Statistical Inverse Methods

Antti Penttilä

Spring 2024

Contents

Chapter 1

Introduction

1.1 Information about course

This is the lecture material for the course "Statistical Inverse Methods", SIM in short. In Finnish, Tilastolliset inversiomenetelmät. Course ID is PAP303.

Five credit points are rewarded from the course. To achieve these points you need to *i*) complete and return weekly exercises, and *ii*) pass the final exam. At least 25 % of the weekly exercises need to be done in order to pass, and completing more will earn you a better grade.

Exercises will include both problems that are to be solved analytically, i.e., with pen and paper, and computer tasks that should be completed using some mathematical or statistical software. We do not specify what kind of software should be used, choose one you are most familiar with or one you would like to learn during the course. Programming or details about specific software are not taught, so you need to have prior knowledge on programming and scientific computing.

Most, if not all the computer task are possible to do using any general purpose mathematical package such as Matlab, Mathematica, Maple etc. Statistical software packages such as R (free, under GNU GPL), or general data-analysis environment such as Python, are also excellent choices for a tool. We do not recommend using low-level programming such as C or Fortran, since too much effort would probably go to writing code for input/output and for producing graphics. On the other hand, software packages with limited amount of generality and versatile programming capabilities such as Excel or SPSS are not recommended either. The University of Helsinki has a license for SAS software, which is a huge statistical (among others) package that is used quite often in, e.g., medical research and business applications, but perhaps because of its vast application areas and history, it is quite complex and a bit cumbersome to use.

Prior knowledge should include mathematical tools that are taught on basic university mathematics courses, e.g., Mathematics for Physicists I and II (Matemaattiset

apuneuvot, FYS1010 and FYS1011). Especially, we will need basic linear algebra and basic multivariate differential calculus. Prior course for statistics is Statistical Analysis of Observations (Havaintojen tilastollinen käsittely, FYS1014). Scientific Computing (Tieteellinen laskenta) I and II (FYS1013 and FYS2085) are recommended, because basic programming, data handling, and plotting are needed in the exercise sessions.

1.1.1 Spring 2024

Up-to-date version of the dates can be found on the course homepage at `https://moodle.helsinki.fi/course/view.php?id=63337`. Lecturer is Dr. Antti Penttilä (antti.i.penttila (a t) helsinki.fi).

1.1.2 Material

The course material, i.e., this handout and exercises, are based on the following course materials or books:

- A. Ekholm, "Johdatus todennäköisyyslaskentaan" and "Johdatus uskottavuuspäättelyyn", handouts
- S. Mustonen, Tilastolliset monimuuttujamenetelmät, book, University of Helsinki
- Course material for "Data-analysis and Inverse Methods in Astronomy, 2012" by M. Juvela, K. Muinonen, H. Haario, and A. Penttilä
- P. Saikkonen, "Lineaariset mallit" and "Epälineaariset mallit", handouts
- C.P. Robert & G. Casella, Monte Carlo Statistical Methods, book, Springer
- E.D. Feigelson & G.J. Jogesh Babu, Modern Statistical Methods for Astronomy — With R Applications, book, Cambridge U. Press

1.1.3 Notations

Throughout this material I will try to maintain a uniform and consistent style on symbol notations. If succeeded, the readability of the formulae will probably be better. Normal weight italic symbols are used for scalars: a, b, c, x. Random variables are usually written with capital letters: X, Y. For theoretical variables, i.e., parameters of distributions and/or theoretical and random properties of random variables such as the expected value or variance, Greek letters are usually used: μ, σ^2.

Functions are written with normal weight and non-italic font: $\sin(), \mathrm{P}()$. If possible, named distributions such as normal distribution are marked with calligraphic font, $\mathcal{N}(\mu, \sigma^2)$.

With multidimensional symbols bold weight is used. Vectors are with bold slanted symbols ($\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\mu}$), and matrices with bold capital non-italics ($\mathbf{X}, \mathbf{A}, \mathbf{B}, \mathbf{\Sigma}$). Vectors can be constructed from components as $\boldsymbol{x} = (1, 2, 3)$ — using () always refers to a column vector, i.e., $n\times1$ matrix. With [] we always refer to matrices, so $\boldsymbol{x} = [1\ 2\ 3]^T$ would also be a (column)vector.

1.2 Random event, probability and random variable

The concepts of random event, probability and random variable are very shortly introduced, since it is probably discussed in previous courses, and we are not going into details behind the philosophical or mathematical measure theory meanings of random variable.

Probability can be interpreted from a frequentist viewpoint — if random phenomena or experiment is repeated and its outcome is statistically stable, the ratio of the number of events where result A is observed, n_A, and the number of all events n will estimate the probability of A. In another words, $\mathrm{P}(A) \approx n_A/n$. Naturally, $0 \leq \mathrm{P}(A) \leq 1$. The actual value of $\mathrm{P}(A)$ may be unknown, but we assume that it is constant.

Frequentist interpretation has some caveats because we often want to consider probability of events that cannot strictly speaking be repeated. Probability is better interpreted through set theory. The sample space $\mathcal{S}$ includes all the possible events s_i. The sample space can be finite, countably infinite or uncountable infinite. All the probability calculus can be derived from three simple axioms for set A in $\mathcal{S}$:

$$\forall A \text{ holds that } \mathrm{P}(A) \geq 0 \tag{1.1}$$

$$\mathrm{P}(\mathcal{S}) = 1 \tag{1.2}$$

$$\text{If } A_1 \cap A_2 \cap \ldots \cap A_n = \emptyset, \text{ then}$$

$$\mathrm{P}(A_1 \cup A_2 \cup \ldots \cup A_n) = \sum_{i=1}^{n} \mathrm{P}(A_i) \tag{1.3}$$

The third axiom tells that if events are mutually exclusive, the probability measure is additive. The third axiom also holds for infinite sets. This set theory interpretation of probability can often be graphically studied by means of Venn diagrams, see Fig. 1.1 for an example.

1.2.1 Some probability laws

Laws of probability can be derived from the three axioms. Some simple and most common definitions are given here. In what follows we will write $A \cap B$ shorter

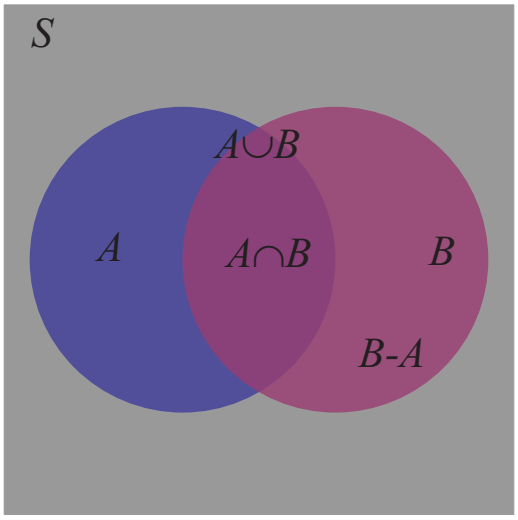


Figure 1.1: Example Venn diagram with some group theory sets.

with AB.

Addition:

$$\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B) - \mathrm{P}(AB) \tag{1.4}$$

that is valid also if $A \cap B \neq \emptyset$.

Conditional probability (*ehdollinen todennäköisyys*): Probability of event A requiring that B has happened, $\mathrm{P}(A|B)$.

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(AB)}{\mathrm{P}(B)} \tag{1.5}$$

Statistical independence (*tilastollinen riippumattomuus*): Events A and B are statistically (or stochastically) independent if and only if $\mathrm{P}(AB) = \mathrm{P}(A)\mathrm{P}(B)$. The usual notation for this is

$$A \perp\!\!\!\perp B \iff \mathrm{P}(AB) = \mathrm{P}(A)\mathrm{P}(B) \tag{1.6}$$

Chain rule:

$$\mathrm{P}(AB) = \mathrm{P}(B)\mathrm{P}(A|B) = \mathrm{P}(A)\mathrm{P}(B|A) \tag{1.7}$$

and theorem of total probability:

$$\mathrm{P}(B) = \sum_{i=1}^{\infty} \mathrm{P}(A_i)\mathrm{P}(B|A_i) \tag{1.8}$$

when the sample space $\mathcal{S}$ has been partitioned into mutually exclusive sets $A_1, \ldots$

Bayes formula:

$$\mathrm{P}(A_i|B) = \frac{\mathrm{P}(A_i)\mathrm{P}(B|A_i)}{\sum_{i=1}^{\infty} \mathrm{P}(A_i)\mathrm{P}(B|A_i)} \tag{1.9}$$

where $\mathrm{P}(A_i)$ is called prior probability and $\mathrm{P}(A_i|B)$ posterior probability.

We will prove and use some of these formulae in the exercises.

1.2.2 Random variable

A random variable (*satunnaismuuttuja*) is a mapping of the result of a random event into real axis. If Y is a random variable, then every possible outcome $s \in \mathcal{S}$ can be coded into real number y. For example, if there are only two possible outcomes, "A will happen, or A will not happen", it is often coded that $Y(A) = 1, Y(\text{not } A) = 0$.

Probability of certain random event to occur follows from set theory notation, $\mathrm{P}(Y = y)$. This is often written also as $\mathrm{P}_Y(y)$ or even as $\mathrm{P}(y)$ for short, if it is evident what random variable is considered. Evidently, from Eqs. (1.1) and (1.2) it follows that $0 \leq \mathrm{P}(Y = y) \leq 1$.

Discrete random variables are such that the set of possible outcomes is finite or countably infinite. Finite set can be for example three categories where the event will fall, and countable infinite set, for example, the set of natural numbers. It is possible that $\mathrm{P}(Y = y_i) = 0$ for some y_i, but from Eq. (1.2) it follows that there must be at least one y_i for which $\mathrm{P}(Y = y_i) > 0$.

Discrete variables can be divided into different scales according to their properties. The nominal scale is the most simple one. In nominal scale the outcome of the event is in finite set of 'categories' for which there is no natural order. An example would be the party a person is voting for. These categories are coded into numbers, but no arithmetic operations are meaningful with the numbers. One cannot say that category '1' is smaller than category '2'. The only possible probability description of nominal variable is to list the probabilities $\mathrm{P}(Y = y)$. The complete list of outcomes and associated probabilities is the *probability mass function* (*pistetodennäköisyysfunktio*)

$$\mathrm{f}(y) = \mathrm{P}(Y = y). \tag{1.10}$$

With ordinal scale variable, the order of the categories is a meaningful concept. For example, many polls may ask if you "agree fully" $(Y = 4)$, "agree partly" $(Y = 3)$, "disagree partly" $(Y = 2)$, or "disagree strongly" $(Y = 1)$. In that case it is meaningful to claim that '4' is more than '3', although operations such as $4 - 3 = 1$ are not meaningful. For ordinal variable, in addition to probability mass function, a *cumulative distribution function* (*kertymäfunktio*) can be defined

$$\mathrm{F}(y) = \mathrm{P}(Y \leq y) = \sum_{u=1}^{y} \mathrm{f}(u). \tag{1.11}$$

See Fig. 1.2 for examples.

The most advance scale for discrete variables is the interval scale. Variable has countable number of outcomes, they can be ordered, and their intervals are meaningful and constant, i.e., $1 < 2 < 3$ and $2 - 1 = 3 - 2 = 1$. Both probability mass function and cumulative distribution function are defined. Furthermore, one can compute with the outcomes, and especially one can compute descriptive statistics such as mean, median or standard deviation.

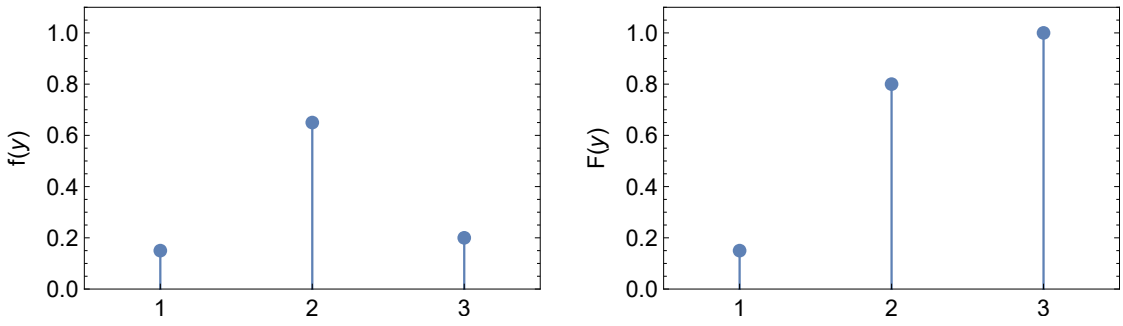


Figure 1.2: Example of probability mass function (on left) and cumulative distribution function (on right) for discrete random variable.

Continuous variables are measured in interval or ratio scales. Ratio scale differs from interval scale by having a unique and non-arbitrary zero value, but there are no real differences in using continuous interval or ratio scale variables in statistics. Most importantly, continuous variables are uncountable infinite. For that reason, the probability of every single outcome is zero. Instead of probability mass function, a non-negative, real valued *probability density function* (pdf, *todennäköisyystiheysfunktio*) is defined so that

$$\mathrm{P}(y_0 < Y \leq y_1) = \int_{y_0}^{y_1} \mathrm{f}(u)du \quad \text{for } y_0 \leq y_1. \tag{1.12}$$

The so-called probability density $\mathrm{f}(y)$ can be non-negative although the probability of single event is zero. The *cumulative density function* (cdf) for a continuous random variable is defined as

$$\mathrm{F}(y) = \mathrm{P}(Y \leq y) = \int_{-\infty}^{y} \mathrm{f}(u)du. \tag{1.13}$$

See Fig. 1.3 for examples.

1.3 Descriptive statistics

The pdf or cdf of a random variable is the complete description of the phenomenon, at least in mathematical sense. However, we often would like to compress that information into some set of numbers that would give us important information on the behavior of the random variable. These numbers are called *statistics* (*tunnusluvut*). In principle, everything that is computed from a pdf or from a random sample is a statistics, but there are some common choices on how distributions or samples are described.

We should remember to make clear difference between theoretical statistics and sample statistics. With theoretical statistics we mean quantities that can be derived

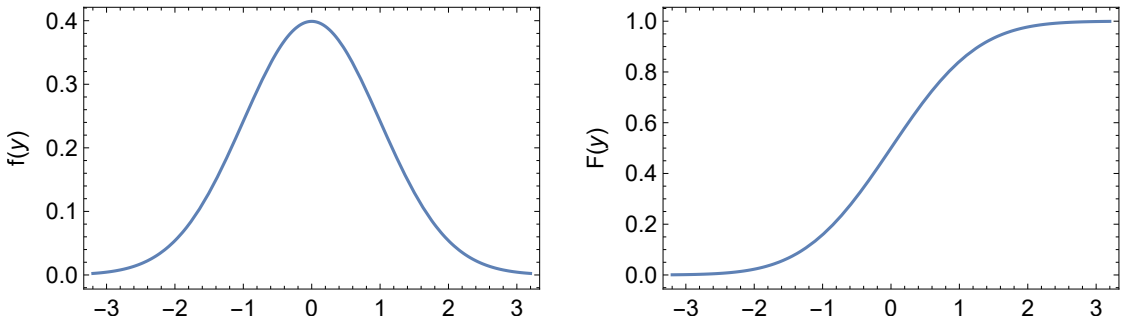


Figure 1.3: Example of probability density function (on left) and cumulative density function (on right) for discrete random variable.

from the pdf of a random variable, even though the pdf might be unknown. The idea is that even if the distribution of the random variable is unknown to us, it 'exists' and we can gather knowledge about it by observing the realized outcomes of the random variable. Theoretical statistics are often marked with Greek letters. The most common example of theoretical statistics and its sample counterpart is the expected value (μ) and the sample mean ($\overline{x}$). Actually, sample mean can also be thought to be random variable ($\overline{X}$) and the mean computed from one particular sample ($\overline{x}$) is the realization of that.

1.3.1 Expectation

The expected value (*odotusarvo*) of variable is the 'center of mass' for a distribution. It is the most common statistics, and many distributions use it as a parameter. Expected value, or the expectation operator $\mathrm{E}(\cdot)$, is defined as

$$\mathrm{E}(Y) = \int_{-\infty}^{\infty} y\, \mathrm{f}(y)\, dy \tag{1.14}$$

for a continuous variable, and

$$\mathrm{E}(Y) = \sum_{y} y\, \mathrm{f}(y) \tag{1.15}$$

for a discrete variable. It is said that the expectation does not exists unless the integral

$$\int_{-\infty}^{\infty} |y|\, \mathrm{f}(y)\, dy \tag{1.16}$$

converges, i.e., has a finite value, and similarly but with sum instead of integral for a discrete variable. The famous example of a distribution without an expected value is the Cauchy distribution.

Expectation is an important statistics and is is useful to know some basic properties of the $\mathrm{E}(\cdot)$ operator. First, it should be noted that a function of a random variable is also a random variable, i.e., if $V = \mathrm{g}(Y)$ then V is a random variable. It can be shown that the expectation of V that is a function of Y can be derived without knowing the pdf of V by

$$\mathrm{E}(V) = \int_{-\infty}^{\infty} \mathrm{g}(y)\,\mathrm{f}(y)\,dy. \tag{1.17}$$

With discrete variable the same holds but with sum instead of integral. Another property is that expectation is a linear operator, i.e.

$$\mathrm{E}(Y_1 + \cdots + Y_n) = \mathrm{E}(Y_1) + \cdots + \mathrm{E}(Y_n) \tag{1.18}$$

$$\mathrm{E}(cY) = c\,\mathrm{E}(Y)\text{ , where } c \text{ is constant} \tag{1.19}$$

1.3.2 Variance

As expectation is a location measure, variance is a dispersion measure. It describes how much a random variable deviates from its expectation on average. Variance is derived as

$$\mathrm{var}(Y) = \mathrm{E}[(Y - \mathrm{E}(Y))^2] = \int_{-\infty}^{\infty} (y - \mathrm{E}(Y))^2\,\mathrm{f}(y)\,dy \tag{1.20}$$

for a continuous variable. Variance must be finite to exist. Instead of operators E and var, symbols μ and σ^2 are often used.

Some properties of variance are dealt next. First,

$$\mathrm{var}(aY + b) = a^2\mathrm{var}(Y). \tag{1.21}$$

Second, for the variance of the sum of *independent* variables $Y_1, \ldots, Y_n$ $\perp\!\!\!\perp$ holds that

$$\mathrm{var}(Y_1 + \cdots + Y_n) = \mathrm{var}(Y_1) + \cdots + \mathrm{var}(Y_n), \tag{1.22}$$

but the same is generally not true if the variables are not independent.

1.3.3 Other statistics

Other commonly used statistics to describe the shape of the distribution include skewness (γ_1, *vinous*) and kurtosis (γ_2, *huipukkuus*). Both are derived from the central moments μ_k of a distribution, $\mu_k = \mathrm{E}[(Y - \mu)^k]$, so that

$$\gamma_1 = \frac{\mu_3}{\sigma^3}\text{ , and }\ \gamma_2 = \frac{\mu_4}{\sigma^4} - 3. \tag{1.23}$$

Kurtosis is defined so that it is zero for standard normal distribution $\mathcal{N}(0, 1)$. Skewness is zero for all symmetric distributions.

One important family of statistics are defined by quantiles. The p'th quantile is the value ξ for which

$$\mathrm{F}(\xi) = p. \tag{1.24}$$

Especially median is the quantile at $1/2$, the middle value of a distribution. Lower or first quartile is at $1/4$ and upper or third quartile at $3/4$. Median and other quantiles are so-called robust statistics, since their values are not heavily effected if the distribution has very wide tails, unlike the expectation or the variance, for example. An example of some of the abovementioned statistics is given in Fig. 1.4.

