System Identification

Control Engineering EN, 3rd year B.Sc.
Technical University of Cluj-Napoca
Romania

Lecturer: Lucian Buşoniu

Part III

Mathematical Background: Linear Regression and Statistics

Motivation

So far, we have been dealing with transient analysis of step-response models. This mostly involved familiar concepts related to linear systems and their time-domain responses.

Many upcoming methods for system identification require additional tools: linear regression and some concepts from probability theory and statistics. We will discuss these tools here.

In this part we redefine some notation (e.g. x, A) to have a different meaning than in the rest of the course.

Table of contents

Table of contents

Regression problem

Problem elements:

- A collection of known samples $y(k) \in \mathbb{R}$, indexed by $k = 1, \ldots, N$: y is the *regressed variable*.
- For each k, a known vector $\varphi(k) \in \mathbb{R}^n$: contains the *regressors* $\varphi_i(k)$, $i = 1, \ldots, n$, $\varphi(k) = [\varphi_1(k), \varphi_2(k) \ldots, \varphi_n(k)]^\top$.
- An unknown *parameter vector* $\theta \in \mathbb{R}^n$.

Objective: identify the behavior of the regressed variable from the data, using the linear model:

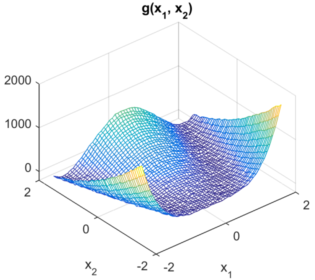
$$y(k) = \varphi^\top(k)\theta$$

Linear regression is classical and very common, e.g. it was used by Gauss to compute the orbits of the planets in 1809.

Regression problem: Two major uses

1. k is a time variable, and we wish to model the time series $y(k)$.
2. k is just a data index, and $\varphi(k) = \phi(x(k))$ where x is an input of some unknown function g. Then $y(k)$ is the corresponding output (possibly corrupted by noise), and the goal is to identify a model of g from the data.
   This problem is also called *function fitting*, *function approximation*, or *supervised learning*.

unknown $g(x)$



$\widehat{g}(x)$

?

Function approximation: Basis functions

For function approximation, the regressors $\phi_i(x)$ in:

$$\phi(x(k)) = [\phi_1(x(k)), \phi_2(x(k)), \dots, \phi_n(x(k))]^\top$$

are also called *basis functions*.

Function approximation: Motivating example 1

We study the yearly income y (in EUR) of a person based on their education level x_1 and job experience x_2 (both measured in years).

We are given a set of tuples $(x_1(k), x_2(k), y(k))$ from a representative set of persons. The goal is to predict the income of any other person by knowing how educated (x_1) and experienced (x_2) they are.

- Take basis functions $\phi(x) = [x_1, x_2, 1]^\top$. So we expect the income to behave like $\theta_1 x_1 + \theta_2 x_2 + \theta_3 = \phi^\top(x)\theta$, growing linearly with education and experience (from some minimum level). Regression involves finding the parameters θ in order to best fit the given data.
- Reality is of course more complicated... so we would likely need more input variables, better basis functions, etc.

Function approximation: Motivating example 2

We study the reaction time y (in ms) of a driver based on their age x_1 (in years) and fatigue x_2 (e.g. on a scale from 0 to 1).

We are given a set of tuples $(x_1(k), x_2(k), y(k))$ from a representative set of persons of various ages and stages of fatigue. The goal is to predict the rection time of any other person by knowing how old (x_1) and tired (x_2) they are.

Regressors example 1: Polynomial of k

Suitable for time series modeling.

$$
\begin{aligned}
y(k) &= \theta_1 + \theta_2 k + \theta_3 k^2 + \ldots + \theta_n k^{n-1} \\
&= \begin{bmatrix} 1 & k & k^2 & \ldots & k^{n-1} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \ldots \\ \theta_n \end{bmatrix} \\
&= \varphi^\top(k)\theta
\end{aligned}
$$

Regressors example 2: Polynomial of x

Suitable for function approximation. For instance, polynomial of degree 2 with two input variables $x = [x_1, x_2]^\top$:

