small value of ρ leads to a small $K(t)$, so it gives a slower convergence

Forgetting factor

the loss function in the least-squares method is modified as

$$f(\theta) = \sum_{k=1}^{t} \lambda^{t-k} \|y(k) - H(k)\theta\|_2^2$$

- λ is called **the forgetting factor** and take values in $(0, 1)$
- the smaller the value of λ, the quicker the previous info will be forgotten
- the parameters are adapted to describe the newest data

Update rule for RLS with a forgetting factor

$$P(t) = \frac{1}{\lambda}\{P(t-1) - P(t-1)H(t)^T[\lambda I + H(t)P(t-1)H(t)^T]^{-1}H(t)P(t-1)\}$$
$$K(t) = P(t)H(t)^T = P(t-1)H(t)^T[\lambda I + H(t)P(t-1)H(t)^T]^{-1}$$
$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)[y(t) - H(t)\hat{\theta}(t-1)]$$

the solution $\hat{\theta}(t)$ that minimizes $f(\theta)$ is given by

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

the update formula follow analogously to RLS by introducing

$$P(t) = \left(\sum_{k=1}^{t} \lambda^{t-k} H(k)^T H(k)\right)^{-1}$$

the choice of λ is a trade-off between convergence and tracking performance

- λ small $\Longrightarrow$ old data is forgotten fast, hence good tracking
- λ close to 1 $\Longrightarrow$ good convergence and small variances of the estimates

Effect of the forgetting factor

consider the problem of tracking a time-varying system

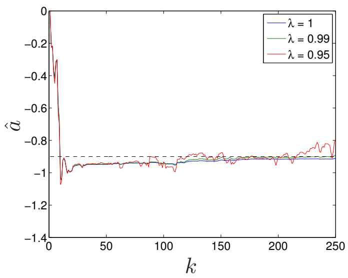
$$y(t) - 0.9y(t-1) = b_0 u(t) + \nu(t), \quad b_0 = \begin{cases} 1.5 & t \leq N/2 \\ 0.5 & t > N/2 \end{cases}$$

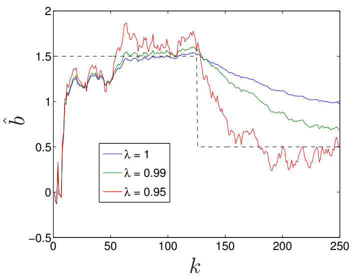
- $u(t)$ is binary white noise
- $\nu(t)$ is white noise of zero mean and variance 1
- identify the system using RLS with 250 points of data
- the parameters are initialized by

$$\hat{\theta}(0) = 0, \quad P(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- the forgetting factors are varied by these values $\lambda = 1, 0.99, 0.95$

graphs show the influence of the forgetting factors





a decrease in the forgetting factor leads to two effects:

- the estimates approach the true value more rapidly
- the algorithm becomes more sensitive to noise

as λ decreases, the oscillations become larger

summary:

- one must have $\lambda = 1$ to get convergence
- if $\lambda < 1$ the parameter estimate can change quickly, and the algorithm becomes more sensitive to noise

for this reason, it is often to allow the forgetting factor to vary with time

a typical choice is to let $\lambda(t)$ tends exponentially to 1

$$\lambda(t) = 1 - \lambda_0^t(1 - \lambda(0))$$

this can be easily implemented via a recursion

$$\lambda(t) = \lambda_0 \lambda(t-1) + (1 - \lambda_0)$$

typical values for $\lambda_0 = 0.99$ ($|\lambda_0|$ must be less than 1) and $\lambda(0) = 0.95$