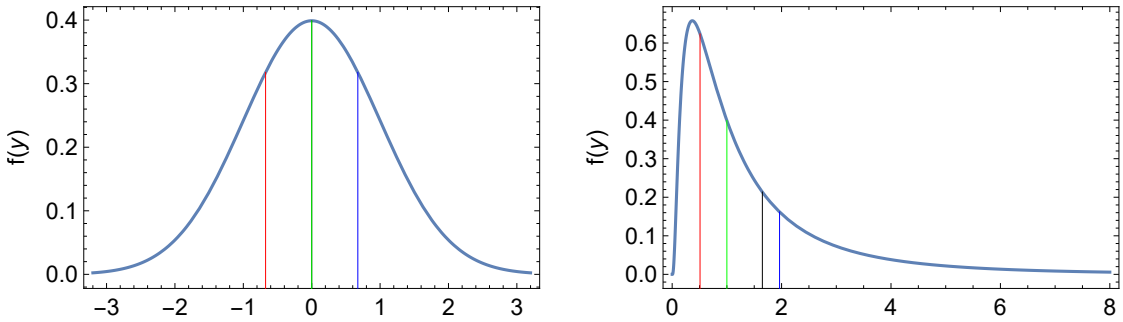


Figure 1.4: Symmetric distribution (normal) on left, and skew distribution (log-normal) on right. For both the place of expected value is marked with black line, median with green, and 1st and 3rd quartiles with red and blue. For symmetric distribution median and μ have the same value.

1.3.4 Covariance

We have not yet introduced multivariate random variables, but still it is best to mention covariance and correlation at this point. Covariance deals with two-dimensional random variable (U, V), and it measures the linear dependence between the variables. Definition for covariance is

$$\mathrm{cov}(U, V) = \mathrm{E}[(U - \mathrm{E}(U))(V - \mathrm{E}(V))] = \mathrm{E}(UV) - \mathrm{E}(U)\mathrm{E}(V) \tag{1.25}$$

Without proof we mention that the expected value of the product of two random variables is

$$\mathrm{E}(UV) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} uv\, \mathrm{f}(u, v)\, du\, dv \tag{1.26}$$

for continuous variables. The $\mathrm{f}(u, v)$ is the joint distribution (*yhteisjakauma*) of U and V. Correlation is covariance that is normalized with standard deviations as

$$\mathrm{cor}(U, V) = \frac{\mathrm{cov}(U, V)}{\sigma_U \sigma_V}. \tag{1.27}$$

Independence is a wider concept than only linear independence, so zero covariance does not imply statistical independence, but the opposite direction is true,

$$U \perp\!\!\!\perp V \implies \mathrm{cov}(U, V) = \mathrm{cor}(U, V) = 0. \tag{1.28}$$

With the concept of covariance we can generalize Eq. (1.22) about the variance of the sum of independent variables to apply also for dependent ones,

$$\mathrm{var}(U + V) = \mathrm{var}(U) + \mathrm{var}(V) + 2\mathrm{cov}(U, V), \tag{1.29}$$

even when $U \not\perp\!\!\!\perp V$.

1.3.5 Sample statistics

All the aforementioned theoretical statistics have their sample counterparts, or sample estimates (*otosestimaatti*), to be exact. The concept and the derivation of an estimate is introduced in the next chapter, but for now we list formulae for these common statistics without proving their estimate properties.

Sample mean $\overline{x}$ is computed as

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \tag{1.30}$$

(sample) standard error as

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})^2, \tag{1.31}$$

and sample covariance as

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}). \tag{1.32}$$

The denominator $n - 1$ is needed instead of n for the estimator to be unbiased, but this is a topic of estimation theory and not dealt with here. Estimates for different quantiles are self-evident and can be made by sorting the sample with n observations and searching for k'th value so that $k/n = p$.

Mean and variance are not robust statistics. If the underlying distribution has heavy tails, i.e., the probability for an extreme values is not 'small', the sample estimate may vary a lot from one sample to another. With astronomical observations, for example, sampling more and more is often not an option, so it is difficult to know whether observations come from a heavy-tailed distribution or not, or if some of the observations are simply wrong or affected by another process. Therefor, it is quite difficult to objectively say if some observations are outliers and should be left out from the analysis or not. However, due to the large effect that 'unusual' observations can have in mean or variance estimates, they are sometimes left out, i.e., data is censored or trimmed. Common practices include trimming out observations with distance to mean larger than three standard deviations and then computing mean and variance again.

1.4 Distributions

The distribution, either the probability mass function for a discrete variable or the probability density function for continuous, is the complete description of a random variable. Alternatively, cumulative functions can be used. One should note that all random variables have distribution and there are infinitive number of distributions, but only few of them are 'known' in the sense that they are named and their formula is given. In this chapter we will list some univariate distributions and their statistics.

1.4.1 Discrete distributions

Bernoulli

Most simple discrete distribution is the Bernoulli distribution for a binary random variable, i.e., a variable with two possible outcomes, 0 and 1. If the probability of having 1 is π, then

$$Y \sim \mathcal{B}(\pi) \implies \mathrm{f}(y) = \pi^y (1-\pi)^{1-y}, \tag{1.33}$$

$$\mathrm{E}(Y) = \pi,\ \mathrm{var}(Y) = \pi(1-\pi),\ y \in \{0, 1\}. \tag{1.34}$$

Notice the notation, $Y \sim \mathcal{B}(\pi)$ should be read as Y has/follows/obeys Bernoulli distribution with parameter π.

Binomial distribution

When more than one identical and independent Bernoulli trials are sampled, the total number of successes (outcome 1) when n trials are done is given by the binomial distribution

$$Y \sim \mathrm{Bin}(n, \pi) \implies \mathrm{f}(y) = \frac{n!}{y!(n-y)!}\pi^y (1-\pi)^{n-y}, \tag{1.35}$$

$$\mathrm{E}(Y) = n\pi,\ \mathrm{var}(Y) = n\pi(1-\pi),\ y = 0, \ldots, n. \tag{1.36}$$

Poisson distribution

Poisson distribution can be used to model counts, i.e., how many times some (rare) event has occurred in one time unit. Good example could be the number of photons that hit the CCD sensor on a telescope per time unit. When the intensity parameter, i.e., the expected number of events per unit time, is λ, the distribution is

$$Y \sim \mathcal{P}(\lambda) \implies \mathrm{f}(y) = \exp(-\lambda)\frac{\lambda^y}{y!}, \tag{1.37}$$

$$\mathrm{E}(Y) = \lambda,\ \mathrm{var}(Y) = \lambda,\ y = 0, \ldots. \tag{1.38}$$

Examples of Poisson and binomial pdf's are shown in Fig. 1.5.

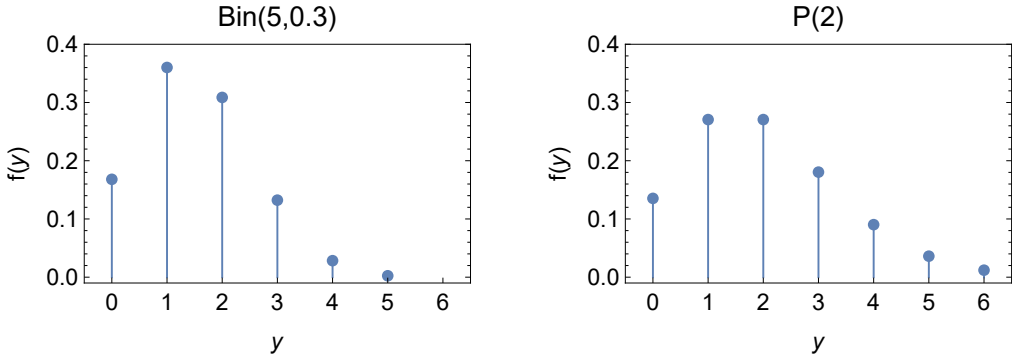


Figure 1.5: Pdf's of binomial (on left) and Poisson (on right) distributions.

1.4.2 Continuous distributions

Normal distribution

Normal distribution is by far the most common distribution due to the fact that it is the limiting distribution of many derived random variables by the central limit theorem, and thus can be used as an approximative distribution to many otherwise too complicated or non-traceable distributions. Gauss derived the distribution to describe errors observed in the movements of planets and planetoids. With parameters μ and σ^2 the distribution is

$$Y \sim \mathcal{N}(\mu, \sigma^2) \implies \mathrm{f}(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right), \tag{1.39}$$

$$\mathrm{E}(Y) = \mu, \ \mathrm{var}(Y) = \sigma^2, \ y \in \mathbb{R}. \tag{1.40}$$

The term standardization (*standardointi*) means that the expected value (or mean) is subtracted from the original value, and the result is scaled (divided) with the standard deviation. This operation is not limited to normal distribution in any way, but if general normal variable $Y \sim \mathcal{N}(\mu, \sigma^2)$ is standardized, the results has $\mathcal{N}(0, 1)$ distribution, a.k.a. the standard normal distribution.

The probability mass in normal distribution between $\mu - k\sigma$ and $\mu + k\sigma$ is approximately 68% with $k = 1$, 95% with $k = 2$, and 99% with $k = 3$. These are the famous one, two and three-sigma intervals that are commonly used in statistical tests and error limits.

The central limit theorem states that, under quite common conditions, pdf of the scaled sum Z of independent and identically distributed (i.i.d.) random variables approaches to normal distribution when the number of summed variables increases without limit. Precisely

$$\text{For i.i.d } Y_1, \ldots, Y_n \text{ with } \mathrm{E}(Y_i) = 0 \text{ and } \mathrm{var}(Y_i) = \sigma^2, \tag{1.41}$$

$$Z = \frac{1}{\sqrt{n}} \sum_i^n Y_i \overset{approx}{\sim} \mathcal{N}(0, \sigma^2), \text{ as } n \to \infty.$$

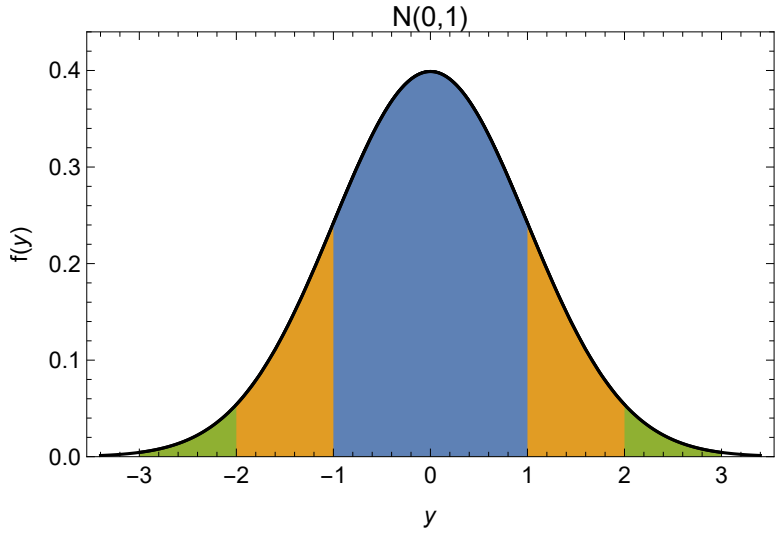


Figure 1.6: One (blue), two (orange) and three-sigma (green) areas in normal distribution.

This has evident implication to the sample mean $\overline{X}$ as a random variable. For large samples, the sample mean should have normal distribution around the true, unknown mean, and the variance of the sample mean around the true value is σ^2/n.

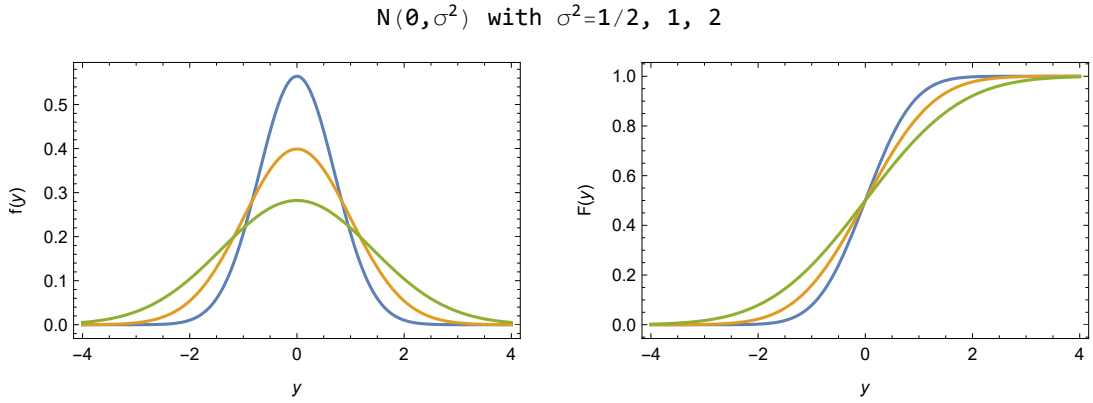


Figure 1.7: Pdf and cdf of normal distributions with different σ.

Exponential distribution

Exponential distribution can be used to model waiting times between two successive events from Poisson distributed variable. When the intensity parameter (same interpretation as with Poisson) is λ, the distribution is

$$Y \sim \mathrm{Exp}(\lambda) \implies \mathrm{f}(y) = \lambda \exp(-\lambda y), \tag{1.42}$$

$$\mathrm{E}(Y) = 1/\lambda,\ \mathrm{var}(Y) = 1/\lambda^2,\ y \geq 0. \tag{1.43}$$

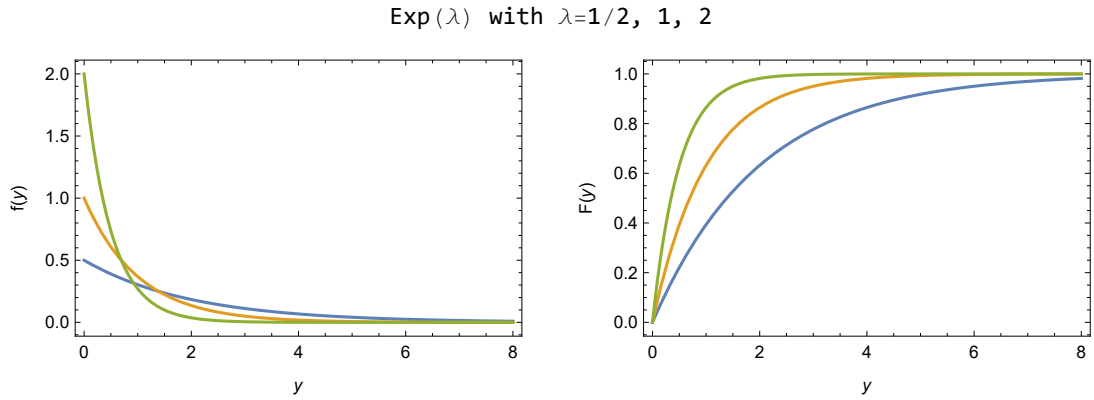


Figure 1.8: Pdf and cdf of exponential distributions with different λ.

Gamma distribution

Gamma distribution is a general case of the exponential distribution, exponential is gamma with index $\kappa = 1$. Gamma is a flexible distribution and is used to model lifetimes and other distances before or between events. With index κ and scale λ, the distribution is

$$Y \sim \text{Gamma}(\kappa, \lambda) \implies \text{f}(y) = \frac{\lambda^\kappa y^{\kappa-1} \exp(-\lambda y)}{\Gamma(\kappa)}, \tag{1.44}$$

$$\text{E}(Y) = \kappa/\lambda, \ \text{var}(Y) = \kappa/\lambda^2, \ y \geq 0 \tag{1.45}$$

where $\Gamma()$ is the gamma function.

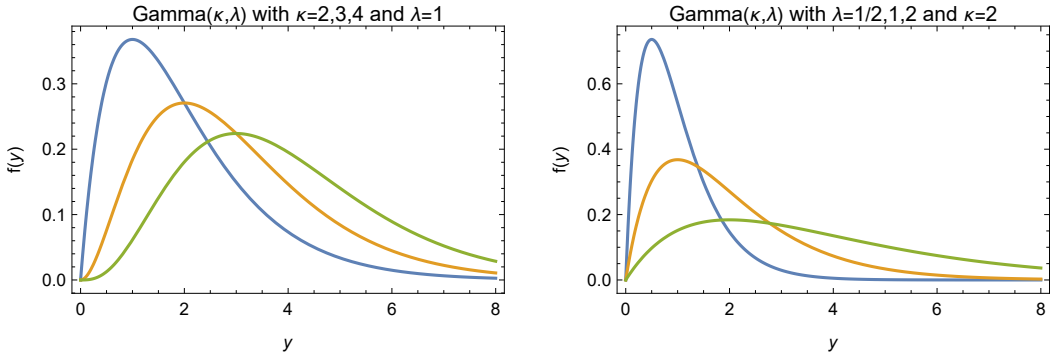


Figure 1.9: Pdf's of gamma distributions with different κ and λ.

Log-normal distribution

Log-normal distribution is yet another distribution for positive-valued variable, and as its name suggest, it is the result of logarithm of a normal-distributed variable. As the normal distribution can be justified through the central limit theorem and sum of i.i.d. variables, log-normal is the limiting distribution for the product

of i.i.d. variables. With parameters μ and σ^2, which refer to the underlying normal distribution, the log-normal distribution is

$$Y \sim \mathcal{LN}(\mu, \sigma^2) \implies \mathrm{f}(y) = \frac{1}{y\sqrt{2\pi}\sigma} \exp\left(-\frac{(\ln(y)-\mu)^2}{2\sigma^2}\right), \tag{1.46}$$

$$\mathrm{E}(Y) = \exp\left(\mu + \frac{1}{2}\sigma^2\right), \ \mathrm{var}(Y) = \exp(\sigma^2 - 1)\exp(2\mu + \sigma^2), \ y \geq 0. \tag{1.47}$$

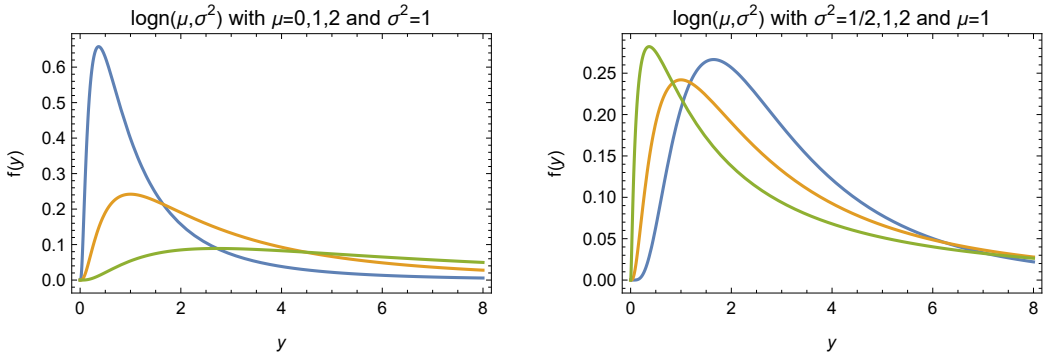


Figure 1.10: Pdf's of log-normal distributions with different μ and σ^2.

Distribution of a function of a random variable

Functions of a random variables introduce new random variables which have their own distributions. The new distribution can be found by replacing the original variable by the inverse transform function and scaling by the derivative of the transform. More formally, let us have original random variable U with known distribution $\mathrm{f}_U(u)$, and a transform function from U to V: $V = \mathrm{g}(U)$. With the following method function g must be differentiable. With inverse transform $\mathrm{h}(V) = \mathrm{g}^{-1}(V) = U$ we can define that

$$\mathrm{f}_V(v) = \mathrm{f}_U(\mathrm{h}(v)) \left|\frac{d\mathrm{h}(v)}{dv}\right| \tag{1.48}$$

Please note that if inverse transform $u = \mathrm{h}(v)$ is a multiple-valued function, for example $u = \pm\sqrt{v}$, then all the possible pdf values must be summed together for $\mathrm{f}_V(v)$, e.g., $\mathrm{f}_V(v) = \mathrm{f}_U(-\sqrt{v})\,|d| + \mathrm{f}_U(\sqrt{v})\,|d|$.

1.5 Statistical plots

A large part of data analysis is to describe the data with methods that compress the important information with numbers (statistics) or with figures. We show here a few typical plots for one-dimensional data, and a scatterplot for multi-dimensional data. Previous pages have already shown examples of probability distribution

plots for both discrete and continuous variables. The corresponding plot for sample data is histogram.

Histogram collects data into bins, and plots the bins so that their height (discrete variable) or area (continuous variable) corresponds to the frequency of the observations in bins. If the purpose of a histogram is to compare against a theoretical distribution, the frequencies must be scaled so that their heights (discrete) or areas (continuous) sum up to one. The number of bins can be chosen freely, but one 'rule-of-thumb' suggests to use number of bins between $\sqrt{n}$ and $2\sqrt[3]{n}$ for data with n observations.