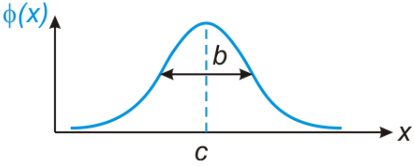
$$
\begin{aligned}
y(k) &= \theta_1 + \theta_2 x_1(k) + \theta_3 x_2(k) + \theta_4 x_1^2(k) + \theta_5 x_2^2(k) + \theta_6 x_1(k) x_2(k) \\
&= \begin{bmatrix} 1 & x_1(k) & x_2(k) & x_1^2(k) & x_2^2(k) & x_1(k)x_2(k) \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} \\
&= \phi^\top(x(k))\theta = \varphi^\top(k)\theta
\end{aligned}
$$

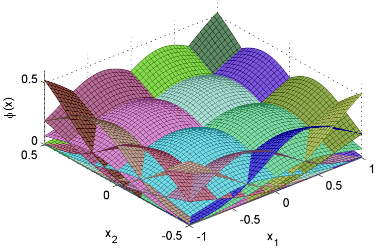
Connection: Project part 1

Regressors example 3: Gaussian basis functions

Suitable for function approximation:

$$\phi_i(x) = \exp\left[-\frac{(x-c_i)^2}{b_i^2}\right] \qquad \text{(1-dim);}$$
$$= \exp\left[-\sum_{j=1}^{d}\frac{(x_j-c_{ij})^2}{b_{ij}^2}\right] \qquad \text{(d-dim)}$$

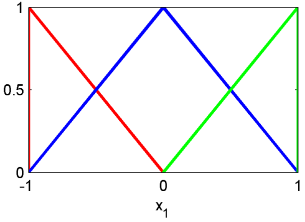


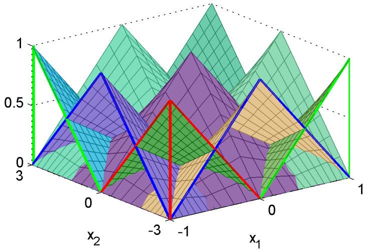


Regressors example 4: Interpolation

Suitable for function approximation.

- *d*-dimensional grid of points in the input space.
- (Multi)-Linear interpolation between the points.
- Equivalent with *pyramidal* basis functions (triangular in 1-dim)





Linear system

Writing the model for each of the N data points, we get a linear system of equations:

$$
\begin{aligned}
y(1) &= \varphi_1(1)\theta_1 + \varphi_2(1)\theta_2 + \dots \varphi_n(1)\theta_n \\
y(2) &= \varphi_1(2)\theta_1 + \varphi_2(2)\theta_2 + \dots \varphi_n(2)\theta_n \\
&\dots \\
y(N) &= \varphi_1(N)\theta_1 + \varphi_2(N)\theta_2 + \dots \varphi_n(N)\theta_n
\end{aligned}
$$

Recall that in function approximation, $\varphi_i(k) = \phi_i(x(k))$

This system can be written in a *matrix form*:

$$
\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \varphi_1(1) & \varphi_2(1) & \dots & \varphi_n(1) \\ \varphi_1(2) & \varphi_2(2) & \dots & \varphi_n(2) \\ \dots & \dots & \dots & \dots \\ \varphi_1(N) & \varphi_2(N) & \dots & \varphi_n(N) \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \dots \\ \theta_n \end{bmatrix}
$$

$$
\boxed{Y = \Phi\theta}
$$

with newly introduced variables $Y \in \mathbb{R}^N$ and $\Phi \in \mathbb{R}^{N \times n}$.

Least-squares problem

If $N = n$, the system can be solved with equality.

In practice, it is a good idea to use $N > n$, due e.g. to noise. In this case, the system can no longer be solved with equality, but only in an approximate sense.