Kalman Filter interpretation

consider a state-space of a time-varying system

$$\begin{aligned} x(t+1) &= A(t)x(t) + Bu(t) + \nu(t) \\ y(t) &= C(t)x(t) + \eta(t) \end{aligned}$$

where $\nu(t), \eta(t)$ are independent white noise with covariances R_1, R_2

Kalman filter:

$$\begin{aligned} \hat{x}(t+1) &= A(t)\hat{x}(t) + B(t)u(t) + K(t)[y(t) - C(t)\hat{x}(t)] \\ K(t) &= A(t)P(t)C(t)^T[C(t)P(t)C(t)^T + R_2]^{-1} \\ P(t+1) &= A(t)P(t)A(t)^T + R_1 \\ &\quad -A(t)P(t)C(t)^T[C(t)P(t)C(t)^T + R_2]^{-1}C(t)P(t)A(t)^T \end{aligned}$$

the linear regression model

$$y(t) = H(t)\theta + \nu(t)$$

can be written as a state-space equation

$$\begin{aligned} \theta(t+1) &= \theta(t) \quad (= \theta) \\ y(t) &= H(t)\theta(t) + \nu(t) \end{aligned}$$

apply the Kalman filter to the state-space equation with

$$A(t) = I, \quad B(t) = 0, \quad C(t) = H(t), \quad R_1 = 0$$

when $R_2 = I$, it will give precisely the basic RLS algorithm in page 14-5

the tracking capability is affected by R_2

Recursive instrument variable method

the IV estimate of a scalar linear system

$$y(t) = H(t)\theta + \nu(t)$$

is given by

$$\hat{\theta}(t) = \left[\sum_{k=1}^{t} Z(k)^T H(k)\right]^{-1} \left[\sum_{k=1}^{t} Z(k)^T y(k)\right]$$

the IV estimate can be computed recursively as

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)[y(t) - H(t)\hat{\theta}(t-1)] \\ K(t) &= P(t)Z(t)^T \quad = \quad P(t-1)Z(t)^T[I + H(t)P(t-1)Z(t)^T] \\ P(t) &= P(t-1) - P(t-1)Z(t)^T[I + H(t)P(t-1)Z(t)^T]^{-1}H(t)P(t-1) \end{aligned}$$

(analogous proof to RLS by using $P(t) = (\sum_{k=1}^{t} Z(k)^T H(k))^{-1}$)

Recursive prediction error method

we will use the cost function

$$f(t,\theta) = \frac{1}{2}\sum_{k=1}^{t} \lambda^{t-k} e(k,\theta)^T W e(k,\theta)$$

where $W \succ 0$ is a weighting matrix

- for $\lambda = 1$, $f(\theta) = \mathbf{tr}(WR(\theta))$ where $R(\theta) = \frac{1}{2}\sum_{k=1}^{t} e(k,\theta)e(k,\theta)^T$
- the off-line estimate of $\hat{\theta}$ cannot be found analytically (except for the LS case)
- it is *not* possible to derive an exact recursive algorithm
- some approximation must be used, and they hold exactly for the LS case

main idea: assume that

- $\hat{\theta}(t-1)$ minimizes $f(t-1,\theta)$
- the minimum point of $f(t,\theta)$ is close to $\hat{\theta}(t-1)$

using a second-order Taylor series approximation around $\hat{\theta}(t-1)$ gives

$$\begin{aligned} f(t,\theta) \approx f(t,\hat{\theta}(t-1)) &+ \nabla f(t,\hat{\theta}(t-1))^T(\theta - \hat{\theta}(t-1)) \\ &+ \frac{1}{2}[\theta - \hat{\theta}(t-1)]^T \nabla^2 f(t,\hat{\theta}(t-1))[\theta - \hat{\theta}(t-1)] \end{aligned}$$

minimize the RHS w.r.t. θ and let the minimizer be $\hat{\theta}(t)$:

$$\hat{\theta}(t) = \hat{\theta}(t-1) - [\nabla^2 f(t,\hat{\theta}(t-1))]^{-1} \nabla f(t,\hat{\theta}(t-1))$$

(Newton-Raphson step)

we must find $\nabla f(t,\hat{\theta}(t-1))$ and $P(t) = [\nabla^2 f(t,\hat{\theta}(t-1))]^{-1}$

details: to proceed, the gradients of $f(t,\theta)$ w.r.t θ are needed

$$\begin{aligned} f(t,\theta) &= \lambda f(t-1,\theta) + \frac{1}{2} e(t,\theta)^T W e(t,\theta) \\ \nabla f(t,\theta) &= \lambda \nabla f(t-1,\theta) + e(t,\theta)^T W \nabla e(t,\theta) \\ \nabla^2 f(t,\theta) &= \lambda \nabla^2 f(t-1,\theta) + \nabla e(t,\theta)^T W \nabla e(t,\theta) + e(t,\theta)^T W \nabla^2 e(t,\theta) \end{aligned}$$

first approximations:

- $\nabla f(t-1,\hat{\theta}(t-1)) = 0$ ($\hat{\theta}(t-1)$ minimizes $f(t-1,\theta)$)
- $\nabla^2 f(t-1,\hat{\theta}(t-1)) = \nabla^2 f(t-1,\hat{\theta}(t-2))$ ($\nabla^2 f$ varies slowly with θ)
- $e(t,\theta)^T W \nabla^2 e(t,\theta)$ is negligible

after inserting the above equations to

$$\hat{\theta}(t) = \hat{\theta}(t-1) - [\nabla^2 f(t,\hat{\theta}(t-1))]^{-1} \nabla f(t,\hat{\theta}(t-1))$$

we will have

$$\hat{\theta}(t) = \hat{\theta}(t-1) - [\nabla^2 f(t, \hat{\theta}(t-1)]^{-1}[e(t, \hat{\theta}(t-1))^T W \nabla e(t, \hat{\theta})(t-1)]$$
$$\nabla^2 f(t, \hat{\theta}(t-1)) = \lambda \nabla^2 f(t-1, \hat{\theta}(t-2)) + \nabla e(t, \hat{\theta}(t-1))^T W \nabla e(t, \hat{\theta}(t-1))$$

(still not suited well as an online algorithm due to the term $e(t, \hat{\theta}(t-1))$)

second approximations: let

$$e(t) \approx e(t, \hat{\theta}(t-1)), \quad H(t) \approx -\nabla e(t, \hat{\theta}(t-1))$$

(the actual way of computing these depends on model structures), then

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t) H^T(t) W e(t)$$

where we denote $P(t) = [\nabla^2 f(t, \hat{\theta}(t-1))]^{-1}$ which satisfies

$$P^{-1}(t) = \lambda P^{-1}(t-1) + H(t)^T W H(t)$$

apply the matrix inversion lemma to the recursive formula of $P^{-1}(t)$

we arrive at recursive prediction error method (RPEM)

algorithm:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t) e(t)$$
$$K(t) = P(t) H(t)^T$$
$$P(t) = \frac{1}{\lambda} \left\{ P(t-1) - P(t-1) H(t)^T [\lambda W^{-1} + H(t) P(t-1) H(t)^T]^{-1} P(t-1) \right\}$$

where the approximations

$$e(t) \approx e(t, \hat{\theta}(t-1)), \quad H(t) \approx -\nabla e(t, \hat{\theta}(t-1))$$

depend on the model structure

Example of RPEM: ARMAX models

consider the scalar ARMAX model

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

where all the polynomials have the same order

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \\ B(q^{-1}) &= b_1 q^{-1} + \cdots + b_n q^{-n} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \cdots + c_n q^{-n} \end{aligned}$$

define

$$\tilde{y}(t,\theta) = \frac{1}{C(q^{-1})}y(t), \quad \tilde{u}(t,\theta) = \frac{1}{C(q^{-1})}u(t), \quad \tilde{e}(t,\theta) = \frac{1}{C(q^{-1})}e(t)$$

we can derive the following relations

$$\begin{aligned} e(t,\theta) &= \frac{A(q^{-1})y(t) - B(q^{-1})u(t)}{C(q^{-1})} \\ \nabla e(t,\theta) &= (\tilde{y}(t-1,\theta),\ldots,\tilde{y}(t-n,\theta),-\tilde{u}(t-1,\theta),\ldots,-\tilde{u}(t-n,\theta), \\ &\qquad -\tilde{e}(t-1,\theta),\ldots,-\tilde{e}(t-n,\theta)) \end{aligned}$$