For data with outliers or otherwise long tails, the widths of the bins can differ in the histogram. Especially then, one must remember that the area of the 'bar' in the histogram is what counts, not the height. An example is shown in Fig. 1.11.

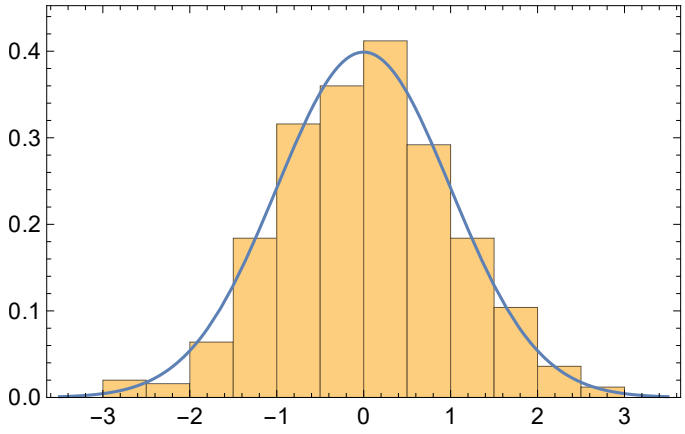


Figure 1.11: Pdf of normal distribution and histogram of 500 normal-distributed random numbers.

If distributions of several variables need to be compared in one figure, a box-and-whiskers plot is quite handy choice. Box-and-whiskers plot shows the range where the data is, and its quartiles. In that way one gets a rough idea on how the data is spread, and about the symmetric / non-symmetric properties of the distribution and tails. The plot is drawn using smallest and largest values of data as 'whiskers', and a box from the first to the third quartile. The median or mean value is drawn in the middle of the box. Example in Fig. 1.12 will enlighten the principle. If there seems to be outliers in the data, the 'whiskers' might use, e.g., 1% and 99% quantiles as the endpoints instead of the smallest and the largest value.

Scatterplots (*sirontakuviot*) are used to show dependence between two or more variables. With many variables the individual i vs. j plots can be organized into matrix of scatterplots.

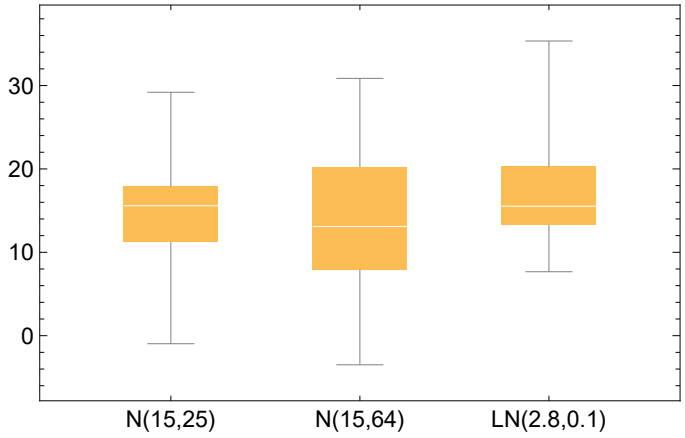


Figure 1.12: Box-and-whiskers plot of three samples of 100 observations from different distributions. The first two are from normal distribution, and the third from log-normal.

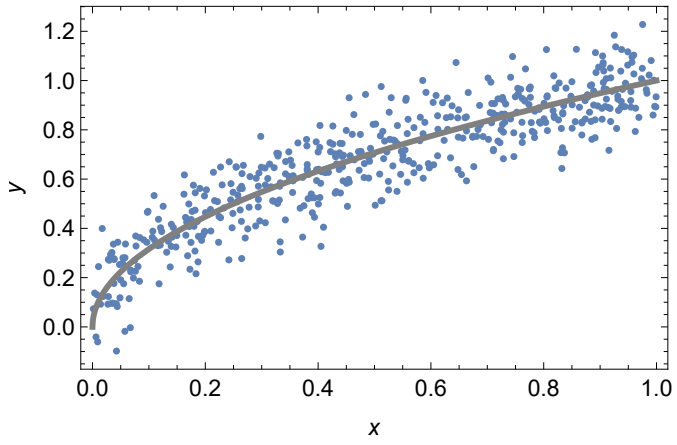


Figure 1.13: Scatterplot of data with $\sqrt{x}$-dependence and normally distributed errors.

Chapter 2

Statistical inference

Statistical inference (*tilastollinen päättely*) is the mathematical theory behind estimates and their distributions. Estimates can be constructed in a way that statistical hypothesis can be tested against their distributions. Estimate and its distribution is the link between model (i.e., distribution and its parameters) and data.

2.1 Likelihood

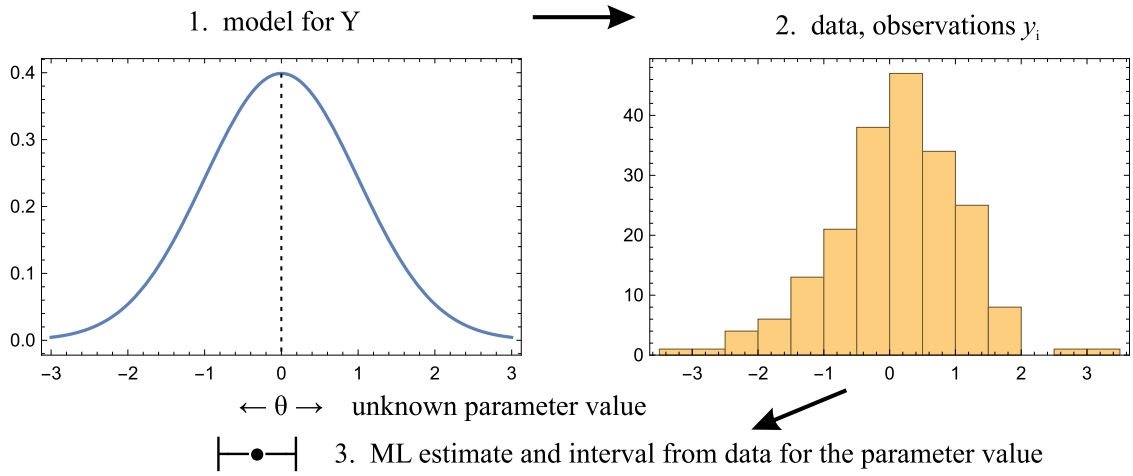


Figure 2.1: Concepts of model, parameter, data, and estimate in statistical inference.

Likelihood (*uskottavuus*) is the key concept in statistical inference. The theory is developed by R.A. Fisher at the beginning of the 20th century. Likelihood deals with data, model, and parameters. First of all, we need to have a model. Model is the statistical distribution that we believe the random variable Y should obey, so the model is a probability density function $f_Y(\cdot)$. Model has parameters but their

values are unknown. In likelihood problems the parameter vector is often noted with $\boldsymbol{\theta}$, although individual distributions usually have traditional conventions with the parameter symbols. For example, normal distribution has $\boldsymbol{\theta} = (\mu, \sigma^2)$.

The final component in likelihood is data. Very seldom we are doing inference based on a single observation y, almost always the data consists of observations $y_1, \ldots, y_n$. In that case, the data is a vector of observations, $\boldsymbol{y}$. In the more general case, the data is vector of multidimensional observations, i.e., matrix $\mathbf{Y}$.

We are not dealing with random processes here, so the observations y_i are identically distributed and the model or its parameters are not assumed to change with time. If there is (auto)correlation between consecutive observations (y_i, y_{i+k}) we are dealing with time series (*aikasarja*), but here we do not consider such cases. We limit ourselves to independent observations, so together with the assumption of non-varying model we deal with *independent, identically distributed* (i.i.d.) observations $\boldsymbol{y} = (y_1, \ldots, y_n)$.

The idea of likelihood is quite simple and straightforward. Let us say that we have reasons to believe that our data is from process that can be described with the normal distribution having fixed and known variance of 1. The only unknown parameter is the expectancy μ. What if we have one observation y_1? We cannot say much, but our best guess would be that $\mu = y_1$, as in Fig. 2.2 a).

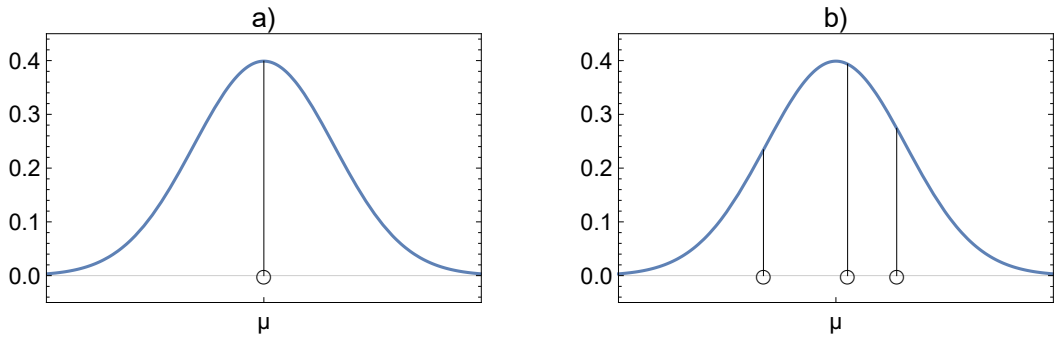


Figure 2.2: Example of normal model with one observation (a) and with three observations (b).

Next, we consider a case with three observations $\boldsymbol{y} = (y_1, y_2, y_3)$, as in Fig. 2.2 b). Intuitively, we should place our normal distribution so that it would somehow fit to all three observations in the best possible way. What is the best possible way? If our model $Y \sim \mathcal{N}(\mu, 1)$ is correct, the probability (density) of observing $Y = y_1$ can be computed from $\mathrm{f}_Y(y_1; \mu, 1)$. As the observations are i.i.d., the joint probability of observing all three can be computed as a product of individual probabilities (densities), $\mathrm{f}_Y(\boldsymbol{y}; \mu, 1) = \mathrm{f}_Y(y_1; \mu, 1) \times \mathrm{f}_Y(y_2; \mu, 1) \times \mathrm{f}_Y(y_3; \mu, 1)$. Please note that with likelihood and related fields both the data and the parameters are usually written out with the pdf as $\mathrm{f}_Y(\boldsymbol{y}; \boldsymbol{\theta})$. The abovementioned procedure is, in a nutshell, the maximum likelihood principle.

2.1.1 Likelihood function

Following the previous procedure we can formulate the likelihood function $\mathrm{L}(\cdot)$ in a more formal way. Likelihood function is

$$\mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}) = \mathrm{c}(\boldsymbol{y})\, \mathrm{f}_Y(\boldsymbol{y}; \boldsymbol{\theta}), \tag{2.1}$$

where the pdf is the joint density function for $\boldsymbol{y}$. Note the small change of paradigm — likelihood function is used to estimate the unknown parameter vector $\boldsymbol{\theta}$, so that is the main parameter of the function, the observed data $\boldsymbol{y}$ is a 'secondary parameter'.

The function $\mathrm{c}(\boldsymbol{y})$ in Eq. (2.1) can be any function involving only the data and not the parameter vector, and in that sense the likelihood function is not uniquely defined. Any function $\mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}) \propto \mathrm{f}_Y(\boldsymbol{y}; \boldsymbol{\theta})$ is likelihood function. This fact can be used to clean out unnecessary constants (i.e., terms independent of $\boldsymbol{\theta}$) from the likelihood, making it a bit simpler.

If we have i.i.d. observations, as we do in almost all the examples here, the likelihood function is the product of the one-dimensional distributions:

$$\mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}) \propto \prod_{i=1}^{n} \mathrm{f}_Y(y_i; \boldsymbol{\theta}) \text{ , if } \boldsymbol{y} \text{ is i.i.d.} \tag{2.2}$$

The likelihood function is used together with the maximum likelihood principle (*suurimman uskottavuuden periaate*). The principle simply states, that we should find the value (i.e., estimate) for our unknown parameter $\boldsymbol{\theta}$ so that we will maximize the likelihood function for the observed data $\boldsymbol{y}$. As L is defined through the joint probability density, we are essentially maximizing the probability of the parameter value, given the data.

In the example in Fig. 2.2 b) we had three observed values: $-1.2, 0, 0.7$. The likelihood function is $\mathrm{L}(\mu) \propto \exp\left(-\left((-1.2-\mu)^2 + (0-\mu)^2 + (0.7-\mu)^2\right)/2\right)$. It is not too hard to see that setting $\mu = -1/6$ will maximize the likelihood, see Fig. 2.3.

Log-likelihood function

The likelihood function is a product of pdf's, and the aim is to maximize that. Taking any monotonic and increasing function of L will not alter the values where the function reaches its extrema points. The logarithm function can be used to reduce the likelihood into simpler form, because logarithm of a product is a sum of logarithms. Therefore, maximum likelihood problems are often solved using the log-likelihood function (*log-uskottavuusfunktio*). Log-likelihood function $\mathrm{l}(\cdot)$ is simply

$$\mathrm{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \log\left(\mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}\right), \tag{2.3}$$

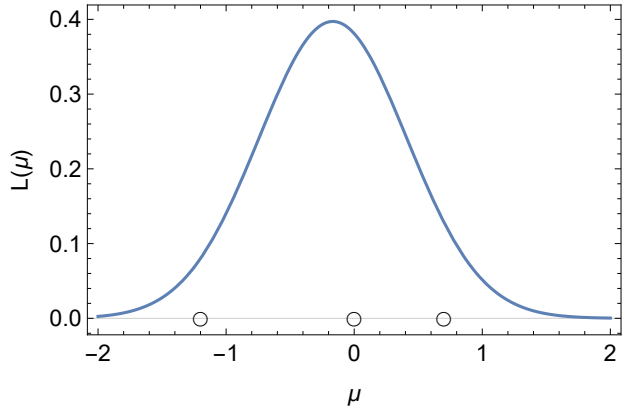


Figure 2.3: Likelihood function of normal model with three observations as in Fig. 2.2 b).

where log stands for natural logarithm. Another convenient property of logarithm is that $\log(\exp(x)) = x$. Many statistical distributions belong to the so-called exponential family, normal distribution being one of them, so the exponential form in likelihood function is quite common. With log-likelihood one can change from a product of exponents to a sum without the exponent functions.

With the log-likelihood function our example in Fig. 2.2 b) would reduce to a task of maximizing $\mathrm{l}(\mu) = -\left((-1.2-\mu)^2 + (0-\mu)^2 + (0.7-\mu)^2\right)$, see Fig. 2.4.

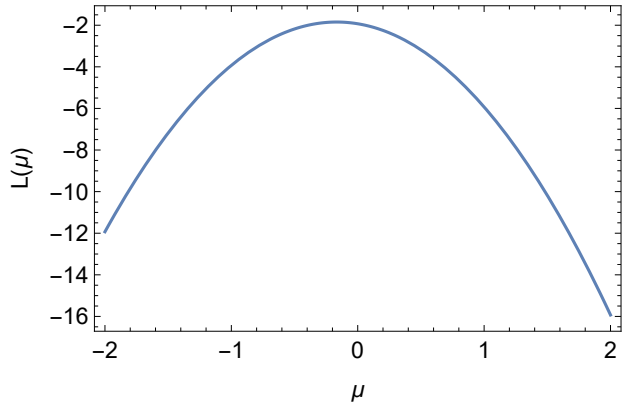


Figure 2.4: Log-likelihood function of normal model with three observations as in Fig. 2.2 b).

2.1.2 Maximum likelihood estimate

The concept of likelihood defines the maximum likelihood (ML) principle (*suurimman uskottavuuden periaate*) in statistics. The maximum likelihood estimate (MLE) of the unknown parameter in our probability model, given the data, is the value $\hat{\boldsymbol{\theta}}$

that maximizes the likelihood (or log-likelihood) function:

$$\mathrm{L}(\hat{\boldsymbol{\theta}}; \boldsymbol{y}) \geq \mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}) \; \forall \boldsymbol{\theta}. \tag{2.4}$$

This $\hat{\boldsymbol{\theta}}$ is the point-estimate (*piste-estimaatti*) to $\boldsymbol{\theta}$.

In most of the cases the likelihood and log-likelihood functions are at least twice differentiable over the whole parameter space. If this is the case, the MLE can be found by studying the first and second derivatives of the (log-)likelihood function. Extrema points of continuous and differentiable functions have zero value of the first derivative. Furthermore, if the extremum point is a maximum, the value of the second derivative at that point is negative.

The conditions described before form the so-called likelihood equations. In the general case the parameter is a vector (of length d here), and the vector of first partial derivatives is called the score function $\mathrm{u}(\cdot)$:

$$\mathrm{u}(\boldsymbol{\theta}; \boldsymbol{y}) = \nabla \, \mathrm{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \left(\frac{\partial \, \mathrm{l}}{\partial \, \theta_1}, \ldots, \frac{\partial \, \mathrm{l}}{\partial \, \theta_d} \right), \tag{2.5}$$

and the Hessian matrix $\mathbf{H}$ is the matrix of the second order partial derivatives:

$$\mathbf{H} = \nabla \nabla^T \, \mathrm{l}(\boldsymbol{\theta}; \boldsymbol{y}) = \left[\frac{\partial^2 \, \mathrm{l}}{\partial \theta_i \, \partial \theta_j} \right]_{ij}. \tag{2.6}$$

With these notations, the MLE $\hat{\boldsymbol{\theta}}$ satisfies the likelihood equations because $\mathrm{u}(\hat{\boldsymbol{\theta}}; \boldsymbol{y}) = \mathbf{0}$ and $\mathbf{H}$ at $\hat{\boldsymbol{\theta}}$ is negative definite.

Properties of maximum likelihood estimate

MLE has some nice properties which make it even more important in statistics. We list the most important here, invariance and asymptotic properties. First, MLE is invariant in re-parametrization. If we would change our parameter of interest so that we would use parameter $\phi := \mathrm{g}(\theta)$, the MLE of the re-parametrized model would still be $\hat{\phi} = \mathrm{g}(\hat{\theta})$.

What is even more important with MLE is that we know its asymptotic distribution, and it is the normal distribution. The proof of that relies on the central limit theorem, but is far too cumbersome for us. So, without proof, we state that

$$\hat{\boldsymbol{\theta}} \xrightarrow{\sim} \mathcal{N}_d \left(\boldsymbol{\theta}, (-\mathbf{H})^{-1} \right). \tag{2.7}$$

That means, at least, four things. First of all, it states that if we have 'enough' data, MLE will approximately follow normal distribution. Note that as the parameter here is a vector, the distribution is multidimensional.

Second, MLE is unbiased. This means that the expectation of MLE is the 'true' $\boldsymbol{\theta}$. Third, MLE is efficient. This concept has not been mentioned before, but it means that the variance of MLE is the smallest possible over all estimators.

Fourth consequence is very important in practice — we have an asymptotic variance for the MLE, so we know how much it typically varies around the true $\boldsymbol{\theta}$. This is the basis for confidence intervals and statistical tests. The asymptotic variance for vector parameter is expressed through the expectation of the Hessian matrix, i.e., the second partial derivatives of the log-likelihood function. While this may seem a bit cumbersome, the good thing is that we usually do not need to derive estimators and their variances ourselves. Somebody else has gone through the trouble and done that for us using the abovementioned equations. For many practical cases the formulas can be reduced to quite simple forms, for example that the variance of mean $\overline{x}$ for normal model is σ^2/n.

For practical use of Eq. (2.7), we will need to plug in some actual numbers instead of the theoretical variables $\boldsymbol{\theta}$ and $\mathbf{H}$. We do this by replacing with our best guess, the MLE. So, $\boldsymbol{\theta} \rightarrow \hat{\boldsymbol{\theta}}$ as the expected value, and also everywhere in $\mathbf{H}$. The negative Hessian matrix $-\mathbf{H}$ where $\boldsymbol{\theta}$ is replaced with $\hat{\boldsymbol{\theta}}$ is called the Fischer information matrix.

2.2 Statistical tests

From estimators and their distributions we can continue to statistical tests and confidence intervals. Let us first deal with confidence intervals.