- *Error* at k: $\varepsilon(k) = y(k) - \varphi^\top(k)\theta$,
  error vector $\varepsilon = [\varepsilon(1), \varepsilon(2), \dots, \varepsilon(N)]^\top$.
- Objective function to be minimized:

$$V(\theta) = \boxed{\frac{1}{2}\sum_{k=1}^{N} \varepsilon(k)^2} = \frac{1}{2}\varepsilon^\top\varepsilon$$

Least-squares problem

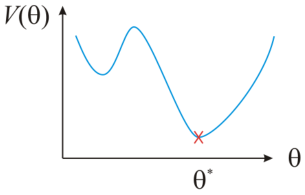
Find the parameter vector $\widehat{\theta}$ that minimizes the objective function:

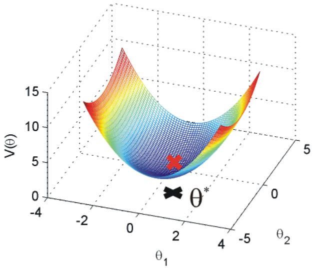
$$\widehat{\theta} = \arg\min_{\theta} V(\theta)$$

Parenthesis: Optimization problem

Given a function V of variables θ, which may be the least-squares objective, or any other function:

find the *optimal function value* $\min_\theta V(\theta)$ and variable values $\theta^* = \arg\min_\theta V(\theta)$ that achieve the minimum.





Note that in the case of linear regression, we use the notation $\widehat{\theta}$; while $\widehat{\theta}$ is still the true solution to the optimization problem given the data, it is still an estimate because the data is noisy

Formal regression solution

After applying some linear algebra:

$$\boxed{\widehat{\theta} = (\Phi^\top \Phi)^{-1} \Phi^\top Y}$$

Remarks:

- The optimal objective value is $V(\widehat{\theta}) = \frac{1}{2}[Y^\top Y - Y^\top \Phi (\Phi^\top \Phi)^{-1} \Phi^\top Y]$.
- Matrix $\Phi^\top \Phi$ must be invertible. This boils down to a good choice of the model (order n, regressors φ) and having informative data.

Alternative expression

$$\Phi^\top \Phi = \sum_{k=1}^{N} \varphi(k)\varphi^\top(k), \Phi^\top Y = \sum_{k=1}^{N} \varphi(k)y(k)$$

So the solution can be written:

$$\widehat{\theta} = \left[\sum_{k=1}^{N} \varphi(k)\varphi^\top(k)\right]^{-1} \left[\sum_{k=1}^{N} \varphi(k)y(k)\right]$$

Advantage: matrix Φ of size $N \times n$ no longer has to be computed, only smaller matrices and vectors are required, of size $n \times n$ and n, respectively.

Solving the linear system

In practice both these inversion-based techniques perform poorly from a numerical point of view. Better algorithms exist, such as so-called orthogonal triangularization.

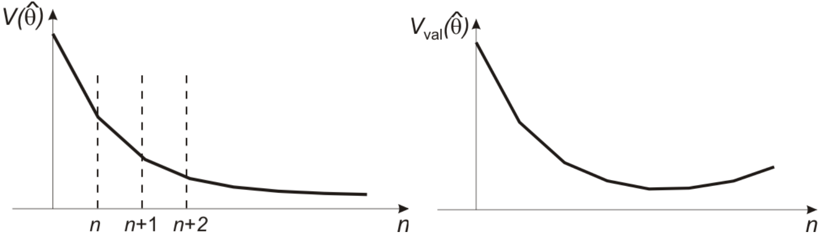
In most cases, MATLAB is competent in choosing a good algorithm. If Φ is stored in variable `PHI` and Y in `Y`, then the command to solve the linear system using matrix left division (backslash) is:

```
theta = PHI \ Y;
```

Better control of the algorithm is obtained by using function `linsolve` instead of matrix left division.

Model choice

Assume that given a model size n, we have a way to generate regressors $\varphi(k)$ that make the model more expressive (e.g., basis functions on a finer grid). Then we expect the objective function to behave as follows:



So we can grow n incrementally and stop when there are no significant improvements in V, or the error V_{val} on the validation data starts growing.

Remark: With noisy data, increasing n too much can lead to overfitting: good performance on the training data, but poor performance on other data. Validation on a separate dataset is essential in practice!

Table of contents

Analytical example: Estimating a scalar

Model:

$$y(k) = b = 1 \cdot b = \varphi(k)\theta$$

where $\varphi(k) = 1 \forall k$, $\theta = b$.