to compute $e(t,\theta)$, we need to process all data up to time t

we use the following approximations

$$\begin{aligned} e(t,\theta) \approx e(t) = y(t) &+ \hat{a}_1(t-1)y(t-1) + \cdots + \hat{a}_n(t-1)y(t-n) \\ &- \hat{b}_1(t-1)u(t-1) - \cdots - \hat{b}_n(t-1)u(t-n) \\ &- \hat{c}_1(t-1)e(t-1) - \cdots - \hat{c}_n(t-1)e(t-n) \end{aligned}$$

$$\begin{aligned} -\nabla e(t,\theta) \approx H(t) = (&-\bar{y}(t-1),\ldots,-\bar{y}(t-n), \\ &\bar{u}(t-1),\ldots,\bar{u}(t-n),\bar{e}(t-1),\ldots,\bar{e}(t-n)) \end{aligned}$$

where

$$\begin{aligned} \bar{y}(t) &= y(t) - \hat{c}_1(t)\bar{y}(t-1) - \cdots - \hat{c}_n(t)\bar{y}(t-n) \\ \bar{u}(t) &= u(t) - \hat{c}_1(t)\bar{u}(t-1) - \cdots - \hat{c}_n(t)\bar{u}(t-n) \\ \bar{e}(t) &= e(t) - \hat{c}_1(t)\bar{e}(t-1) - \cdots - \hat{c}_n(t)\bar{e}(t-n) \end{aligned}$$

Comparison of recursive algorithms

we simulate the following system

$$y(t) = \frac{1.0q^{-1}}{1-0.9q^{-1}}u(t) + \nu(t)$$

- $u(t), \nu(t)$ are indepentdent white noise with zero mean and variance 1
- we use RLS,RIV, RPEM to identify the system

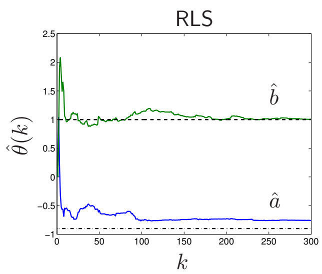
model structure for RLS and RIV:

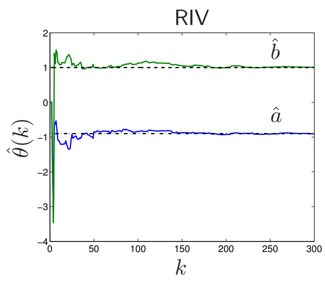
$$y(t) + ay(t-1) = bu(t-1) + \nu(t), \quad \theta = (a,b)$$

model structure for RPEM:

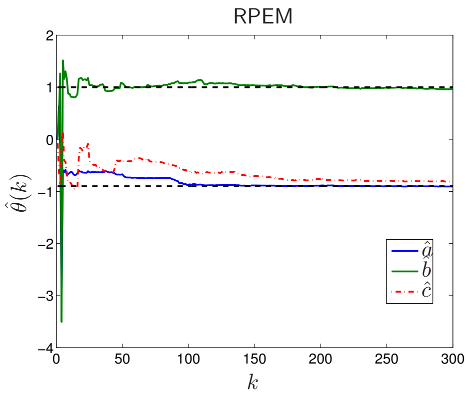
$$y(t) + ay(t-1) = bu(t-1) + \nu(t) + c\nu(t-1), \quad \theta = (a,b,c)$$

Numerical results





- RLS does not give consistent estimates for systems with correlated noise
- this is because RLS is equivalent to an off-line LS algorithm
- in contrast to RLS, RIV gives consistent estimates
- this result follows from that RIV is equivalent to an off-line IV method



- RPEM gives consistent estimates of a, b, c
- the estimates $\hat{a}$ and $\hat{b}$ converge more quickly than $\hat{c}$

Common problems for recursive identification

- excitation
- estimator windup
- $P(t)$ becomes indefinite

excitation it is important that the input is persistently excitation of sufficiently high order