2.2.1 Confidence intervals

The MLE is a point-estimate, it gives us the most probable value for the unknown parameter of our model. In the same manner, any statistics, whether MLE or any other $t := t(\boldsymbol{y})$, are point-estimates. On the other hand, the data that we have observed, $\boldsymbol{y}$, is just one possible outcome of the random process. If we would repeat the experiment or redo the observations, we would get different data vector $\boldsymbol{y}^*$. Following the thought, we would also get another value for the statistics, t^* that would probably differ from the original t. As the observations $\boldsymbol{y}$ and $\boldsymbol{y}^*$ are both realizations of a random variable $\boldsymbol{Y}$, also the *estimates* t and t^* are realizations of a random *estimator* $T := t(\boldsymbol{Y})$.

For that reason, often the point-estimate alone is not enough for us for data-analysis purposes. A more interesting would be to know an interval where the statistics would most probably be, even if we would repeat the experiment over and over again. This interval is called the confidence interval (CI; *luottamusväli*), or the credible interval in Bayesian inference.

The p 100 % confidence interval (e.g., 95 %) for parameter $\boldsymbol{\theta}$ is the region Ω_p where the true value of parameter lies with p 100 % confidence. More formally

$$\mathrm{P}(\boldsymbol{\theta} \in \Omega_p) = p, \tag{2.8}$$

although there are some philosophical issues in frequentist probability concept that require slightly different formulation*. The Eq. (2.8) does not define how the area Ω_p is chosen. There are some options for that, but with symmetric distributions (of T) all the options lead to the same conclusion — the area Ω_p should be chosen so that it is a symmetric interval around the $\boldsymbol{\theta}$, and only $(1-p)$ 100 % of the density is left out from the tails of the pdf. Thus, CI for one-dimensional parameter and symmetric distribution is such that

$$\mathrm{P}(\hat{\theta} - c \leq \theta \leq \hat{\theta} + c) = p. \tag{2.9}$$

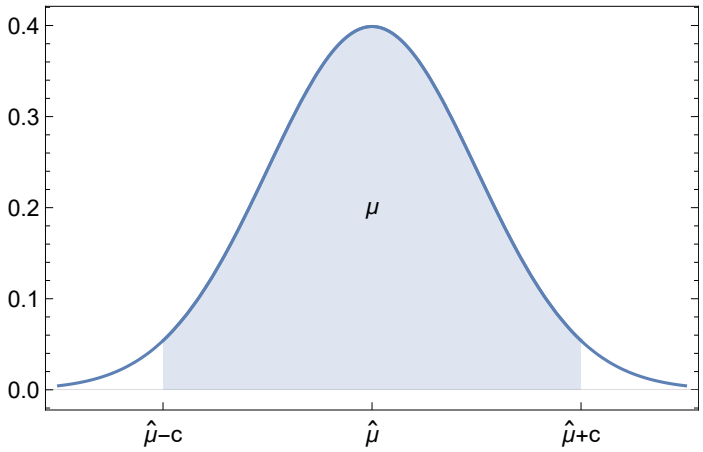


Figure 2.5: Confidence interval $(\hat{\mu} - c, \hat{\mu} + c)$ for μ, when data is from normal distribution.

Asymptotic confidence interval for maximum likelihood estimate

The MLE of a parameter has the normal distribution as the asymptotic distribution, as showed in Eq. (2.7). The variance of that distribution is given by the Hessian matrix as $(-\mathbf{H})^{-1}$, or, in practice, by the Fischer information matrix $\mathbf{F}$ where the unknown parameter $\boldsymbol{\theta}$ has been replaced by $\hat{\boldsymbol{\theta}}$:

$$\mathbf{F} = -\mathrm{E}(\mathbf{H}) = -\mathbf{H}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \tag{2.10}$$

So, in one-dimensional case with scalar value f as the Fischer information matrix, the standard deviation of the MLE is $1/\sqrt{f}$, and the confidence interval is of the form

$$\mathrm{P}(\hat{\theta} - \frac{\xi}{\sqrt{f}} \leq \theta \leq \hat{\theta} + \frac{\xi}{\sqrt{f}}) = p. \tag{2.11}$$

*Actually, in frequentist sense the parameter value is an unknown but constant value, and probability is not meaningful for it. The interval should be formulated using statistics as random variable, $T := t(\boldsymbol{Y})$. Still, in practice the interpretation is more or less the same, and in Bayesian concept it is allowed to speak about the probability of the parameter.

The coefficient ξ depends on the selected confidence level p. The ξ is selected so that the probability in standard normal pdf $\phi(\cdot)$ from $-\xi$ to ξ is p,

$$\int_{-\xi}^{\xi} \phi(x)dx = p. \tag{2.12}$$

For 95 % CI ($p = 0.95$) this value is 1.96, and similarly 2.58 for 99 % CI. To be exact, Eq. (2.13) with ξ from normal distribution is only the asymptotic result. If the probability model actually is normal distribution, the ξ-values should be taken from the Student's t-distribution with $n - 1$ degrees of freedom. The difference is not large, in practice it is something to be taken into account if sample size is, say, less than 10. Example of normal and t-distributions are shown in Fig. 2.6.

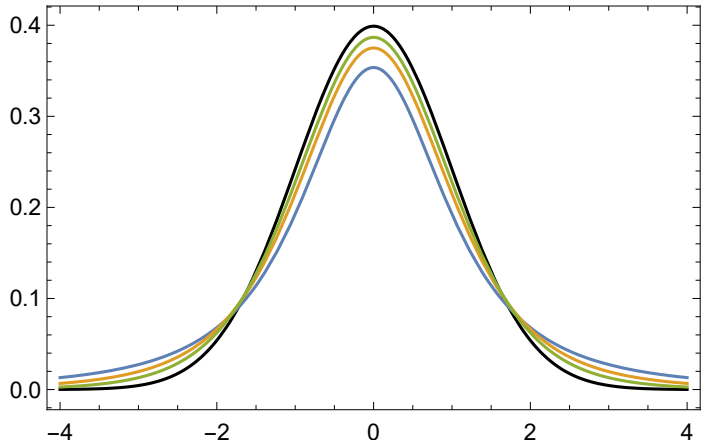


Figure 2.6: Standard normal distribution (black) and t-distribution with 2 (blue), 4 (orange), and 8 (green) degrees of freedom.

Confidence interval for mean

Mean $\overline{y}$ is the most common statistics. With normal distribution as the model, it is the MLE for the expected value, but the same is true for many other (symmetric) distributions and their location parameters. And, due to the asymptotic behavior of the mean, normal distribution is at least its asymptotic distribution.

As we know that the standard deviation of mean is $s/\sqrt{n}$, following Eq. (2.11) the (asymptotic) CI for mean around the unkown expectancy μ is

$$\mathrm{P}(\overline{y} - \xi\frac{s}{\sqrt{n}} \leq \mu \leq \overline{y} + \xi\frac{s}{\sqrt{n}}) = p. \tag{2.13}$$

On the asymptotic distribution of MLE

The asymptotic result in Eq. (2.7) is extremely important in statistical inference, since it provides tools that can be used to construct asymptotic confidence intervals

and tests for any ML estimate. However, one should remember that the result is true only *asymptotically* when the amount of data goes to infinity. In many cases the asymptotic result is 'accurate enough' quite fast, typically with some tens of observations. However, one should be careful using the asymptotic results with small amounts of data, and especially with estimates from limited range (e.g., $0 \leq \theta \leq 1$) having values close to their limits.

As an example, there is the asymptotic distribution and the 95 % confidence interval shown in the case with three samples from exponential distribution having the rate parameter $\lambda = 1/4$, see Fig. 2.7. One can see how the asymptotic distribution for the MLE extends to the negative side, although exponential distribution is defined only for positive rate parameters. Also the 95% confidence interval extends below zero. In the same Fig. 2.7 there is also the 'exact distribution' of the MLE, received via simulation. The exact distribution is clearly non-symmetric, and respects the limit of zero for the possible values of the parameter.

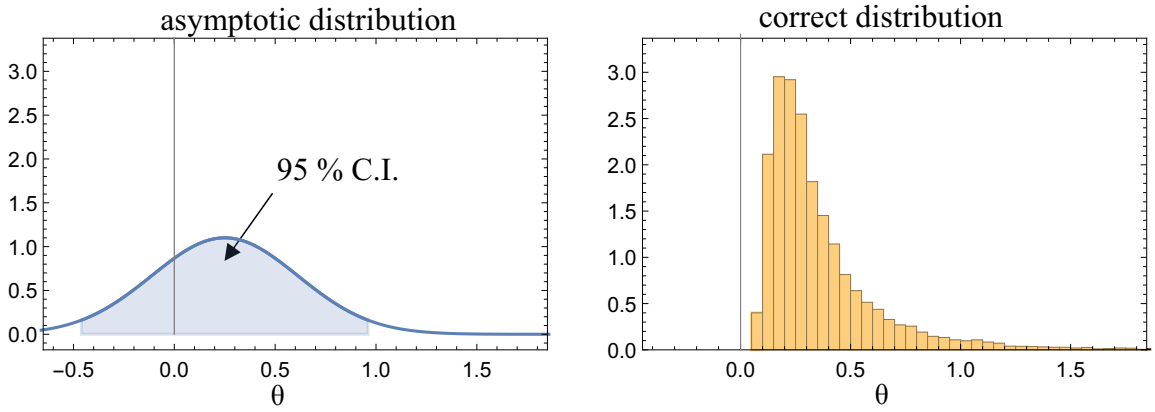


Figure 2.7: In the left, the asymptotic distribution of the MLE for the rate parameter in exponential distribution for random sample with $n = 3$. The correct parameter value, used in sampling, was $1/4$. In the right, the correct distribution of the rate parameter MLE directly from simulation.

Continuing the same example, if we increase our sample size to $n = 30$, the asymptotic result is much better, see Fig. 2.8. The 95 % CI nicely stays above zero, and the exact distribution is already almost symmetric.

2.2.2 Tests

With statistical tests we can check the likelihood of our hypothesis against the observed data, and make conclusions based on quantitative results. For tests we need suitably constructed test statistics $t(\boldsymbol{y})$ and a hypothesis, the so-called null hypothesis H_0 (*nollahypoteesi*). The null hypothesis needs to define the probability model for the test statistics, i.e., we must be know how $T|H_0$ is distributed.

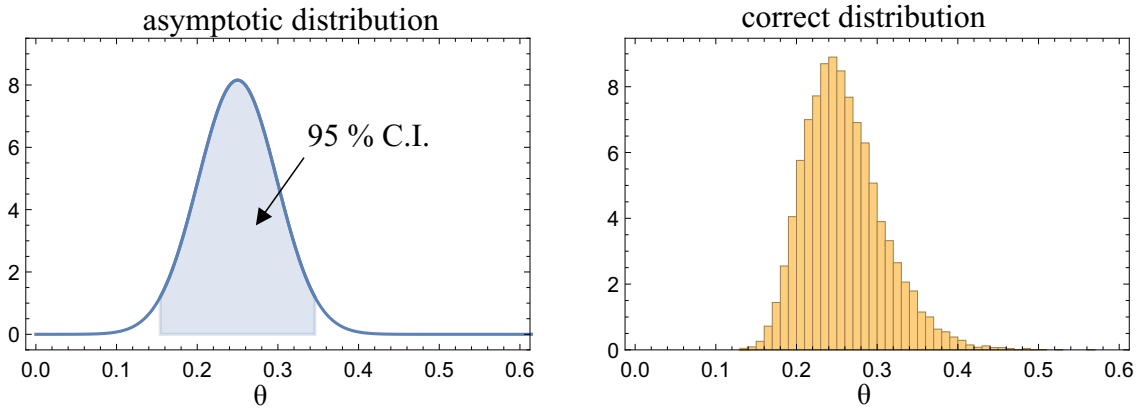


Figure 2.8: Same as in Fig. 2.7, but the sample is $n = 30$.

If the data shows that our null hypothesis is very unlikely to be true, then we conclude that the alternative hypothesis H_1 (*vastahypoteesi*) seems more plausible. While the null hypothesis defines either one point in the parameter space, or at least some (small) set of parameters, the alternative hypothesis is its complement and does not define single value for the parameter, rather a single value that the parameter is not. For example, one could test with the mean from normally distributed data if (H_0) the $\mu = c$ or, (H_1) the $\mu \neq c$.

p-value of a test

The principle of statistical tests lies in the distribution of $T|H_0$ and in the likelihood of observed t. As said, we must know the pdf of $T|H_0$, i.e., $\mathrm{f}_{T|H_0}(t)$. With that knowledge we can calculate the probability of observing *as extreme value* of T as we have, or *even more extreme*, on the condition that H_0 is true. We return to the question of 'even more extreme' later, but for now we just formulate that

$$\begin{aligned} \mathrm{P}(T \text{ more extreme as } t|H_0) &= \int_{t \text{ more extreme}} \mathrm{f}_{T|H_0}(x)dx \\ &= 1 - \int_{t \text{ less extreme}} \mathrm{f}_{T|H_0}(x)dx = p. \end{aligned} \quad (2.14)$$

Now, the philosophy is that if it is not that unlikely to observe such values of the statistic t when H_0 is true, we should not reject it. We do not say that H_0 is proven, but that there is no evidence that it should be rejected. If the p-value is very small it is quite unlikely to observe such value of t if H_0 is true. In that case we have two possibilities — either H_0 is not true, or a very unlikely event has happened. When the p-value is small enough, we tend to rule out the very unlikely event and say that H_0 is rejected and H_1 is accepted with a certain p-value. See Fig. 2.9 for an example of test statistics where $T|H_0$ obeys χ^2-distribution, and the corresponding p-value.

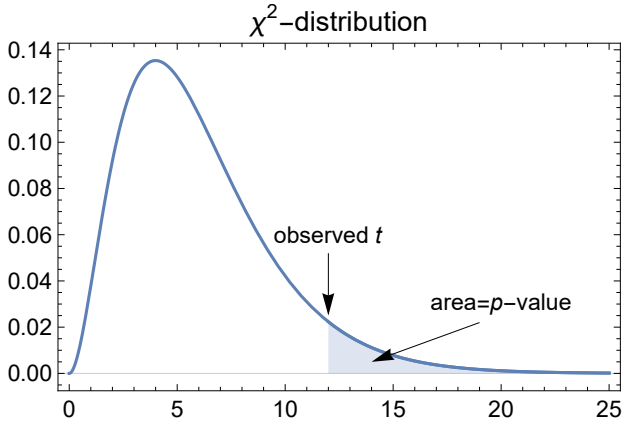


Figure 2.9: χ^2-distribution, observed test statistics t and the area corresponding to p-value of a one-tailed test.

A certain conservative attitude is adopted with testing, and typical p-values where H_0 is rejected are 0.10, 0.05 and 0.01. In times before computers it was common that just these three p-values were used, because tabulated values were looked up from tables containing these three cases. Nowadays one can as easily compute the exact p-value for the test and report that.

Limitations of statistical tests

With statistical tests one needs to understand their capabilities and limitations. Tests are quite good to quantify observed facts when there is moderate amount of data in hand. With just a few observations the uncertainty is usually so large, that it is very hard to reject H_0. With large amount of data the problem is the opposite — it is quite easy to reject H_0. This is because the test usually states that there is evidence of deviation from H_0. What the test does not quantify that well is how large the deviation from H_0 is, and especially, does it have any practical consequences. For example, if one tests the correlation between two variables, H_0 is that there is no correlation, $\rho = 0$. Often with real-world data, the parameter ρ might deviate slightly from zero. When the number of observations increase, the test becomes stronger and picks up smaller and smaller differences from zero. Therefore, with large data it is easy to conclude that the correlation is not zero, and thus there is correlation, but the amount of correlation can be very small and not significant within the physical/real-world context behind the data. That said, statistical tests are very useful with moderate number of observations and with moderate deviations from H_0 when it is difficult to see without statistics if the deviation is 'unusual' or not.

Rejection areas

We need to define what we mean in Eq. (2.14) by areas where t is 'even more extreme'. That depends on the distribution of the test statistics, and on the alternative hypothesis. First, if the test statistics can have both negative and positive values, the distribution must be symmetric around zero. This is the case, for example, if the test statistics has normal or t-distribution under H_0. If we cannot say beforehand if it is impossible to have smaller or larger values of t than assumed in H_0, our alternative hypothesis must be two-tailed (*kaksisuuntainen*), i.e., $H_0 : \theta = c$, $H_1 : \theta \neq c$. In this case (symmetric distribution, two-tailed H_1), the rejection area for test is such that

$$\mathrm{P}(T \geq \mathrm{abs}(t)|H_0) = 2\int_{\mathrm{abs}(t)}^{\infty} \mathrm{f}_{T|H_0}(x)dx = 2\int_{-\infty}^{-\mathrm{abs}(t)} \mathrm{f}_{T|H_0}(x)dx$$
$$= 1 - \int_{-\mathrm{abs}(t)}^{\mathrm{abs}(t)} \mathrm{f}_{T|H_0}(x)dx = p. \quad (2.15)$$

If we have some a priori knowledge so that we can rule out, for example, positive values of t, we have one-tailed (*yksisuuntainen*) alternative hypothesis $H_1 : \theta < c$ and the rejection area is

$$\mathrm{P}(T \leq t|H_0) = \int_{-\infty}^{t} \mathrm{f}_{T|H_0}(x)dx = 1 - \int_{t}^{\infty} \mathrm{f}_{T|H_0}(x)dx = p, \quad (2.16)$$

and in similar manner for alternative hypothesis $H_1 : \theta > c$ but with integration limits changed accordingly.

The test statistics might have distribution that is only valid for positive values, for example the χ^2 or F-distribution. These distributions are not symmetric, and we have to choose carefully the rejection area. If our statistics is close to zero and we have one-tailed H_1, the test is defined as

$$\mathrm{P}(T \leq t|H_0) = \int_{0}^{t} \mathrm{f}_{T|H_0}(x)dx = 1 - \int_{t}^{\infty} \mathrm{f}_{T|H_0}(x)dx = p. \quad (2.17)$$

With observed test statistics 'large' and with one-tailed H_1, the test is

$$\mathrm{P}(T \geq t|H_0) = \int_{t}^{\infty} \mathrm{f}_{T|H_0}(x)dx = 1 - \int_{0}^{t} \mathrm{f}_{T|H_0}(x)dx = p. \quad (2.18)$$

If we cannot rule out beforehand the small or large values of t, we must choose two-tailed test. Then, as we observe t to be either (i) close to zero or (ii) large, we choose (i) Eq. (2.17) or (ii) Eq. (2.18) and multiply the p-value in the correct equation by two to get the two-tailed p-value.

Mean tests

To list some tests, let us first consider the mean test, i.e., test for the expected value. The data is $\boldsymbol{y}$, and the statistics of interest is the mean value $\overline{y}$. The null hypothesis is of form $\mu = \mu_0$. For practical reasons we rather use the test statistics

$$t = \frac{\overline{y} - \mu_0}{s/\sqrt{n}}, \tag{2.19}$$

where s is the sample standard deviation. From Eq. (2.7) we know that the asymptotic distribution of $T|H_0$ is the standard normal distribution. We can formally say that

$$H_0 \colon \mu = \mu_0 \implies T \overset{\text{approx.}}{\sim} \mathcal{N}(0,1). \tag{2.20}$$

Actually, if we know that the distribution of data is normal, we can replace the asymptotic distribution with the exact one: $T \sim t_{n-1}$, the Student's t-distribution with $n-1$ degrees of freedom.

In Fig. 2.10 there are 10 random numbers that are sampled from $\mathcal{N}(0.1, 1)$ distribution. Our H_0 is that $\mu = \mu_0 = 0$, and that distribution is shown in subfigure a) together with the data. The test statistics t is calculated ($t \approx 1.29$) and the areas $]-\infty, -t]$ and $[t, \infty]$ drawn in subfigure b) together with the distribution of $T|H_0$, the t-distribution with 9 degrees of freedom. The p-value, i.e., the colored area in subfig b), is 0.23. Therefore, we do not have enough evidence against $H_0 : \mu = 0$ and we cannot reject that possibility, although in here the data actually comes from distribution with $\mu = 0.1$.