For all N data points:

$$y(1) = \varphi(1)\theta = 1 \cdot b$$
$$\dots$$
$$y(N) = \varphi(N)\theta = 1 \cdot b$$

In matrix form:

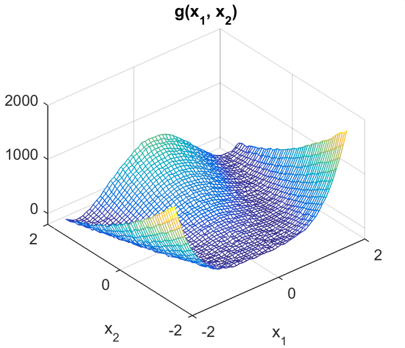
$$\begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \theta$$
$$Y = \Phi\theta$$

Analytical example: Estimating a scalar (continued)

$$
\begin{aligned}
\widehat{\theta} &= (\Phi^\top \Phi)^{-1} \Phi^\top Y \\
&= \left(\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \\
&= N^{-1} \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \\
&= \frac{1}{N} (y(1) + \ldots + y(N))
\end{aligned}
$$

Intuition: Estimate is the average of all measurements, filtering out the noise.

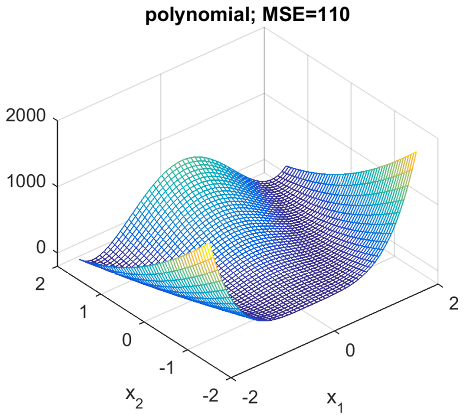
Example: Approximating the "banana" function



- Function $g(x_1, x_2) = (1 - x_1)^2 + 100[(x_2 + 1.5) - x_1^2]^2$, called Rosenbrock's banana function (unknown to the algorithm).
- Approximation data: 200 input points (x_1, x_2), randomly distributed over the space $[-2, 2] \times [-2, 2]$; and corresponding outputs $y = g(x_1, x_2)$, affected by noise.
- Validation data: a uniform grid of 31×31 points over $[-2, 2] \times [-2, 2]$ with their corresponding (noisy) outputs.

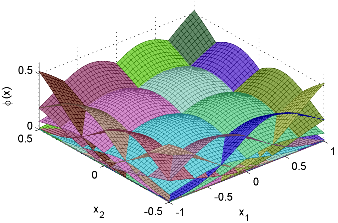
Banana function: Results with polynomial

Polynomial of degree 4 in the two input variables (15 parameters):

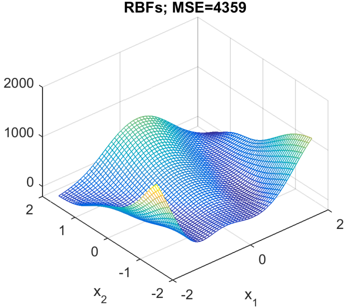


Banana function: Results with radial basis functions

Recall radial basis functions:

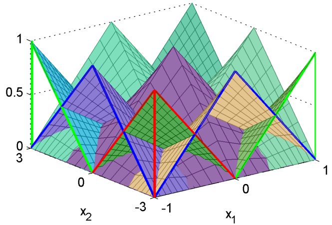


Results with 6 $\times$ 6 RBFs, with centers on an equidistant grid and width equal to the distance between two centers:



Banana function: Results with interpolation

Recall pyramidal basis functions from interpolation:



Results with a 6 $\times$ 6 interpolation grid (corresponding to 6 $\times$ 6 basis functions):

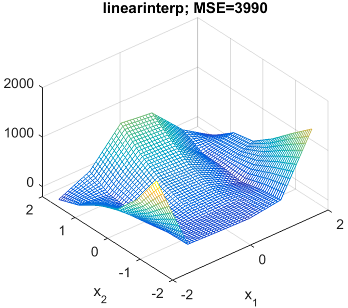


Table of contents

Discrete random variable

Consider a set $\mathcal{X}$ containing possible values x. A corresponding random variable X is described by the probability mass function:

Definition

The probability mass function (PMF) of X is the list of the probabilities of all individual values $p(x_0), p(x_1), \dots$. The probabilities must sum up to 1: $P(X \in \mathcal{X}) = \sum_{x \in \mathcal{X}} p(x) = 1$.