Estimator windup

some periods of an identification experiment exhibit poor excitation

consider when $H(t) = 0$ in the RLS algorithm, then

$$\hat{\theta}(t) = \hat{\theta}(t-1), \quad P(t) = \frac{1}{\lambda}P(t-1)$$

- $\hat{\theta}$ becomes constant as t increases
- P increases exponentially with time for $\lambda < 1$
- when the system is excited again ($H(t) \neq 0$), the gain

$$K(t) = P(t)H(t)^T$$

will be very large and causes an abrupt change in $\hat{\theta}$

- this is referred to as *estimator windup*

Solution: do not update $P(t)$ if we have poor excitation

Indefinite $P(t)$

$P(t)$ represents a covariance matrix

therefore, it must be symmetric and positive definite

rounding error may accumulate and make $P(t)$ indefinite

this will make the estimate diverge

the solution is to note that any positive definite matrix can be factorized as

$$P(t) = S(t)S(t)^T$$

and rewrite the algorithm to update $S(t)$ instead

References

Chapter 9 in
T. Söderström and P. Stoica, *System Identification*, Prentice Hall, 1989

Chapter 11 in
L. Ljung, *System Identification: Theory for the User*, 2nd edition, Prentice Hall, 1999

Lecture on
Recursive Identification Methods, System Identification (1TT875), Uppsala University, http://www.it.uu.se/edu/course/homepage/systemid/vt05

Exercises

14.1 Update rule for recursive instrument variable method. The instrument variable estimate of a linear model

$$y(t) = H(t)\theta + \nu(t)$$

where $y(1), y(2), \ldots, y(t)$ are measured, is given by

$$\hat{\theta}(t) = \left[\sum_{k=1}^{t} Z(k)^T H(k)\right]^{-1} \left[\sum_{k=1}^{t} Z(k)^T y(k)\right]. \tag{14.1}$$

Define

$$P(t) = \left(\sum_{k=1}^{t} Z(k)^T H(k)\right)^{-1}.$$

Show that the recursive formula of the IV method is

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)[y(t) - H(t)\hat{\theta}(t-1)] \\
K(t) &= P(t)Z(t)^T \quad = \quad P(t-1)Z(t)^T[I + H(t)P(t-1)Z(t)^T] \\
P(t) &= P(t-1) - P(t-1)Z(t)^T[I + H(t)P(t-1)Z(t)^T]^{-1}H(t)P(t-1)
\end{aligned}$$

14.2 Recursive least-squares with a forgetting factor. Use a simple first-order scalar model

$$y(t) = ay(t-1) + bu(t-1) + \nu(t)$$

to describe the input/output data given in `data-rls-ff`. Determine a and b by using recursive least-squares update rule with two forgetting factors $\lambda = 1$ and $\lambda = 0.9$. Compare tracking performance of the estimates $\hat{a}$ and $\hat{b}$ between the two values of λ. The initial estimate is set to zero and the initial covariance matrix of the error is $P(0) = I$. Can you guess if there is a time-varying parameter in the model ? You must write your own MATLAB codes for recursive least-squares. Using the built-in command `rarx` in the system identification toolbox is not allowed. But you can verify the result with `rarx` command. Plot the convergence of $\theta(t)$ and attach your MATLAB codes in the work sheet. Provide the final values of the estimate $\hat{a}$ and $\hat{b}$.

Chapter 15

Applications of system identification

Since 2015, we have initiated a term project in this course where students can propose or choose a real-world problem that applies a technique of model estimation. In this chapter, we provide short descriptions of some term projects during 2015-2017 collected from student reports.

15.1 Rainfall Grid Interpolation from Rain Gauge and Rainfall Predicted from Satellite Data

Contributors of this work:

- Petchakrit Pinyopawasutthi
- Pongsorn Keadtipod
- Tanut Aranchayanont
- Piyatida Hoisungwan (co-advisor from Dept. of Water Resource)

Rain fall is conventionally collected by a rain gauge on stations which is accurate but scarcely available in spatial domain. In order to improve an interpolation of rainfall between stations, rainfall data predicted from satellite is introduced. Both data set are merged by a linear estimator to interpolate a rainfall map and the matrix coefficients of each term are chosen in the least-squares sense with constraints from prior structures of those matrices.