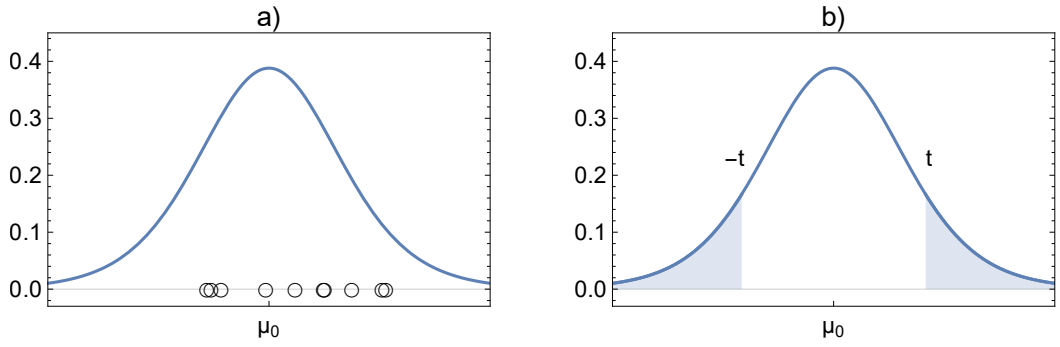


Figure 2.10: Data and H_0-distribution in left (a), observed value of t and the distribution according to H_0 in right (b).

Similar mean test can be also constructed for two samples and the difference of their mean values. One has to assume that the samples have the same distributions (expect for the location parameter) and that their variances σ_1^2 and σ_2^2, while unknown, are equal. In that case,

$$H_0 \colon \mu_1 - \mu_2 = d_0 \implies T = \frac{(\overline{y}_1 - \overline{y}_2) - d_0}{s_p\sqrt{1/n_1 + 1/n_2}} \sim t_{n_1+n_2-2}, \tag{2.21}$$

where pooled variance

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}. \tag{2.22}$$

In what follows we will shortly describe some tests, but the list is not by far complete. You will notice that almost all the distributions for test statistics are either Student's t-distribution, χ^2-distribution or F-distribution. This is simply because all these distributions are derived from normal distribution: t-distribution from the ratio of normal variable and its standard deviation, χ^2-distribution from sum of squared normal variables, and F-distribution from ratio of normal variables.

Variance tests

For variance of one normal distributed sample the test is

$$H_0 \colon \sigma^2 = \sigma_0^2 \implies T = (n-1)\frac{s^2}{\sigma_0^2} \sim \chi^2_{n-1}. \tag{2.23}$$

For two normal distributed samples the test for equal variance is

$$H_0 \colon \sigma_1^2 = \sigma_2^2 \implies T = \frac{s_1^2}{s_2^2} \sim \mathcal{F}_{n_1-1,n_2-1}, \tag{2.24}$$

and the alternative hypothesis will define the rejection area to either Eq. (2.17) or (2.18). For two-tailed test one needs to adjust the p-value to $2p$.

Correlation test

The linear correlation, the value of correlation coefficient ρ and its sample statistics $r = \mathrm{cor}(\boldsymbol{x}, \boldsymbol{y})$, can be tested against being zero. The test is

$$H_0 \colon \rho = 0 \implies T = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \sim t_{n-2}, \tag{2.25}$$

and rejection area is defined by Eq. (2.15) for two-tailed, and by Eq. (2.16) for one-tailed test.

Kolmogorov-Smirnov test

Kolmogorov-Smirnov (K-S) test is our first non-parametric test. It can be used to test if the observed distribution differs from theoretical distribution, and the test is valid for all (continuous) distributions. The test is based on the empirical cdf and the theoretical cdf. The test statistics t is defined as $t = \sqrt{n}D$, where D is the maximum difference between the two cdf's, see Fig. 2.11.

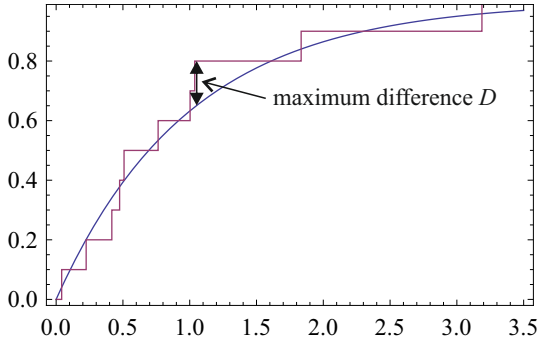


Figure 2.11: Empirical and theoretical cumulative distribution functions and the Kolmogorov-Smirnov difference D.

The K-S test is always one-tailed, and the test statistic has Kolmogorov distribution if H_0 that the sample comes from the theoretical distribution is true. The rejection area is defined as in Eq. (2.18).

There is a similar version for K-S test between two empirical distributions, check for example Wikipedia for the details.

Goodness-of-fit test

Goodness-of-fit test can be used for discrete variables. It is formulated as

$$H_0: \text{Empirical distribution obeys the theoretical one} \implies$$
$$T = n \sum_{i=1}^{n} \frac{(o_i - e_i)^2}{e_i} \sim \chi^2_{n-1-m}, \quad (2.26)$$

and large values speak against H_0 as in Eq. (2.18). The terms o_i are the observed probabilities (proportions) of class/value/category i in the sample, and terms e_i are the expected probabilities if H_0 is true. The variable m in the degrees of freedom for the χ^2-distribution is the number of unknown parameter values estimated from the data for the theoretical distribution. For example, if we want to test if the observed proportions come from uniform (discrete) distribution, we do not need to estimate any parameter values from the data, and $m = 0$.

Independence test

The same test statistics as above can be used to test the independence between two-dimensional categorical variable, i.e., proportions in two-way contingency tables (cross tabulations, *ristiintaulukko*). For this test, every observation has two properties, A and B, so the observation can be associated into one cell in the contingency table. The proportions of the associations are counted, resulting the following table

$A \setminus B$	1	...	k	Σ
1	o_{11}	...	o_{1k}	A_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
m	o_{m1}	...	o_{mk}	A_m
Σ	B_1	...	B_k	1

The expected proportions, if the two properties A and B are independent, can be estimated from the product of the marginal proportions: $e_{ij} = A_i B_j$. The test statistics is computed over all the rows and columns, and

$$H_0 \colon A \perp\!\!\!\perp B \implies T = n \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \sim \chi^2_{(m-1)(k-1)}, \tag{2.27}$$

and large values speak against H_0 as in Eq. (2.18).

Chapter 3

Linear model

3.1 Introduction

Linear model (LM, *lineaarinen malli*) or (linear) regression analysis (*regressioanalyysi*) is a family of models that is used to analyze dependence between scalar dependent variable (*selitettävä muuttuja, vastemuuttuja*) and one or more explanatory variables (*selittävä muuttuja*).

The term *regression* refers to regression towards mean, the fact that the expected value (i.e. 'mean') is the best prediction to a random variable. We construct the linear model in such a way that it actually models the expected value of the dependent variable, and the difference between the model and the observations is the 'random part' of the model.

3.1.1 Systematic part of linear model

The terminology in LM is such that the observed values of the explanatory variable $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ik})$ are collected together into $n \times k$ data matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \ldots & x_{1k} \\ & \ddots & \\ x_{n1} & \ldots & x_{nk} \end{bmatrix}, \tag{3.1}$$

and the observed values of the dependent variable are collected to vector $\boldsymbol{y} = (y_1, \ldots, y_n)$. Linear regression refers to a model where the functionality between the explanatory and the dependent variables is linear. With a common choice of symbol $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ for the regression coefficients, i.e., the linear function between the variables, we end up with

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta}, \tag{3.2}$$

or, for a single observation i:

$$y_i = \boldsymbol{x}_{i\cdot} \cdot \boldsymbol{\beta} = \beta_1 x_{i1} + \cdots + \beta_k x_{ik}. \tag{3.3}$$

The equations above describe the systematic part of LM, there is no random component included yet.

3.1.2 Random part of the linear model

The systematic part of LM does not say anything about random variables or deviations between the model and reality. For that we need to introduce randomness into LM. That is done via the residuals (*residuaali, jäännös*). The idea is that the systematic part of the model is described perfectly by Eq. (3.2), but the randomness is added to the equation and that explains the errors between the model and the observations. With residual ϵ (random variable) this means that LM for one observation is

$$Y_i = \boldsymbol{x}_{i\cdot} \cdot \boldsymbol{\beta} + \epsilon_i = \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i, \tag{3.4}$$

or, in matrix form for all the observations

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{3.5}$$

i.e.

$$\begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ & \ddots & \\ x_{n1} & \dots & x_{nk} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}. \tag{3.6}$$

Figure 3.1 shows an example of one-dimensional linear model and Fig. 3.2 for two-dimensional model.

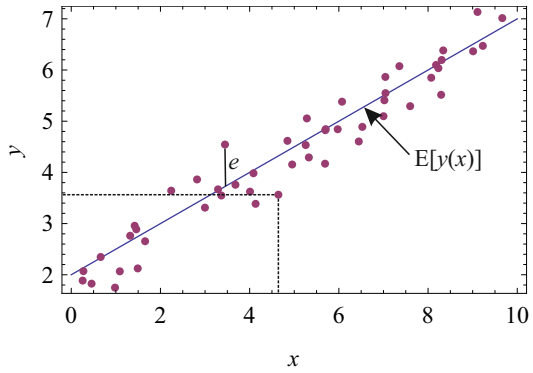


Figure 3.1: Concepts in regression model — data x, dependent variable y, regression model $\hat{y} = \mathrm{E}[y(x)]$, and residual ϵ.

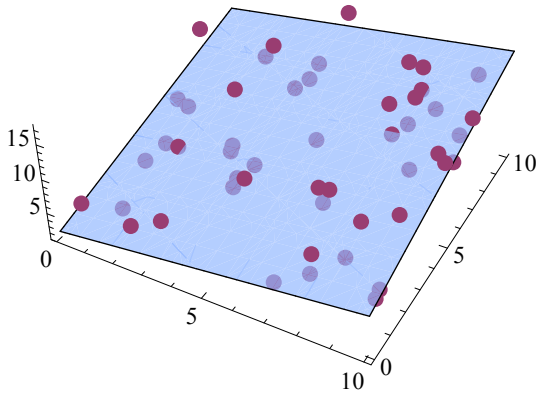


Figure 3.2: A linear model with two explanatory variables.

3.1.3 Assumptions for linear model

Some assumption are needed to make LM statistically and technically valid. The so-called standard assumption are:

1. Explanatory variable is non-random. There are ways to go around this assumption, and this is more important in principle than in practice. Anyway, it should be noted that LM in its basic form does not take possible errors in $\mathbf{X}$ into account in any way.*

2. Explanatory variables are not (completely) linearly dependent on each other. There cannot be an explanatory variable whose values can be computed as a linear combination from the other explanatory variables. This will indicate that, for example, the correlation coefficient ρ between any two explanatory variables cannot have values -1 or 1. This is mostly a technical assumption, since if violated, the matrix $\mathbf{X}^T\mathbf{X}$ is singular and cannot be inverted. The inversion will be needed in the estimation of LM as you will see later. We can run into numerical problems also in cases where an explanatory variable is *almost* a linear combination of the other variables.

3. The expected value of each residual is zero, i.e., $\mathrm{E}(\epsilon_i) = 0\ \forall i$, or $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$. This is a vital assumption, since it guarantees that we are modeling the expected value of Y with the systematic part of our model, because now

$$\begin{aligned}\mathrm{E}(Y_i) = \mathrm{E}(\beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \epsilon_i) &= \beta_1 x_{i1} + \ldots + \beta_k x_{ik} + \mathrm{E}(\epsilon_i) \\ &= \beta_1 x_{i1} + \ldots + \beta_k x_{ik}. \quad (3.7)\end{aligned}$$

4. The variance of the residuals are constant, i.e., $\mathrm{var}(\epsilon_i) = \sigma^2\ \forall i$, or $\mathrm{var}(\boldsymbol{\epsilon}) = \sigma^2\mathbf{1}$. This is the so-called homoscedasticity assumption. In many cases where this

*For LM when there is significant measurement error in explanatory variables, see *Total least squares* from, e.g., `https://en.wikipedia.org/wiki/Total_least_squares`.

is initially not true, it is possible to weight the samples so that this assumption becomes true for the weighted model (dealt later in this chapter). For the dependent variable this indicates that $\mathrm{var}(Y_i) = \sigma^2$.

5. There is no correlation/covariance between the residuals, i.e., $\mathrm{cov}(\epsilon_i, \epsilon_j) = 0 \; \forall i \neq j$ or $\mathrm{cov}(\boldsymbol{\epsilon}) = \sigma^2 \mathbf{I}_n$. The lack of (auto)correlation rules out time-series from standard linear model.

You may notice that there are no assumptions about the normality of the residuals. These are not needed for LM to be 'valid' in statistical sense. However, if normality can be assumed, it will allow us to do certain statistical inference dealing with confidence intervals, tests, etc. But, even in cases where normality is not assumed per se, results derived from normal assumption are usually asymptotically valid. The normal assumption states that

$$\boldsymbol{\epsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \boldsymbol{I}_n), \tag{3.8}$$

and thus

$$\boldsymbol{Y} \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_n), \tag{3.9}$$

3.1.4 Linear model is linear with respect to model coefficients

An important detail to notice with LM and its formulation (e.g. Eq. (3.5)) is that only the functional dependence between the data and the dependent value needs to be linear, i.e., of form $\mathbf{X}\boldsymbol{\beta}$. The data itself can be transformed by any linear or nonlinear function. The justification is simple — if we want to use $\mathrm{f}(x_i)$ where f is any function in LM instead of x_i, we can just introduce new variable $x_i^* = \mathrm{f}(x_i)$ into matrix $\mathbf{X}$. More generally, $\boldsymbol{Y} = \mathrm{f}(\mathbf{X})\boldsymbol{\beta} + \boldsymbol{\epsilon} = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\epsilon}$. In Fig. 3.3 there are examples of one-dimensional LM's where the dependence is through x^2 or $\log(x)$.

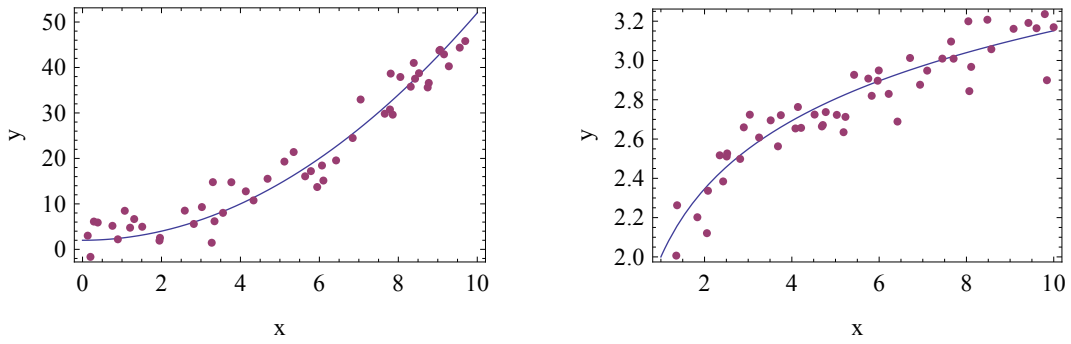


Figure 3.3: Examples of two linear models with one explanatory variable.

Constant term

One application to the above is the constant term (*vakiotermi*) in LM, β_0. You will often see models in the form of

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i, \tag{3.10}$$

but this is a simple transformation to data matrix. If you introduce constant value of 1 as the first variable, you will end up with previous equation. Thus, constant term is introduced to LM by constructing data matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1k} \\ \vdots & & \ddots & \\ 1 & x_{n1} & \dots & x_{nk} \end{bmatrix}. \tag{3.11}$$

With the constant term it is a popular convention that the coefficients are re-numbered[†] from 0 to k, instead of 1 to $k+1$.

Interaction term

With multivariate linear model a common 'derived variable' is the so-called interaction term (*yhteisvaikutustermi*), i.e., variable of type $x_j x_l$. With the interaction term present the (hyper)planes from LM with only linear x_j's transforms into models that are not (hyper)planes with respect to original x_j's. In Fig. 3.4 there are examples of two-dimensional LM's where the dependence is not in the form of (hyper)plane as respect to x_1 and x_2.

Transformation into linear

The fact that the explanatory variables can be transformed can also be applied to the whole model equation and the dependent variable Y_i, but with certain conditions. Let us have an example of model where the systematic part is $y_i = \beta_0 x_{i1}^{\beta_1} \cdots x_{ik}^{\beta_k}$. By applying logarithm function to both sides of the equation, we end up with new dependent and explanatory variables: $y_i^* = \log(y_i) = \log(\beta_0) + \beta_1 \log(x_{i1}) + \cdots + \beta_k \log(x_{ik}) = \beta_0^* + \beta_1 x_{i1}^* + \cdots + \beta_k x_{ik}^*$. The transformed model is linear.

The one important thing to consider in transformations is that it does not only transform the systematic part, but the residuals also. With the example above, residuals must be additive to the transformed model. That implies that they were multiplicative in the original one, i.e., $Y_i = \beta_0 x_{i1}^{\beta_1} \cdots x_{ik}^{\beta_k} \epsilon_i$. If this is not reasonable model for the residuals, the transformed model violates the LM form.

[†]Please note that in the next sections, the formulae for the confidence intervals, parameter tests, model performance etc. use the symbol k as the number of coefficients in the model. If you include a constant term and re-number from 0 to k, you will actually have $k+1$ coefficients and should revise the formulae accordingly.

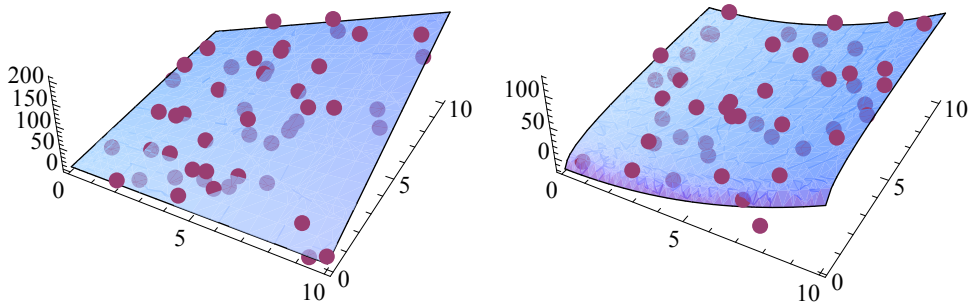


Figure 3.4: Examples of two linear models with two explanatory variables. In left, dependence is of form $\beta_0 + \beta_1 x_1 + \beta_2 x_1 x_2$, and in right of form $\beta_0 + \beta_1 x_1^2 + \beta_2 \log(x_2)$.

Categorical variables

Categorical variables (*luokittelumuuttujat*, i.e., discrete variables with reasonably small number of possible values) can be used in linear models, although there should usually be continuous variables also present in the model. Model with only categorical variables can be analyzed better as a special case of LM using analysis-of-variance (ANOVA) methods. The recipe for including categorical variables is again to encode the categories to one or more explanatory variables.

Let us have categorical variable c that has $p+1$ different outcomes (categories), coded here with numbers $0, 1, \ldots, p$. We can introduce a set of p new variables $\{g_{i1}, \ldots, g_{ip}\}$ into $\mathbf{X}$. We need one 'reference category', for example the case $c = 0$. With the reference category we have the set as $\{0, \ldots, 0\}$. With the case $c = 1$ we have $\{1, 0, \ldots, 0\}$, with $c = 2$, $\{0, 1, 0, \ldots, 0\}$ etc., and finally with $c = p$, $\{0, \ldots, 0, 1\}$. Now the augmented data matrix row for, e.g., observation with $c = 2$ and $p+1 = 4$ would be $\boldsymbol{x}_i^* = (0, 1, 0, x_{i1}, \ldots, x_{ik})$.

With the data matrix augmented with new variables coded from the categorical variable, the systematic part of ML is

$$y_i = \beta_0 + \beta_1 g_{i1} + \ldots + \beta_p g_{ip} + \beta_{(p+1)} x_{i1} + \ldots + \beta_{(p+k)} x_{ik}, \tag{3.12}$$

and the model can be estimated in normal manner. The additional limitation with categorical variables is that if we do variable selection or model diagnostics (see later in the chapter), the augmented variables must be dealt as a group.

The interpretation of the model with augmented variables for categories is that the constant term β_0 is now related to the case with $c = 0$. The regression coefficients of the new categorical variable, β_j's, estimates the difference in y when moving from the reference class to class $c = j$. There is a technical reason behind the reference

class having zeros for all the new variables — otherwise the 'constant' variable 1 would be sum of the new variables, and that would violate the beforementioned assumption 2 with ML.