Example: The number rolled with a dice is a discrete random variable, with six possible values $\mathcal{X} = \{1, 2, ..., 6\}$. For a fair dice, the PMF is $p(x) = 1/6$ for all x, from 1 to 6.

Note that $\mathcal{X}$ may contain finitely or infinitely many elements, but in the latter case, the set must be countable (Intuition: elements can be listed / indexed by the natural numbers $0, 1, 2, \dots$).

Continuous random variable

Consider now an interval $\mathcal{X} \subseteq \mathbb{R}$ of real numbers, with each individual real value denoted x. A corresponding, continuous-valued random variable X is described by the probability density function:

Definition (semi-formal)

The probability density function (PDF) of X is a function $f : \mathcal{X} \to [0, \infty)$, such that the probability of obtaining a value in some subinterval $[a, b] \subseteq \mathcal{X}$ is:

$$P(X \in [a, b]) = \int_a^b f(x)dx$$

The PDF must satisfy $P(X \in \mathcal{X}) = \int_{x \in \mathcal{X}} f(x)dx = 1$.

Remark: $\mathcal{X}$ might also be the full set of real numbers.

Example 1: Uniform PDF

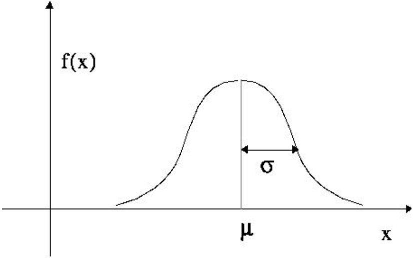
We wish to characterize the angle at which a roulette wheel ends up, as a fraction of a full turn: $x \in [0, 1]$. In this case, $\mathcal{X} = [0, 1]$. For a fair roulette, each value x has equal probability.

Try first to define a PMF; it should have the same value everywhere. Take any nonzero value c; since there are uncountably, infinitely many values in the interval $[0, 1]$, we have that $P(X \in [0, 1])$, which sums up c an infinite amount of times, is infinite. This means $p(x)$ must be 0 for all x! But this has no useful meaning – we cannot use a PMF to describe this case.

Meaningful probabilities can only be defined for intervals, and that is why we need a PDF f. In particular, to have uniform probabilities, we want $P(x \in [a, b]) = b - a$, which means $f(x) = 1$.

Example 2: Gaussian PDF

Similar in shape, but not in meaning, with Gaussian basis functions.



$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Parameters: mean μ, and variance σ^2 (their meaning is clarified later)

The Gaussian distribution arises very often in nature: e.g., distribution of IQs in a human population. It is also called the normal distribution, and denoted $\mathcal{N}(\mu, \sigma^2)$.

Expected value

Definition

$$\mathrm{E}\left\{X\right\} = \begin{cases} \sum_{x \in \mathcal{X}} p(x)x & \text{for discrete random variables} \\ \int_{x \in \mathcal{X}} f(x)x & \text{for continuous random variables} \end{cases}$$

Intuition: the average of all values, weighted by their probability; the value “expected” beforehand given the probability distribution.

The expected value is also called *mean*, or *expectation*.

Examples:

- For a fair dice with X the value of a face, $\mathrm{E}\left\{X\right\} = \frac{1}{6}1 + \frac{1}{6}2 + \ldots + \frac{1}{6}6 = 7/2$.
- If X has a Gaussian PDF $\mathcal{N}(\mu, \sigma^2)$, then $\mathrm{E}\left\{X\right\} = \mu$.

Expected value of a function

Consider a function $g : \mathcal{X} \to \mathbb{R}$, depending on some random variable X. Then $g(X)$ is itself a random variable, and:

$$\mathrm{E}\left\{g(X)\right\} = \begin{cases} \sum_{x \in \mathcal{X}} p(x) g(x) & \text{if discrete} \\ \int_{x \in \mathcal{X}} f(x) g(x) & \text{if continuous} \end{cases}$$

Variance

Definition

$$\mathrm{Var}\left\{X\right\} = \boxed{\mathrm{E}\left\{(X - \mathrm{E}\left\{X\right\})^2\right\}} = \mathrm{E}\left\{X^2\right\} - (\mathrm{E}\left\{X\right\})^2$$

Intuition: the "spread" of the random values around the expectation.