15.2 Parameter estimation of Gumbel distribution for flood peak data

Contributors of this work

- Jitin Khemwong
- Tiwat Boonyawiwat
- Piyatida Hoisungwan (co-advisor from Dept. of Water Resource)

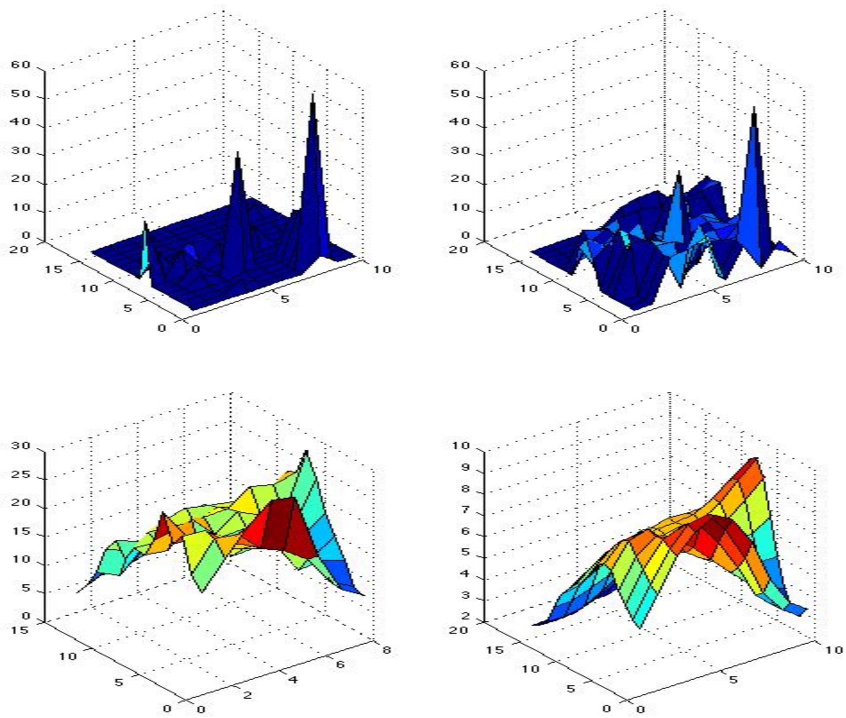


Figure 15.1: Rainfall from ground station (top) and rainfall from satellite station (bottom), displayed in form of grid matrix.

This project focuses on fitting Gumbel distribution to flood peak data of Chao Phraya river which depends on four rivers Ping, Wang, Yom, and Nan. First, the marginal probabilities of the river is obtained by fitting the old flood peak data with the Gumbel distribution where the parameters are estimated by using Maximum likelihood and Method of moments technique. Second, the relationship between the Chao Phraya river and others is investigated by considering the return period of bivariate Gumbel distribution. Third, multivariate Gumble distribution is being considered since it can describe the joint probability density function of flood peaks of all five rivers. Although the multivariate Gumbel distribution is expected to provide better information about the rivers, the formulation is too complicated, so this we focus solely on univariate and bivariate Gumbel distribution.

15.3 Solar Forecasting using Time Series Models

Contributors of this work

- Maxime Facquet
- Supachai Suksamosorn
- Veenakorn Suphatsatienkul
- Vichaya Layanun