3.2 Estimation of the linear model

The first task in LM analysis is to estimate the coefficients $\boldsymbol{\beta}$ for the model. The LM is implicitly assumed to refer to a case where the L^2-norm between model and observations is minimized. This combination of LM and minimization of L^2-norm is called the method of *least squares* or *ordinary least squares* (OLS, *pienimmän neliösumman menetelmä, PNS*). With OLS the values for the coefficients can be computed analytically, which is generally not the case with non-linear models or with other than the L^2-norm.

So, in OLS we want to minimize the sum of squared residuals (or errors, SSE):

$$SSE = \sum_i^n (y_i - \beta_1 x_{i1} - \ldots - \beta_k x_{ik})^2 = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|^2 . \quad (3.13)$$

The solution to the minimization above can be derived by solving the root of its derivative. Without details, it will give us the so-called normal equations (NE)

$$\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\boldsymbol{y}. \quad (3.14)$$

The solution to NE is the estimate to the model, $\boldsymbol{b} = \hat{\boldsymbol{\beta}}$:

$$\boldsymbol{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}. \quad (3.15)$$

With the estimate $\boldsymbol{b}$ for $\boldsymbol{\beta}$ we can compute the observed residuals, $\boldsymbol{e} = \boldsymbol{y} - \mathbf{X}\boldsymbol{b}$, and again this is the estimate for the random variable $\boldsymbol{\epsilon}$. Now the SSE can be expressed with

$$SSE = \|\boldsymbol{e}\|^2 , \quad (3.16)$$

and the residual variance σ^2 (*jäännösvarianssi*) of the model can be estimated by s^2 as

$$s^2 = \frac{1}{n-k} SSE. \quad (3.17)$$

Note that to compute the OLS estimate $\boldsymbol{b}$, the matrix inversion in Eq. (3.15) can be avoided, which can be preferable with large number of variables k because the matrix to be inverted, $\mathbf{X}^T\mathbf{X}$, is a $k \times k$ matrix. The solution to NE in Eq. (3.14) can be computed with LU- or Cholesky decomposition and Gaussian elimination.

3.2.1 Properties of OLS estimate

We can derive quite easily some properties of the OLS estimate $\boldsymbol{b}$. Most importantly, it holds that

$$\mathrm{E}(\boldsymbol{b}) = \boldsymbol{\beta}, \tag{3.18}$$

and

$$\mathrm{cov}(\boldsymbol{b}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}. \tag{3.19}$$

These properties do not require any assumption of normal distribution for the residuals $\boldsymbol{\epsilon}$. However, if we assume that residuals follow normal distribution we can show that the OLS estimate is also the maximum likelihood estimate, and that

$$\boldsymbol{b} \sim \mathcal{N}_n(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}) \tag{3.20}$$

3.2.2 Weighted linear model

Weighed LM comes up in cases where the variance of residuals or of the dependent variable is not constant. The observations where variance is small should influence 'more' to the estimate, they should 'weight' more. This means that instead of $\mathrm{var}(\epsilon_i) = \sigma^2$ we have $\mathrm{var}(\epsilon_i) = \sigma^2/w_i$, where w_i is the weight of the observation. In matrix formulation this is written as

$$\mathrm{cov}(\boldsymbol{\epsilon}) = \sigma^2\mathbf{V}, \tag{3.21}$$

where $\mathbf{V}$ is a diagonal matrix $\lceil 1/w_1 \cdots 1/w_n \rfloor$.

The estimation of weighted LM is derived with the help of (Cholesky) decomposition $\mathbf{V} = \mathbf{C}\mathbf{C}^T$. Multiplying LM by $\mathbf{C}^{-1}$ from left we get

$$\mathbf{C}^{-1}\boldsymbol{y} = \mathbf{C}^{-1}\mathbf{X}\boldsymbol{\beta} + \mathbf{C}^{-1}\boldsymbol{\epsilon}, \tag{3.22}$$

which can be written as $\boldsymbol{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\epsilon}^*$. It is easy to see that

$$\mathrm{E}(\boldsymbol{\epsilon}^*) = \mathbf{C}^{-1}\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0} \tag{3.23}$$

and

$$\mathrm{cov}(\boldsymbol{\epsilon}^*) = \mathbf{C}^{-1}\mathrm{cov}(\boldsymbol{\epsilon})(\mathbf{C}^{-1})^T = \sigma^2\mathbf{C}^{-1}\mathbf{C}\mathbf{C}^T(\mathbf{C}^T)^{-1} = \sigma^2\mathbf{I}_n, \tag{3.24}$$

so that the transformed model is a regular LM. For the estimation of $\boldsymbol{\beta}$ one does not even need to form the decomposition, since

$$\begin{aligned}\boldsymbol{b} = \left((\mathbf{C}^{-1}\mathbf{X})^T\mathbf{C}^{-1}\mathbf{X}\right)^{-1}(\mathbf{C}^{-1}\mathbf{X})^T\,\mathbf{C}^{-1}\boldsymbol{y} &= \left(\mathbf{X}^T(\mathbf{C}\mathbf{C}^T)^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{C}\mathbf{C}^T)^{-1}\boldsymbol{y} \\ &= (\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}^{-1}\boldsymbol{y}. \end{aligned} \tag{3.25}$$

This equation above means that the weighted model can be estimated quite similarly as the normal LM, only including an extra weight matrix $\mathbf{V}$. Actually, the procedure is valid for any positive definitive $\mathbf{V}$, therefore it is called the generalized linear model and it allows also covariance between the residuals.

3.3 Diagnostics of linear model

The estimation of linear model, as seen above, is not too complicated. Main interests for researcher with LM is usually the diagnostics for the model. These include checks regarding the model assumptions, selection of variables, confidence intervals etc.

3.3.1 Validity of model assumptions

The assumptions behind LM were introduced in Sec. 3.1.3. The validity of the assumptions can be assessed with the observed residuals of the model

$$\boldsymbol{e} = \boldsymbol{y} - \mathbf{X}\boldsymbol{b}, \tag{3.26}$$

or even better, with standardized (i.e., studentized) residuals r_i:

$$r_i = \frac{e_i}{s\sqrt{1-p_{ii}}}, \tag{3.27}$$

where s is the estimate of the residual standard deviation, see Eq. (3.17). The term p_{ii} is part of the covariance matrix of the observed residuals:

$$p_{ii} \text{ is } [\mathbf{P}]_{ii} \text{ in } \quad \mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T. \tag{3.28}$$

With weighted model where v_{ii} are elements $[\mathbf{V}]_{ii}$ in $\mathrm{cov}(\boldsymbol{\epsilon}) = \mathbf{V}$, the standardized residuals are

$$r_i = \frac{e_i}{\sqrt{v_{ii}}\sqrt{1-p_{ii}}}. \tag{3.29}$$

With residuals, the best way to study the validity of different assumptions is to draw figure(s) of (standardized) residuals against explanatory variables, or against predicted response $\hat{\boldsymbol{y}} = \mathbf{X}\boldsymbol{b}$.

Model is unbiased

The first assumption to check with the model is the assumption 3 in Sec. 3.1.3, which says that the expected value of residuals should be zero, $\mathrm{E}(\boldsymbol{\epsilon}) = \mathbf{0}$. As the observed residuals should estimate the theoretical ones, the (standardized) residuals should have a mean value of zero. If the mean of the observed residuals is not zero, there are missing variables in the model, or the data cannot be explained with a linear model.

An example is shown in Fig. 3.5. The data is produced from $y = x^2 + \epsilon$, and two models are fitted. First model is $y = \beta_1 x$, and the second is the correct one, $y = \beta_1 x^2$. This can be seen in the residual plot, where the residuals from $y = \beta_1 x$ are clearly biased with a nonzero mean value, at least locally (see range from 0 to 4, and then from 4 to 5). Residuals from $y = \beta_1 x^2$ show random, non-systematic variation around zero, as is expected if the assumptions of LM are valid.

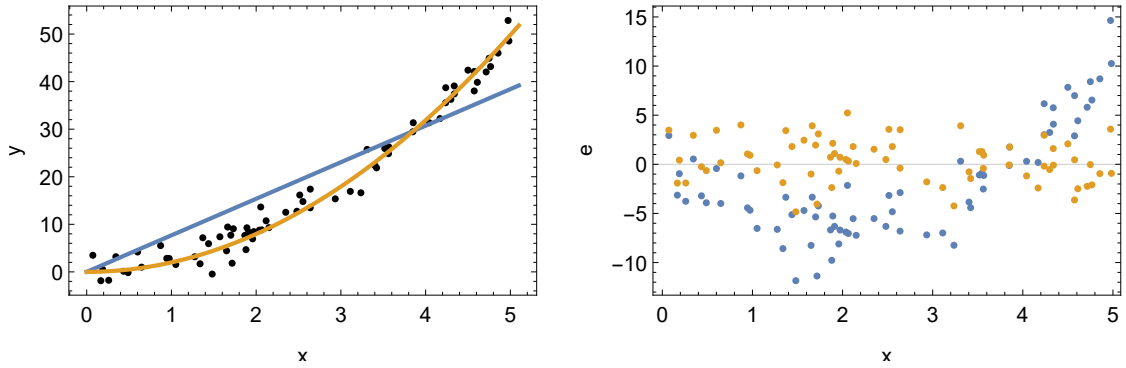


Figure 3.5: Observations and two linear models on left, and their residuals on right. Blue color is for model $y = \beta_1 x$, and orange color for $y = \beta_1 x^2$.

Residuals are homoscedastic

The assumption 4 in Sec. 3.1.3 says that the residuals should be homoscedastic, i.e., the variance of residuals should be constant. This can be quite reliably checked graphically from residual plots. In Fig. 3.6 we show example of homoscedastic and heteroscedastic residuals. In many cases the heteroscedasticity can be removed by choosing suitable weighting for the observations, i.e. modeling out the trends in variance.

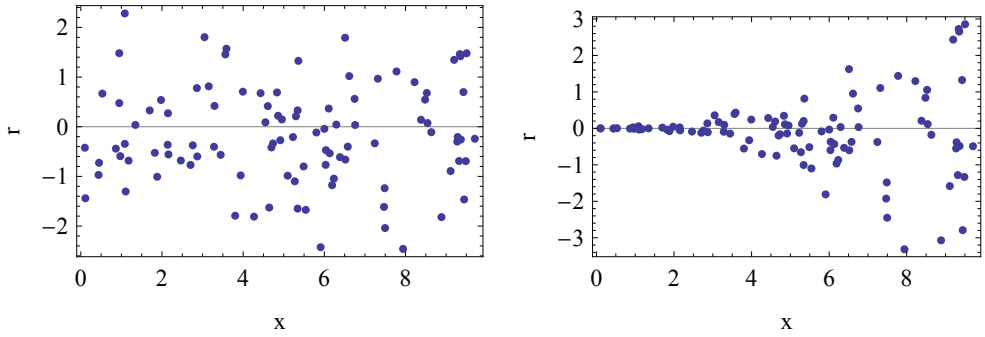


Figure 3.6: Example of homoscedastic residuals (on the left) and heteroscedastic residuals (on the right).

Residuals are normal-distributed

The assumptions 1, 2, and 5 from Sec. 3.1.3 cannot be verified from residual plots. The first one (explanatory variable is non-random) requires background information from the observation event and the physics behind the data. The second one (explanatory variables not linearly dependent) is seen as difficulties in the numerical estimation of the model. The validity of the assumption 5 (no correlation between the residuals) can be seen from residuals, but without further information

about the process it is not possible to distinguish that effect from the possible bias resulting from selecting wrong variables to the model.

The 'extra' assumption about normality, however, can be tested from the residuals. If the residuals seem to follow normal distribution, all the tests and confidence intervals regarding LM are more reliable. There are special tests for normality, e.g., Saphiro-Wilk or Anderson-Darling, but one graphical analysis tool is the so-called quantile-quantile (Q-Q) plot.

The Q-Q-plot is drawn so that the theoretical quantiles of the residuals are plotted against the residuals. Let us first sort the (standardized) residuals so that $\boldsymbol{e}_{[]} = (e_{[1]} \leq e_{[2]} \leq \ldots \leq e_{[n]})$. Then, we form the corresponding empirical cumulative distribution values $\boldsymbol{c} = (\mathbf{1}/(n+1), 2/(n+1), \ldots, n/(n+1))$. The theoretical quantiles are now computed with the inverse cumulative distribution function of standard normal distribution from the c_i's as $t_i = \mathrm{F}^{-1}(c_i)$. Finally, the pairs $(t_i, e_{[i]})$ are plotted as in Fig. 3.7.

If the data is from normal distribution, the pairs should lie approximately in a $y = x$ line in the plot. Large deviations from the line is a sign of non-normal distribution.

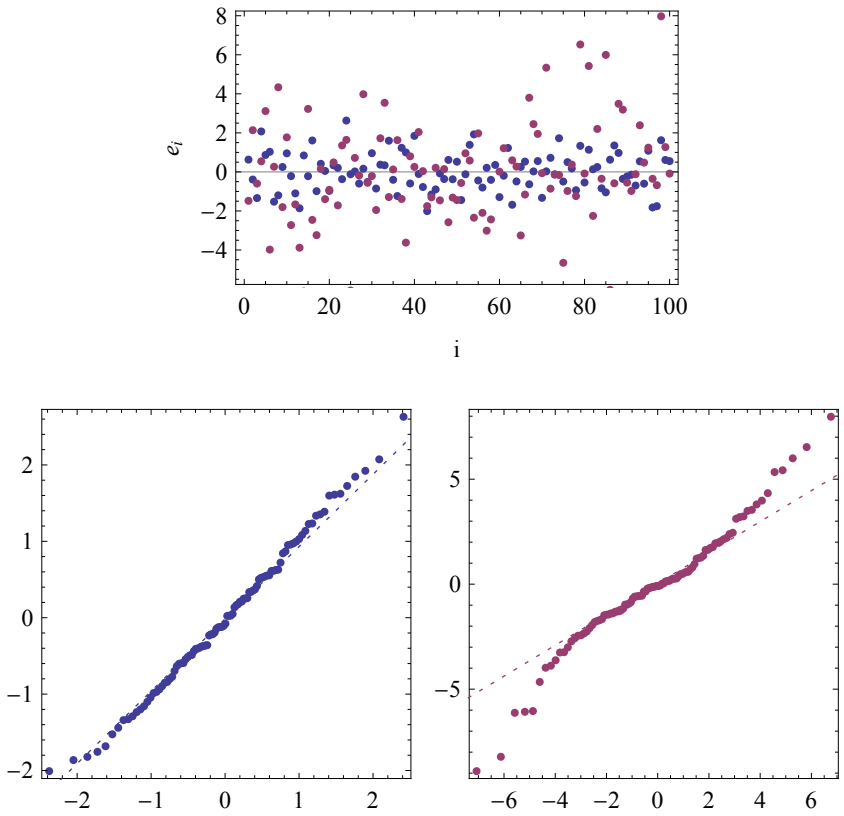


Figure 3.7: Residuals that are normally (blue) or non-normally (red) distributed in the top, and their Q-Q-plots in the bottom.

3.3.2 Model performance

The overall performance of LM is generally measured from the amount the observations deviate from the model, and that is measured by the observed sum of squared residuals (*residuaalineliösumma*), SSE

$$SSE = \boldsymbol{e} \cdot \boldsymbol{e} = \|\boldsymbol{e}\|^2 = \sum_i^n e_i^2 = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{b}\|^2 = \sum_i^n (y_i - \boldsymbol{x}_i \cdot \boldsymbol{b})^2, \tag{3.30}$$

or by the observed residual variance $s^2 = SSE/(n-k)$, where k is the number of parameters in the model. The smaller the SSE, the better the model fits to observations.

The SSE does not take into account the general variability of the dependent variable Y, only the amount of variability around the model. Therefore the coefficient of determination R^2 (*selitysaste*) is preferred, because it relates the residual variance to the total variance. The coefficient of determination is defined as

$$R^2 = 1 - \frac{SSE}{SST}, \tag{3.31}$$

where the sum of squares total (*kokonaisneliösumma*) is

$$SST = \sum_i^n (y_i - \overline{y})^2 = \boldsymbol{y} \cdot \boldsymbol{y} - n\overline{y}^2 \tag{3.32}$$

The R^2 is always between 0 and 1, and can be said to be the fraction of the variance explained by the model. For that reason, R^2 is often given in per cents.

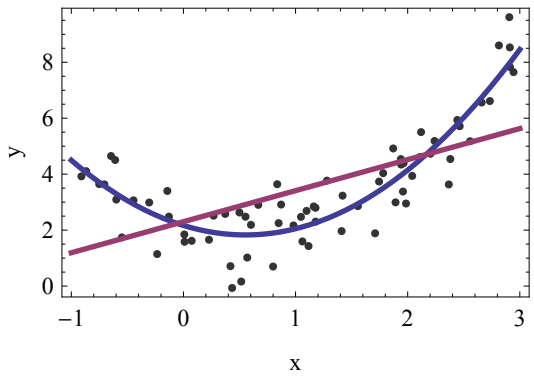


Figure 3.8: Observations and two fitted models. Red line is for model $y = \beta_0 + \beta_1 x$ and blue line for $y = \beta_0 + \beta_1 x + \beta_2 x^2$. The R^2-values for the models are 39 % (red) and 82 % (blue).

3.3.3 Variable diagnostics

If we have physical model for the observations, we know what kind of explanatory variables to include. Often, however, we need to find suitable model just by 'guessing' or trying different choices. In these cases it is very important to be able to say if certain variables are or are not important for the model. The importance can be tested.

In LM a variable x_j (which can also be any function of the 'original' x), is not important if its coefficient β_j is zero, because then it will not influence to the prediction. Of course the estimate b_j is practically never exactly zero, so we need to have a measure which tells how close it must be to zero to be unnecessary. That depends on the variability of the explanatory and the dependent variable. The test statistics t_j that can be used to study the importance of variable x_j is defined as

$$t_j = \frac{b_j}{s\sqrt{m^{ii}}}, \tag{3.33}$$

where s is the observed residual standard error, and b_j the estimate for the coefficient β_j. The factor m^{ii} is the element (i,i) from matrix $\mathbf{M}^{-1} = (\mathbf{X}^T\mathbf{X})^{-1}$.

The null hypothesis H_0 is that $\beta_j = 0$, i.e. it is not important in the model. Under H_0, the test statistics is (asymptotically) t-distributed with $n-k$ degrees of freedom, and the rejection area is defined by Eq. (2.13). The standard practice for reporting LM fit is to construct a table of its coefficient estimates, their standard deviations, test statistics, and p-values:

$$\begin{array}{c|cccc} \beta_0 & b_0 & s\sqrt{m^{00}} & b_0/s\sqrt{m^{00}} & 2\,\mathrm{F}_T(-\mathrm{abs}\left(b_0/s\sqrt{m^{00}}\right)) \\ \vdots & \vdots & & & \\ \beta_k & b_k & s\sqrt{m^{kk}} & b_k/s\sqrt{m^{kk}} & 2\,\mathrm{F}_T(-\mathrm{abs}\left(b_k/s\sqrt{m^{kk}}\right)) \end{array}$$

where F_T is the CDF of t-distribution with $n-k$ degrees of freedom.

Let us take an example. In Fig. 3.9 we have 50 observations and fitted model of form $y = \beta_0 + \beta_1 x + \beta_2 x^2$. This fit could be reported as:

	estimate	s.d.	test statistics	p-value
β_0	1.84	0.157	11.7	1.46×10^{-15}
β_1	1.36	0.246	5.53	1.4010^{-6}
β_2	-0.0790	0.107	-0.738	0.464

The conclusion of the report is that the p-value for coefficient β_2 is large, much larger than e.g. 5 %. The H_0 stating that $\beta_2 = 0$ cannot be rejected. Because $\beta_2 = 0$, the variable x^2 is unnecessary in the model and should be removed. A new model of $y = \beta_0 + \beta_1 x$ should be fitted.