$$\begin{aligned}\mathrm{Var}\left\{X\right\} &= \begin{cases} \sum_{x\in\mathcal{X}} p(x)(x - \mathrm{E}\left\{X\right\})^2 & \text{if discrete} \\ \int_{x\in\mathcal{X}} f(x)(x - \mathrm{E}\left\{X\right\})^2 & \text{if continuous} \end{cases} \\ &= \begin{cases} \sum_{x\in\mathcal{X}} p(x)x^2 - (\mathrm{E}\left\{X\right\})^2 & \text{if discrete} \\ \int_{x\in\mathcal{X}} f(x)x^2 - (\mathrm{E}\left\{X\right\})^2 & \text{if continuous} \end{cases}\end{aligned}$$

Examples:

- For a fair dice, $\mathrm{Var}\left\{X\right\} = \frac{1}{6}1^2 + \frac{1}{6}2^2 + \ldots + \frac{1}{6}6^2 - (7/2)^2 = 35/12$.
- If X has a Gaussian PDF $\mathcal{N}(\mu, \sigma^2)$, then $\mathrm{Var}\left\{X\right\} = \sigma^2$.

Notation

We will generically denote $\mathrm{E}\left\{X\right\} = \mu$ and $\mathrm{Var}\left\{X\right\} = \sigma^2$.

Quantity $\sigma = \sqrt{\mathrm{Var}\left\{X\right\}}$ is called *standard deviation*.

Probability: Independence

Definition

Two random variables X and Y are called independent if:

- in the continuous case, $f_{X,Y}(x,y) = f_Y(x) \cdot f_Y(y)$.
- in the discrete case, $p_{X,Y}(x,y) = p_X(x) \cdot p_Y(y)$.

where $f_{X,Y}$ denotes the joint PDF of the vector (X, Y), f_X and f_Y are the PDFs of X and Y, and similarly for the PMFs p.

Examples:

- The event of rolling a 6 with a dice is independent of the event of getting a 6 at the previous roll (or, indeed, any other value and any previous roll).
- The event of rolling *two consecutive* 6-s, however, is not independent of the previous roll!

(Incidentally, failing to understand the first example leads to the so-called gambler's fallacy. Having just had a long sequence of bad–or good–games at the casino, means nothing for the next game!)

Covariance

Definition

$$\mathrm{Cov}\left\{X, Y\right\} = \boxed{\mathrm{E}\left\{(X - \mathrm{E}\left\{X\right\})(Y - \mathrm{E}\left\{Y\right\})\right\}} = \mathrm{E}\left\{(X - \mu_X)(Y - \mu_Y)\right\}$$

where μ_X, μ_Y denote the means (expected values) of the two random variables.

Intuition: how much the two variables “change together” (positive if they change in the same way, negative if they change in opposite ways).

Remark: $\mathrm{Var}\left\{X\right\} = \mathrm{Cov}\left\{X, X\right\}$.

Uncorrelated variables

Definition

Random variables X and Y are uncorrelated if $\mathrm{Cov}\,\{X, Y\} = 0$. Otherwise, they are correlated.

Examples:

- The education level of a person is correlated with their income.
- Hair color is uncorrelated with income (or at least it should be, ideally).

Remarks:

- If X and Y are independent, they are uncorrelated.
- But the reverse is not necessarily true! Variables can be uncorrelated and still dependent.

Table of contents

Vectors of random variables

Consider a vector $\boldsymbol{X} = [X_1, \ldots, X_N]^\top$ where each X_i is a continuous, real random variable. This vector has a *joint PDF* $f(\boldsymbol{x})$, with $\boldsymbol{x} \in \mathbb{R}^N$.

Definitions

Expected value and *covariance matrix* of $\boldsymbol{X}$:

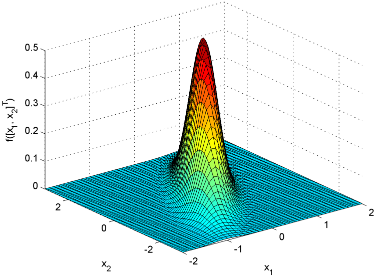
$$\mathrm{E}\{\boldsymbol{X}\} := [\mathrm{E}\{X_1\}, \ldots, \mathrm{E}\{X_N\}]^\top = [\mu_1, \ldots, \mu_N]^\top, \text{ denoted } \boldsymbol{\mu} \in \mathbb{R}^N$$