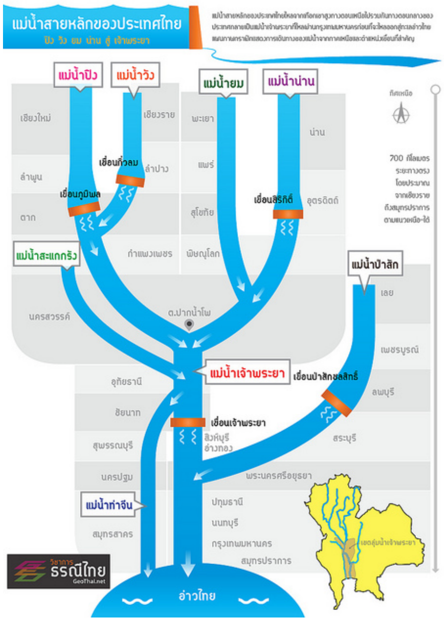


Figure 15.2: Thailand river map. Courtesy of `http://geothai.net`

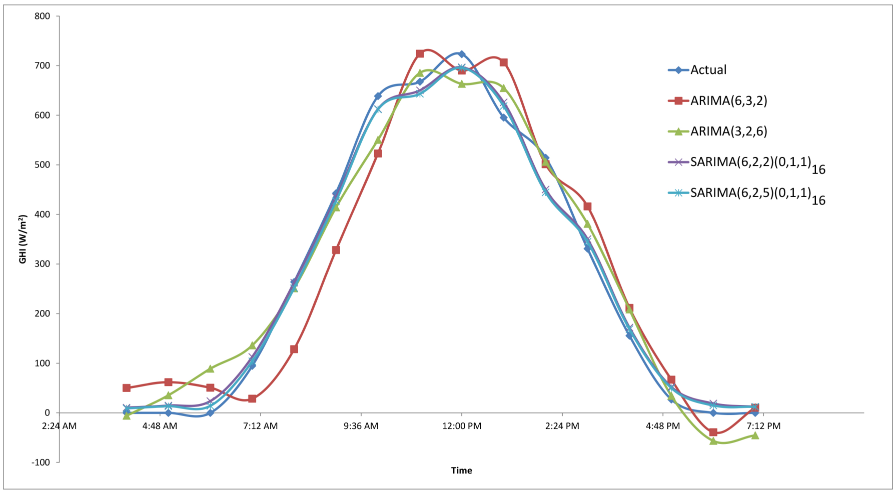


Figure 15.3: Comparison of an hour ahead solar irradiance prediction using different time series models

The improvement of technologies in renewable energies is crucial to its proper development. The accuracy of solar forecasting allows a higher efficiency in for solar grids. In this study we forecast global solar irradiance in Bangkok with data of the past 4 years. A modern time series analysis is used, more specifically an auto regressive moving average model including differential term (ARIMA). We conduct experiments to consider several models with different orders. Different model selection criteria are used in order to choose what seems to be the best fitting models, such as AIC, ACF behavior and prediction error. We found that a seasonal component in the model has to be considered. The best model will be a seasonal ARIMA$(2,2,2)(0,1,1)_{16}$ which will be in our view point the best trade of among AIC, simplicity and prediction error.

15.4 An Identification of Building Temperature System

Contributors of this work

- Chanthawit Anuntasethakul
- Natthapol Techaphanngam
- Natdanai Sontrapornpol

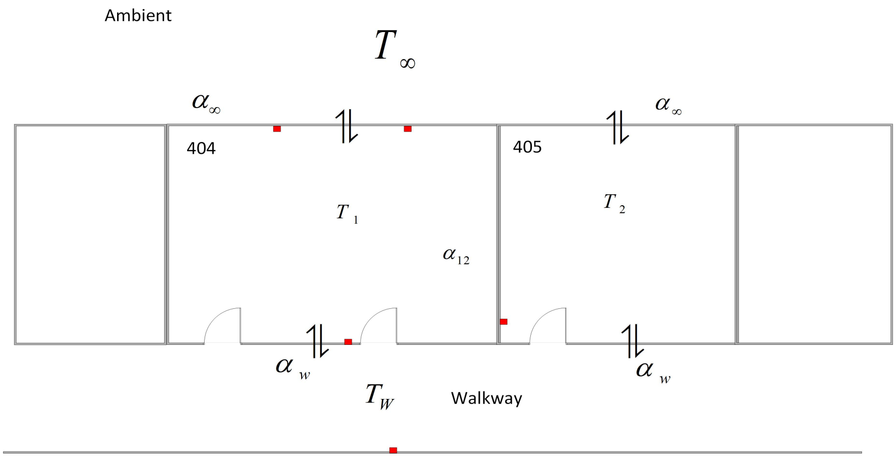


Figure 15.4: Building temperature system.