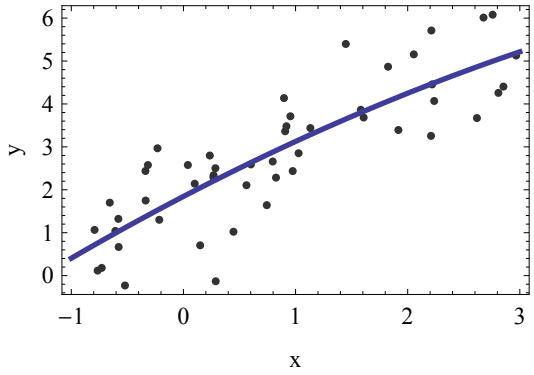


Figure 3.9: Observations and fit $y = 1.84 + 1.36x - 0.0790x^2$.

Confidence regions and the distribution of the estimated coefficients

Following from previous tests we can also construct confidence intervals for single variables in the model, or confidence regions for multiple variables. The main result that we need is that the vector of the estimated coefficients should follow, at least approximately, the multinormal distribution:

$$\hat{\boldsymbol{\beta}} \overset{\text{approx.}}{\sim} \mathcal{N}_k\left(\boldsymbol{b}, s^2(\mathbf{X}^T\mathbf{X})^{-1}\right) \tag{3.34}$$

Confidence intervals for individual coefficients can be constructed using this relation. Confidence regions for multiple coefficients will be (hyper)ellipsoids due to the properties of multinormal distribution (discussed later in Sec. 6).

The covariance matrix of the coefficient estimate $\mathbf{C} = \text{cov}(\hat{\boldsymbol{\beta}}) = s^2(\mathbf{X}^T\mathbf{X})^{-1}$ is interesting as such for diagnostic purposes. Or rather, correlation matrix $\boldsymbol{\Sigma}$ with elements

$$[\boldsymbol{\Sigma}]_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}}\sqrt{C_{jj}}} \tag{3.35}$$

is interesting. If the cross-correlations out of the diagonal of the correlation matrix are close to zero, the variables in the model are close to being independent. Independent variables are a good thing, since they introduce explanatory power to the model that is not covered by other variables. If there are cross-correlations close to ± 1, the variables in the model are correlated. That means that they more or less 'measure the same quantity' or 'explain the same phenomena'. Usually one of the two highly cross-correlated variables should be removed from the model.

3.3.4 Model selection

Model selection is a procedure where the correct explanatory variables are not known beforehand, and decisions on the variables that are selected to the final model are based on the variable diagnostics. The selection procedure is not always very straightforward, and that is because the possible cross-correlations mentioned

above in the previous section and in Eq. (3.34). The cross-correlations are the reason that variables can be added or removed to the model only one by one, not in groups. When, for example, the variable with the largest p-value is removed from the model, the p-values of the remaining variables will change. Furthermore, the order of the least important variables might change.

There are two different procedures that can be used in automated model selection — the forward selection and the backward elimination. With small number of variable candidates in the model, all possible combinations can be checked. As the number of variable candidates increase, the number of possible combinations becomes too large for every combination to be computed. Search methods have to be incorporated. In forward selection the best possible single variable is added to the model at one round, and this is continued. In backward elimination one starts from the full model, i.e. from the model with all the possible variables. In each round the worst variable is removed. The ranking of variables is based on their p-values. The bidirectional elimination is a combination of the forward- and backward methods.

Selection criteria

We can have competing models either by manual selection of a few sets of variables, or as the result from the model selection tree. A quantitative measure to compare different models as whole is needed to select the best models from the possible ones. The coefficient of determination R^2 could seem as a possible measure between the models, but it has one unwanted property. If you have set of variables A, and you add one variable x_j, the R^2 for the latter model is always as large or larger as for the former model. In another words, new variable cannot add 'negative' explanatory power, it always contributes positively to R^2. Only models with exactly the same number of variables can be compared fairly using R^2.

Therefore, different measures of the 'goodness-of-fit' have been developed that take into account the number of explanatory variables that is used to reach certain level of R^2. In one way or another, there is a 'penalty' from adding more variables. The most important model selection criteria are adjusted R^2 (R^2_{adj}), Akaike Information Criterion (AIC), and Bayesian Information Criterion (BIC). These are defined as:

$$R^2_{adj} = R^2 - (1 - R^2)\frac{k}{n-k} \tag{3.36}$$

$$AIC = n \log\left(\frac{SSE}{n}\right) + 2k \tag{3.37}$$

$$BIC = n \log\left(\frac{SSE}{n}\right) + \log(n)k \tag{3.38}$$

Large values for R^2_{adj} are 'good', while for AIC and BIC small values are searched for. The three different criteria 'punish' a bit differently from adding variables, but all are quite good in practice. The BIC is perhaps commonly preferred over the others.

Chapter 4

Nonlinear model

4.1 Introduction

Nonlinear model (NLM, *epälineaarinen malli*) is an extension to linear model where the systematic part of the model is no longer a linear function $\mathbf{X}\boldsymbol{\beta}$. Generally, NLM is of form

$$Y_i = \mathrm{f}(x_{i1}, \ldots, x_{ik}; \beta_1, \ldots, \beta_p) + \epsilon_i = \mathrm{f}(\boldsymbol{x}_i; \boldsymbol{\beta}) + \epsilon_i \tag{4.1}$$

for $i = 1, \ldots, n$ observations, k variables and p parameters. Note that for LM $k = p$, but this is not requirement in NLM. In vector form the NLM is

$$\boldsymbol{Y} = \mathbf{f}(\mathbf{X}; \boldsymbol{\beta}) + \boldsymbol{\epsilon}, \tag{4.2}$$

where $\boldsymbol{Y}$ is $n \times 1$, $\mathbf{X}$ is $n \times k$, $\boldsymbol{\beta}$ $p \times 1$, and $\boldsymbol{\epsilon}$ $n \times 1$. Function $\mathbf{f}$ is vector-valued function $(\mathrm{f}(\boldsymbol{x}_1; \boldsymbol{\beta}), \ldots, \mathrm{f}(\boldsymbol{x}_n; \boldsymbol{\beta}))$. In what follows we might shorten $\mathrm{f}(\boldsymbol{x}_i; \boldsymbol{\beta})$ to $\mathrm{f}_i(\boldsymbol{\beta})$ or even to f_i.

4.1.1 Some nonlinear models

Some nonlinear model types are introduced here, but because any (non)linear function f will introduce NLM, the list is merely just a small set of examples. First of all, multiplicative model is NLM if errors are additive, i.e.

$$Y_i = \beta_0 x_{i1}^{\beta_1} \cdots x_{ik}^{\beta_k} + \epsilon_i \tag{4.3}$$

Please note that if errors are also multiplicative, the model can be transformed into linear:

$$Y_i = \beta_0 x_{i1}^{\beta_1} \cdots x_{ik}^{\beta_k} e^{\epsilon_i} \quad \Rightarrow \tag{4.4}$$

$$\log(Y_i) = \log(\beta_0) + \beta_1 \log(x_{i1}) + \ldots + \beta_k \log(x_{ik}) + \epsilon_i \tag{4.5}$$

In modeling the degree of linear polarization in atmosphereless Solar System targets such as asteroids covered with regolith, or dust in comets coma, the so-called trigonometric model is used. It is defined as

$$Y_i = \beta_1 \, \sin(x_i)^{\beta_2} \, \cos(x_i/2)^{\beta_3} \, \sin(x_i - \beta_4) + \epsilon_i, \tag{4.6}$$

where x_i is the phase angle and Y_i is the degree of linear polarization. The function is shown in Fig. 4.1(a).

A model for limited growth is shown in Fig. 4.1(b). The model is

$$Y_i = \beta_1 + \beta_2 \left(1 - e^{-\beta_3 x_i}\right) + \epsilon_i, \tag{4.7}$$

The growth starts from β_1 and is limited by $\beta_1 + \beta_2$. The parameter β_3 controls the speed of growth.

A growth curve can be defined so that it will reach its maximum, but slowly decline after that. A model that is shown in Fig. 4.1(c) is

$$Y_i = \beta_1 + \frac{\beta_2 x_i}{\beta_3 + x_i + \beta_4 x_i^2} + \epsilon_i, \tag{4.8}$$

The growth starts from β_1 and reaches its maximum at $\sqrt{\beta_3/\beta_4}$, but will then decrease.

One more type of growth curves is the S-type curves such as the logistic function in Fig. 4.1(d):

$$Y_i = \frac{\beta_1}{1 + e^{-\beta_2(x_i - \beta_3)}} + \epsilon_i, \tag{4.9}$$

where β_1 controls the limiting value of the growth, β_2 its steepness, and β_3 the location where positive derivative turns into negative.

Many of the NLM's can be derived as a solution for differential equation, for example the growth curves (b) and (d).

4.2 Model estimation

Most of the model estimation and diagnostics are done more or less the same way as in linear model. The main difference is, that results regarding the distribution of parameters, i.e. parameter errors, are always asymptotic, and that the model estimation is a numerical optimization problem. With LM the model estimate is given in closed form, and results regarding parameter distributions are exact under the normal assumption.

Let us derive the NLM parameter estimate from the maximum likelihood principle, although the same result can be reached from the 'minimal least squares' principle. Our model, now with normal assumption, is that

$$\begin{aligned} \epsilon_i \perp\!\!\!\perp \epsilon_j, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ , or alternatively} \\ Y_i \perp\!\!\!\perp Y_j, \quad Y_i \sim \mathcal{N}(\mathrm{f}_i(\boldsymbol{\beta}), \sigma^2) \end{aligned} \tag{4.10}$$

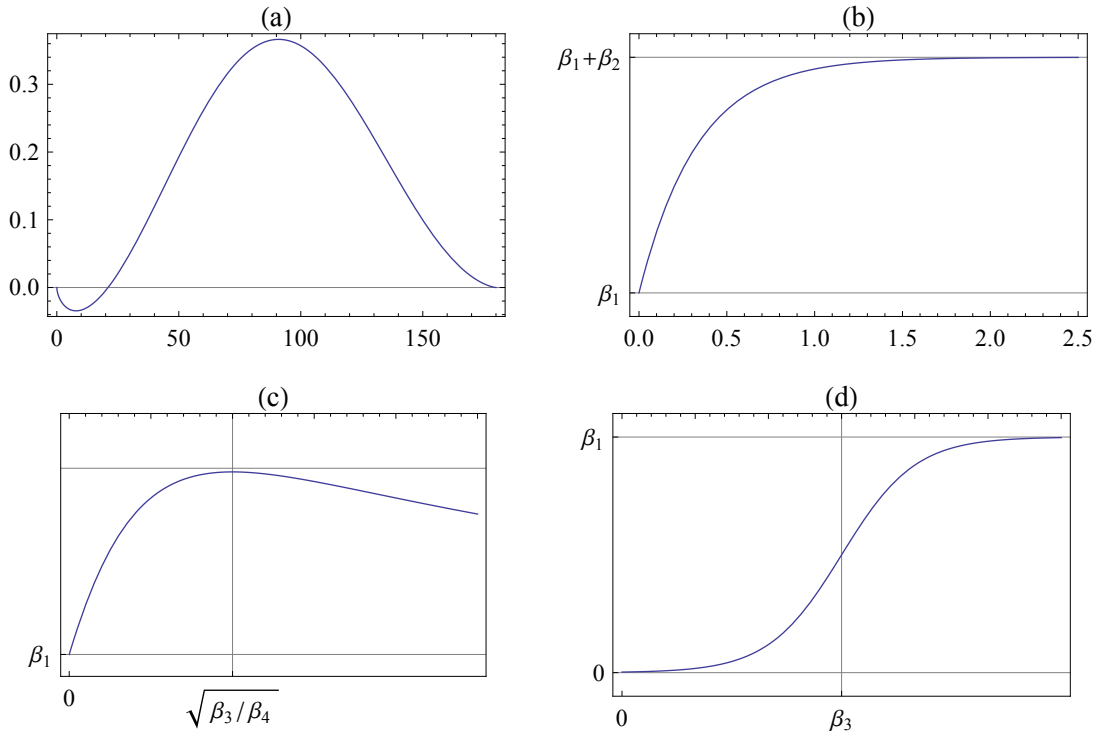


Figure 4.1: Four examples of different models in nonlinear regression.

Because the i.i.d observations, the likelihood function for the model is

$$\mathrm{L}(\boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\sum^{n}\left(y_i - \mathrm{f}_i(\boldsymbol{\beta})\right)^2\right) \tag{4.11}$$

We will write the squared residual sum in a shorter form, $S(\boldsymbol{\beta}) = \sum^n \left(y_i - \mathrm{f}_i(\boldsymbol{\beta})\right)^2$, and state that the log-likelihood function for the model is

$$\mathrm{l}(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}S(\boldsymbol{\beta}) \tag{4.12}$$

The maximum of the log-likelihood gives the ML estimates for the NLM. Regarding to parameter vector $\boldsymbol{\beta}$, we can easily see that estimate $\boldsymbol{b} = \hat{\boldsymbol{\beta}}$ must minimize the sum of squared residuals $S(\boldsymbol{\beta})$. When inputting that back to log-likelihood, derivating with respect to σ^2, and searching for root, we find that $s^2 = \hat{\sigma}^2 = \frac{1}{n}S(\boldsymbol{b})$.

Contrary to linear model, the estimate $\boldsymbol{b}$ cannot (usually) be expressed in closed form. The minimization of $S(\boldsymbol{\beta})$ must be done numerically. Quite generally Gauss-Newton or Levenberg-Marquardt algorithms are used.

4.2.1 Parameter properties

The asymptotic properties of the NLM estimates $\boldsymbol{b}$ and s^2 can be found by analyzing the Hessian matrix of the MLE's (see Eq. (2.6)). After some cumbersome calculus,

we can find that for the residual variance we have

$$s^2 \overset{as.}{\sim} \mathcal{N}\left(\sigma^2, \frac{2\sigma^4}{n}\right), \tag{4.13}$$

and for the actual parameters

$$\boldsymbol{b} \overset{as.}{\sim} \mathcal{N}_n\left(\boldsymbol{\beta}, \sigma^2\left(\mathbf{F}(\boldsymbol{\beta})^T\mathbf{F}(\boldsymbol{\beta})\right)^{-1}\right). \tag{4.14}$$

The matrix $\mathbf{F}(\boldsymbol{\beta})$ is short for the $n \times p$ partial derivative matrix with elements

$$\mathbf{F}(\boldsymbol{\beta}) = \left[\frac{\partial \mathrm{f}_i(\boldsymbol{\beta})}{\partial \beta_j}\right]_{ij}. \tag{4.15}$$

The tests regarding individual parameters in NLM are done in similar manner than with LM, only change being that instead of matrix $\mathbf{M}^{-1} = (\mathbf{X}^T\mathbf{X})^{-1}$ in LM (see Eqs. (3.33)–(3.35)) we have matrix $\mathbf{M}^{-1} = (\mathbf{F}(\boldsymbol{\beta})^T\mathbf{F}(\boldsymbol{\beta}))^{-1}$ in NLM.

The model diagnostics with e.g. residual plots are also done as with LM. The covariance matrix of $\boldsymbol{b}$ is even more important than with LM — highly correlated parameters are hard to estimate with numerical methods. Moving to a different parametrization might help.

Chapter 5

Nonparametric regression and distribution estimation

Nonparametric methods in statistics refer to analysis methods which try to avoid assuming any particular parametric distribution in the model. Usually, the assumption to be avoided is the normal distribution assumption. As contrary to the name nonparametric (*epäparametrinen*), these methods usually have a large number of parameters.

Nonparametric methods are used in all fields in data-analysis. For example, there is a variety of nonparametric tests available. However, here we mention only two nonparametric methods — spline regression and kernel density estimation.

5.1 Spline regression and other smoothing techniques

Sometimes the functional form or dependence between explanatory variable(s) and dependent variable is not interesting in such, but only some kind of smooth description of the behavior. In these cases either direct smoothing of the data or regression smoothing is searched for.

There are many different data smoothing techniques, from which moving average or moving median are the most simple ones. In these, the values y_i are replaced by average (or median) over a smoothing window that holds k observations around the i'th observation. An example of such smoothings are shown in Fig. 5.1 with window size of 10. Other, more advanced methods include, e.g., LOESS or LOWESS smoothing.

One more interesting smoothing or nonparametric regression technique is the spline regression. This method should not be mixed with spline interpolation where all

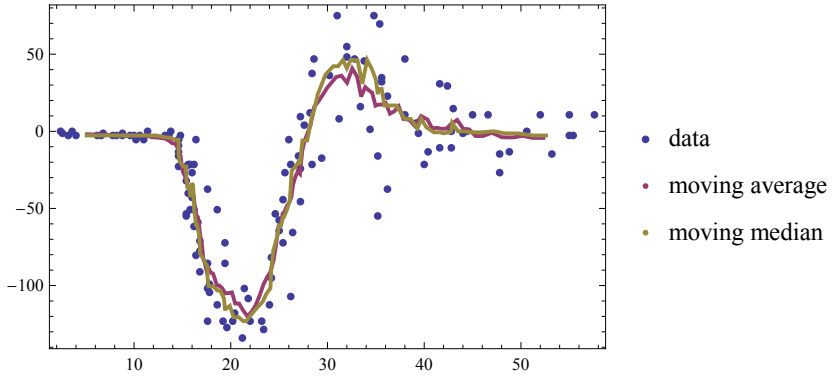


Figure 5.1: Moving average and moving median smoothing to the data.

the variability of the observations is reproduced. In spline regression, a small number of so-called cubic B-splines that are local third-order polynomials are used as a basis for linear regression. When the spline basis $\mathrm{B}_j(x)$ is formed, the sum of these, $\sum_j \beta_j \mathrm{B}_j(x)$ is fitted to the data in least-square sense.

The spline basis functions are distributed to the range of explanatory variables x_i evenly, or preferably to the quantiles of the data. We will not go into details with B-spline basis derivation, there are suitable material in e.g. Wikipedia or in Numerical Recipies. A spline regression for the data in the previous moving average/median example is shown in Fig. 5.2, together with the cubic spline basis that is distributed along x to 7 quantiles of the data plus the end-points, 0%, 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, 100%.

For technical reasons, the spline basis if formed with knots where the end-points are repeated four times in the knot list, so with k quantiles there are $k + 2 \times 4$ knots in the basis. With those knots, total of $k + 4$ splines are available.

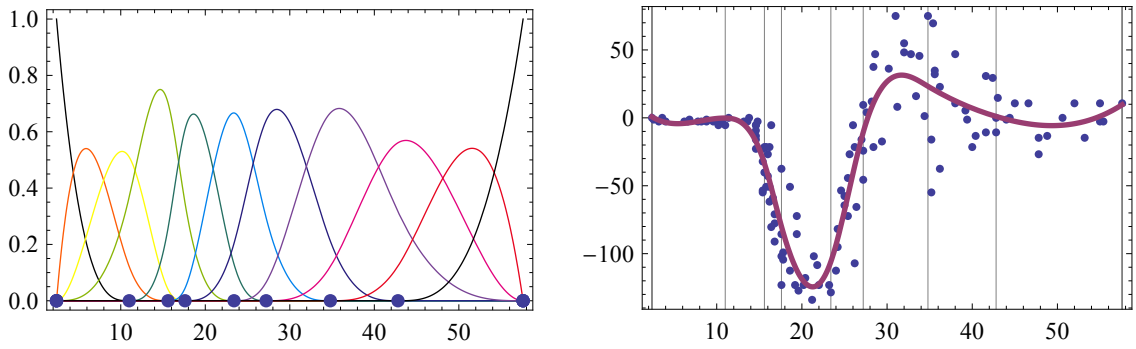


Figure 5.2: Spline basis for 7 quantiles and end-points of the data (left) and fitted regression spline of the basis functions (right).

5.2 Kernel estimation

Kernel estimation (*ydinestimointi*) is a nonparametric method for estimating (continuous) distribution (pdf) of the data. The methods works for both one-dimensional or multidimensional data. The result of kernel estimation is not a parametrized, close-formed distribution, but a numerical function that can be used to compute values of the distribution estimate.

The idea of kernel estimation is quite simple. Every observation x_i in the data is replaced by a kernel function $\mathrm{K}_i(x; x_i, h)$, and the total kernel estimate is the scaled sum of kernels:

$$\mathrm{K}(x; \boldsymbol{x}, h) = \frac{1}{n} \sum_{i}^{n} \mathrm{K}_i(x; x_i, h), \tag{5.1}$$

where $\boldsymbol{x}$ is the data vector, x the value where the distribution is evaluated, and h is the smoothing parameter (*siloitusparametri*).