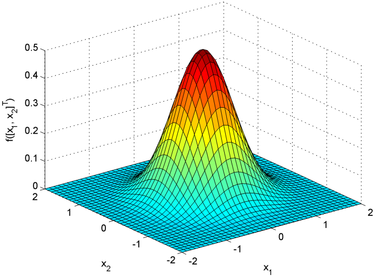
$$\mathrm{Cov}\{\boldsymbol{X}\} := \begin{bmatrix} \mathrm{Cov}\{X_1, X_1\} & \mathrm{Cov}\{X_1, X_2\} & \cdots & \mathrm{Cov}\{X_1, X_N\} \\ \mathrm{Cov}\{X_2, X_1\} & \mathrm{Cov}\{X_2, X_2\} & \cdots & \mathrm{Cov}\{X_2, X_N\} \\ \cdots & \cdots & \cdots & \cdots \\ \mathrm{Cov}\{X_N, X_1\} & \mathrm{Cov}\{X_N, X_2\} & \cdots & \mathrm{Cov}\{X_N, X_N\} \end{bmatrix}$$

$$= \mathrm{E}\left\{(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^\top\right\}, \text{ denoted } \Sigma \in \mathbb{R}^{N,N}$$

Remarks: $\mathrm{Cov}\{X_i, X_i\} = \mathrm{Var}\{X_i\}$. Also, $\mathrm{Cov}\{X_i, X_j\} = \mathrm{Cov}\{X_j, X_i\}$, so matrix Σ is symmetrical.

Example: Multivariate Gaussian (normal)





The PDF of a vector $\boldsymbol{X}$ with a Gaussian joint distribution can be written:

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^N \sqrt{\det(\Sigma)}} \exp\left(-(\boldsymbol{x} - \boldsymbol{\mu}) \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})^\top \right)$$

parameterized by the vector mean $\boldsymbol{\mu}$ and covariance matrix Σ (assumed positive definite, so $\det(\Sigma) > 0$ and Σ^{-1} exists).

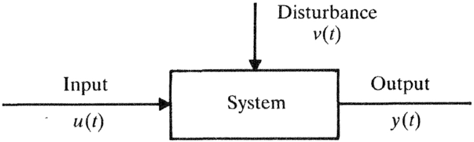
Stochastic process

Definition

A stochastic process $\boldsymbol{X}$ is a sequence of random variables $\boldsymbol{X} = (X_1, \ldots, X_k, \ldots, X_N)$.

It is in fact just a vector of random variables, with the additional structure that the index in the vector has the meaning of time step k.

In system identification, signals (e.g., inputs, outputs) will often be stochastic processes evolving over discrete time steps k.



Zero-mean white noise

Definition

The stochastic process $\boldsymbol{X}$ is zero-mean white noise if: $\forall k$, $\mathrm{E}\left\{X_k\right\} = 0$ (zero-mean), and $\forall k, k' \neq k$, $\mathrm{Cov}\left\{X_k, X_{k'}\right\} = 0$ (the values at different steps are uncorrelated). In addition, the variance $\mathrm{Var}\left\{X_k\right\}$ must be finite $\forall k$.

Stated concisely using vector notation: mean $\boldsymbol{\mu} = \mathrm{E}\left\{\boldsymbol{X}\right\} = \mathbf{0} \in \mathbb{R}^N$ and covariance matrix $\Sigma = \mathrm{Cov}\left\{\boldsymbol{X}\right\}$ is diagonal (with positive finite elements on the diagonal).

In system identification, noise processes often affect signal measurements, and we will sometimes assume that the noise is white and zero-mean.

Stationary process

Signal values at different time steps can be correlated (e.g. when they depend on the output of some dynamic system). Nevertheless, signals are sometimes required to be stationary, in the sense:

Definition

The stochastic process $\boldsymbol{X}$ is stationary if $\forall k$, $\mathrm{E}\left\{X_k\right\} = \mu$, and $\forall k, k', \tau$, $\mathrm{Cov}\left\{X_k, X_{k+\tau}\right\} = \mathrm{Cov}\left\{X_{k'}, X_{k'+\tau}\right\}$.

The mean is the same at every step, whereas the covariance depends on only the relative positions of time steps, not their absolute positions.