In this project, we aim to estimate the system matrices of a state-space model of a building temperature system via system identification. Only one-dimensional heat transfer and change in an internal energy have been concerned while neglecting effects caused by humidity, solar irradiance, and air leakage. Temperature data and air-conditioners input energy data can be obtained via the Chulalongkorn University Building Energy Management System (CUBEMS). Since the temperature data and energy input data can be measured, we choose a least-squares estimation. The results show that the dynamic matrix obtained via least-squares method is stable. The input matrix is forced to have a same structure as a state-space equation we have derived. In addition, we validate our model with the new data set. The results show that the system matrices we estimated still provide a good fitting performance calibrated by using mean-squares errors.

15.5 Modeling of Photovoltaic System

Contributor of this work: Janenarong Klomklao

Power forecasting of photovoltaic (PV) system using weather data is an important factor for planning the maintenance operations. This project presents nonlinear power prediction model based on a single-diode model with series resistance. The model required irradiance and cell temperature as inputs in order to identify model parameters. The method used to estimate the model parameters is nonlinear least square method with constraints. The initial guess for this optimization problem was obtained from analysis of derived model equation and specification values provided by manufacturers documentation. The proposed model were compared to two polynomial models and an artificial neural network (ANN) model in terms of mean squared error (MSE). The results indicated that the nonlinear model provided the least MSE compared to other models.

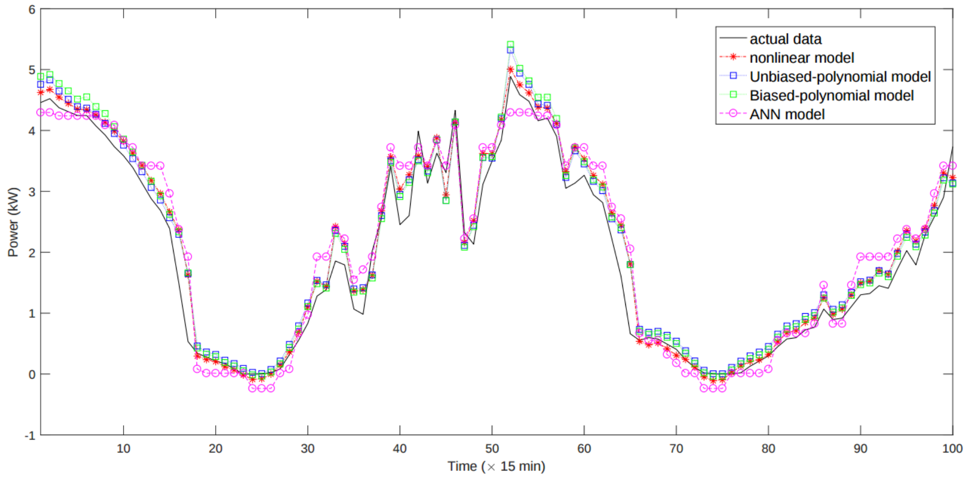


Figure 15.5: Fitting results of converted solar powers using PV conversion models.

Bibliography

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. Available: `www.stanford.edu/~boyd/cvxbook`.

[2] P.S.P. Cowpertwait and A.V. Metcalfe. *Introductory time series with R*. Springer Science & Business Media, 2009.

[3] W.H. Greene. *Econometric Analysis*. Pearson, 2000.

[4] J.D. Hamilton. *Time series analysis*. Princeton Univ Press, 1994.

[5] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2nd edition, 2009.

[6] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge university press, 2nd edition, 2013.

[7] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear estimation*. Prentice Hall, New Jersey, 2000.

[8] T. Katayama. *Subspace methods for system identification*. Springer Science & Business Media, 2006.

[9] L. Ljung. *System identification: Theory for the User*. Springer, 1998.

[10] J.P. Norton. *An Introduction to Identification*. Dover, 2009.

[11] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: Theory–Implementation–Applications*. Springer Science & Business Media, 2012.

[12] R.H. Shumway and D.S. Stoffer. *Time series analysis and its applications: with R examples*. Springer Science & Business Media, 2010.

[13] T. Söderström and P. Stoica. *System Identification*. Prentice Hall International, London, 1989.

[14] P. C. Young. *Recursive estimation and time-series analysis: an introduction*. Springer Science & Business Media, 2012.