The choice of the kernel function should not be too critical, any non-negative function that is symmetric around its maximum and integrates to one should do. One suitable choice is to use the pdf of normal distribution, with expected value $\mu = x_i$ and variance $\sigma^2 = h^2$. So, kernel is

$$\mathrm{K}_i(x; x_i, h) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{(x - x_i)^2}{2h^2}\right). \tag{5.2}$$

More important than the actual shape of the kernel should be the choice of the smoothing parameter h. There are different advice, one of such is the method of Silverman:

$$h = s \left(\frac{4}{p+2}\right)^{\frac{1}{p+4}} n^{-\frac{1}{p+4}}, \tag{5.3}$$

where p is the dimension of the data, and s the standard deviation of the data in one-dimensional case. An example of kernel estimation of the density function for three observations is shown in Fig. 5.3.

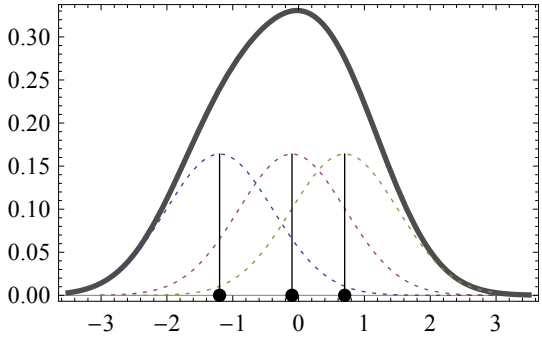


Figure 5.3: Three observations, normal pdf kernels and the kernel density estimate of the pdf.

Kernel estimation suits quite well for multidimensional cases, too. For these, a multidimensional normal distribution pdf can be used as the kernel with covariance matrix $\mathbf{H}$ (when following Silverman's rule) as

$$\mathbf{H} = \mathbf{S}\left(\frac{4}{p+2}\right)^{\frac{2}{p+4}} n^{-\frac{2}{p+4}}. \tag{5.4}$$

Example for two-dimensional kernel estimate is shown in Fig. 5.4.

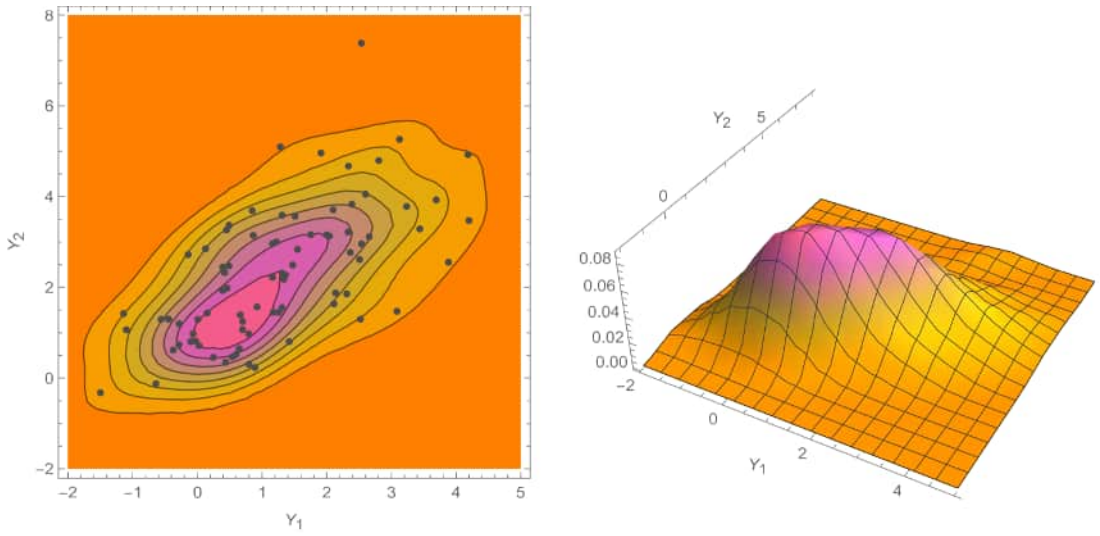


Figure 5.4: Two-dimensional observations and kernel estimate for the pdf. On left, a contour plot of the estimate with the data, on right, 3-D surface plot of the kernel estimate.

Chapter 6

Multivariate methods

Multivariate methods in data-analysis refer to the vast collection of methods that are applied to data with several variables. In principle, regression analysis (linear or nonlinear models) with multiple variable data is also a multivariate method, but usually multivariate regression is treated separately. Different clustering, classification, pattern recognition, and dimension reduction methods are in the core of multivariate data-analysis.

6.1 Multivariate distributions

Multivariate distributions are distributions for vector-valued random variables, and multivariate pdf's and cdf's are functions from $\mathbb{R}^n$ to positive real axis $\mathbb{R}^+$. Apart from the fact that the variable is multidimensional, they are just like one-dimensional distributions.

With one-dimensional distributions, there are plenty of different choices available. With multiple dimensions, the multivariate normal distribution governs the field and other choices are rare. With independent variables this is not an issue, since the joint distribution of independent components is the product of the one-dimensional distributions. With just a few components these distributions are often called by the names of the individual components, e.g., gamma-normal distribution for the product distribution of gamma and normal distributed variables.

6.1.1 Multinormal distribution

Multinormal distribution for p-dimensional random vector $\boldsymbol{Y}$, $\mathcal{N}_p$, is parametrized by p-dimensional vector of expected values $\boldsymbol{\mu}$ and $p\times p$-dimensional covariance matrix $\boldsymbol{\Sigma}$. The pdf is

$$\mathrm{f}(\boldsymbol{y};\boldsymbol{\mu},\boldsymbol{\Sigma}) = (2\pi)^{-\frac{p}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right), \qquad (6.1)$$

where $\det(\cdot)$ is the determinant of a matrix.

The covariance matrix $\mathbf{\Sigma}$ has all the information about the dependencies between multinormal variables. Two variables Y_i and Y_j are independent if $[\mathbf{\Sigma}]_{ij} = \sigma_{ij} = \sigma_{ji} = 0$. In that case their correlation is also zero. Note that for other than multinormal variables it can be that the (linear) correlation between the variables is zero, but that they are not independent. For normal distribution, however, correlation is equivalent to dependency.

The possible dependency can be generalized to groups of variables. Let us say that the random vector $\boldsymbol{Y}$ constitutes of k components $A_1, \ldots, A_k$, and m components $B_1, \ldots, B_m$. The random vector, expected value vector, and the covariance matrix can be partitioned into submatrices or subvectors:

$$\boldsymbol{Y} = [\boldsymbol{A}\ \boldsymbol{B}]^T = [A_1\ \cdots\ A_k\ B_1\ \cdots\ B_m]^T \tag{6.2}$$

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_A\ \boldsymbol{\mu}_B]^T = [\mu_{A_1}\ \cdots\ \mu_{A_k}\ \mu_{B_1}\ \cdots\ \mu_{B_m}]^T \tag{6.3}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{AA} & \mathbf{\Sigma}_{AB} \\ \mathbf{\Sigma}_{AB} & \mathbf{\Sigma}_{BB} \end{bmatrix} \tag{6.4}$$

Now, if the variables $\boldsymbol{A}$ are all independent of $\boldsymbol{B}$, it means that $\mathbf{\Sigma}_{AB} = \mathbf{0}$. Furthermore, it holds now that $\mathbf{A} \sim \mathcal{N}_k(\boldsymbol{\mu}_A, \mathbf{\Sigma}_{AA})$ and similarly for $\mathbf{B}$. Two examples of pdf's of two-dimensional normal distribution are shown in Fig. 6.1. The variables are independent in the first example, and dependent on the second.

Construction of multinormal distribution

It might be useful to understand how multinormally distributed variables are formed. First of all, we need p random variables Z_i that are independently and normally distributed. Without loss of generality, we can assume at this point that they all are distributed as $Z_i \sim \mathcal{N}(0, 1)$.

Second, let us have a $p \times p$ matrix of coefficients c_{ij}, $\mathbf{C}$. Third, we need a vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)$. Now we can construct a new random vector $\boldsymbol{Y}$ as

$$\begin{aligned} Y_1 &= c_{11}Z_1 + \ldots + c_{1p}Z_p + \mu_1 \\ Y_2 &= c_{21}Z_1 + \ldots + c_{2p}Z_p + \mu_2 \\ &\vdots \\ Y_p &= c_{p1}Z_1 + \ldots + c_{pp}Z_p + \mu_p \end{aligned} \tag{6.5}$$

which can be written shorter as

$$\boldsymbol{Y} = \mathbf{C}\boldsymbol{Z} + \boldsymbol{\mu} \tag{6.6}$$

After this transform $\boldsymbol{Y}$ has multinormal distribution $\boldsymbol{Y} \sim \mathcal{N}_p(\boldsymbol{\mu}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^T$.

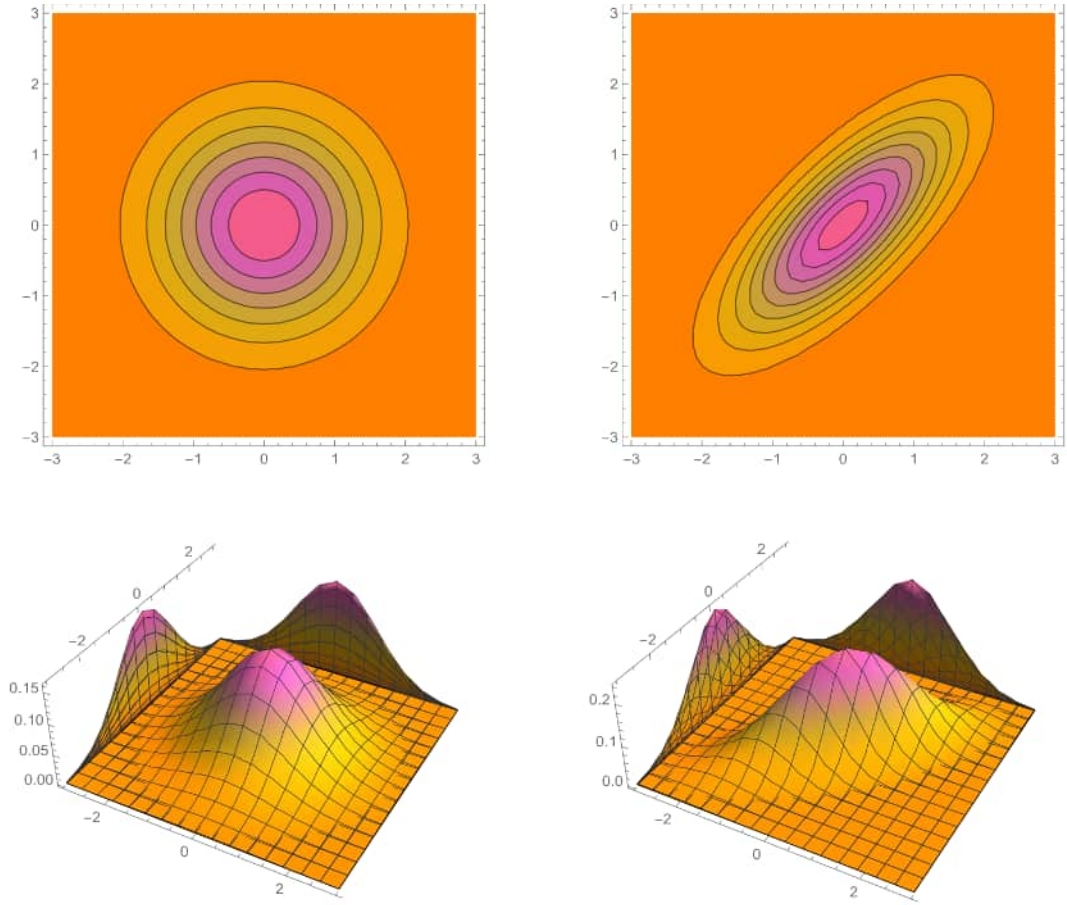


Figure 6.1: Contour plots (upper row) and 3D plots (lower row) of two-dimensional normal distribution. Distribution on left has no dependence ($\rho = 0$) between the variables, while distribution on the right has $\rho = 0.75$.

The construction of multinormal variables above can be used to create samples of (pseudo)random numbers from multinormal distribution. The creation of standard $(0, 1)$ normal random numbers is available in almost all software packages, so it is easy to create sample $\boldsymbol{Z} = (Z_1, \ldots, Z_p)$. The required covariance matrix should be decompositioned with Cholesky decomposition $\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$, or preferably with eigendecomposition (*ominaisarvohajotelma*) $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues. In the latter case, $\mathbf{C} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}$. Now Eq. (6.6) can be directly applied to $\boldsymbol{Z}$ to get the multivariate random sample:

$$\boldsymbol{Y} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Z} + \boldsymbol{\mu}. \tag{6.7}$$

Because $\boldsymbol{\Lambda}$ is diagonal matrix, $\boldsymbol{\Lambda}^{1/2}$ is simply $\left[\sqrt{\Lambda_{11}} \cdots \sqrt{\Lambda_{pp}}\right]$. Note that the equation above is for one sample vector $\boldsymbol{Y}$. If you need to construct a matrix $\mathbf{Y}$ where all the rows are from the same multinormal distribution, $\boldsymbol{Y_i}. \sim \mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})\ \forall i$, then the matrix version of the construction is as

$$\mathbf{Y} = \mathbf{Z}\,\boldsymbol{\Lambda}^{1/2}\,\mathbf{U}^T + \mathbf{1}_{n,p}\,\mathrm{diag}(\boldsymbol{\mu}). \tag{6.8}$$

where $\mathbf{1}_{n,p}$ is $n \times p$ matrix full of ones, and $\mathrm{diag}(\cdot)$ is an operator that constructs a diagonal matrix of the values.

Mahalanobis distance

The Mahalanobis distance is a generalized distance measure that is suitable for multinormal-distributed variables. Let us have an example of two-dimensional sample from multinormal distribution as in Fig. 6.2. The two variables might measure completely different quantities and thus have different scales. The expectancy of the distribution is at $(100, 1)$. Let us say that we have three interesting observations, the red, green and the blue dots in the figure. One might want to know which one is furthest from the expected value.

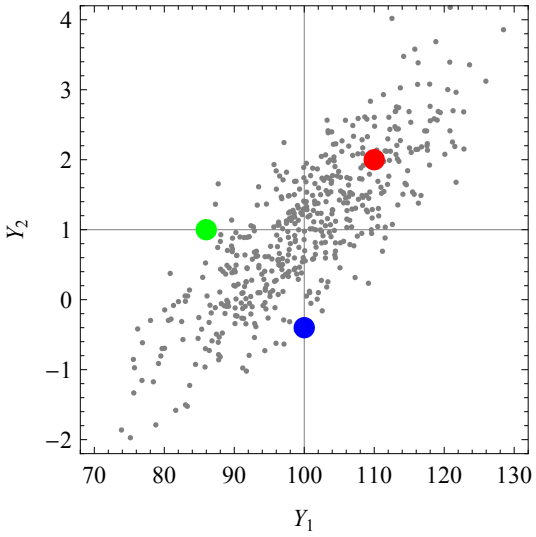


Figure 6.2: Random multinormal sample and Mahalanobis distance.

The expected value (mean) has coordinate $\overline{\boldsymbol{y}} = (\overline{y_1}, \overline{y_2})$. The squared Euclidean distance to mean would be $D_e^2 = (\boldsymbol{y} - \overline{\boldsymbol{y}})^T(\boldsymbol{y} - \overline{\boldsymbol{y}})$. In this case, the distances would be about 10 (red), 14 (green), and 1.4 (blue) for the three colored dots. Euclidean distance is clearly a bad measure in this case, since it assumes that both coordinate axes Y_1 and Y_2 have the same scale.

An improved version of the distance measure could be constructed if the observations would be normalized (scaled with their standard deviations) before taking the Euclidean distance. However, that procedure would not take into account the evident strong correlation between the variables. After normalization the points would all have approximately the same Euclidean distances to mean. Still, based on the gray sample points from the distribution, it would seem that the red point is "more typical" and should have the smallest distance from the mean.

The Mahalanobis distance takes both the scales of the different axis and the corre-

lation into account. The distance is defined as

$$D_m = \left((\boldsymbol{y} - \overline{\boldsymbol{y}})^T \, \mathbf{S}^{-1} \, (\boldsymbol{y} - \overline{\boldsymbol{y}}) \right)^{1/2}, \tag{6.9}$$

where S is the sample estimate of the covariance matrix. One can see that the Mahalanobis distance is Euclidean distance that is weighted by the inverse of the covariance matrix. For multinormal sample this is the correct distance measure to be used.

Test of multinormality with Mahalanobis distance

There are a number of tests for multinormality, each focusing on different requirements for a multinormal sample. The Mahalanobis distance can also be used to test multinormality. It can be shown that the squared Mahalanobis distances in multinormal sample should have the χ^2-distribution with p degrees of freedom. The Q-Q plot, as described in Fig. 3.7 and the related text, can be used to graphically check the distribution assumption. Sorted squared distances are plotted on the vertical axis, and quantiles from the $\chi^2(p)$-distribution of the squared distances on the horizontal axis. The points should lie close to diagonal line if the sample is from multinormal distribution.

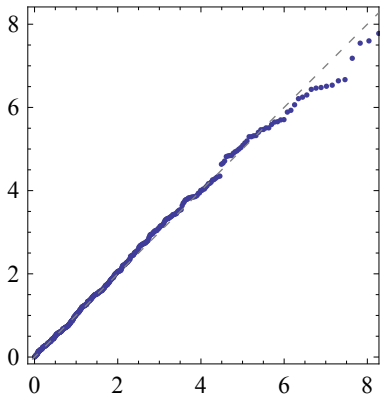


Figure 6.3: Q-Q-plot of the squared Mahalanobis distances against χ^2-distribution from the sample in Fig. 6.2

6.2 Dimension reduction

In multivariate analysis we often need to deal with data that has many (tens, hundreds) properties (i.e., variables) measured or observed. However, even the visual presentation of such data can be difficult, not even mentioning the statistical analysis. It is quite possible that there are underlying dependencies among the variables. Finding these dependencies can be valuable in such, but it can also help to reduce the number of variables.

With reduced dimensions the visual data mining, clustering, classification etc. will become easier. In this section we will go through two methods to analyze the underlying *linear* dependencies in the data, the principle component analysis (PCA) and the linear discriminant analysis (LDA). If correlated variables exist in the data, both the methods can be used to find them, to create a set of new variables based on these correlations, and to reduce the dimensionality of the data by dropping out the non-important new variables.

The difference between the PCA and the LDA is that the PCA is suitable for so-called *unsupervised* analysis where we have no prior knowledge about the possible groups/clusters/classes in the data, and LDA for *supervised* analysis where we have a 'training set' for which we already know the groups/clusters/classes of every observation.

6.2.1 Principle component analysis

Principle component analysis (PCA, *pääkomponenttianalyysi*) is one of the most important multivariate methods, especially in natural sciences. In social sciences Factor Analysis (*faktorianalyysi*) is similar and popular method, but PCA is more 'physical' while there are more possibilities to subjective judgment in factor analysis.

The importance of PCA comes from its wide applicability. PCA can be used in visual analysis, clustering, pattern recognition, exploratory data analysis, variable reduction, searching for dependency structures etc. Furthermore, PCA is quite straightforward to implement and is 'objective' in the sense that it does not need any parameters to be set.

PCA can be understood perhaps the easiest way by a geometrical approach. In Fig. 6.4 (a) there are contour ellipses from two-variate normal distribution. There is correlation between the variables, so the axis of the ellipsoids are not parallel to the coordinate axis. What the PCA does is that is searches for these axis of the contour ellipses and then transforms the data so that the ellipse axis are the new coordinate vectors. After PCA the new variables (coordinate axis) are uncorrelated, as shown in Fig. 6.4 (b).

Implementing principle component transform

The PCA can be implemented quite easily in a computing environment where there are tools for matrix algebra and for eigenvalue decomposition. The data matrix $\mathbf{Y}$ has n rows, one for each observation, and p columns for the variables. First the data matrix needs to be centered or standardized. If the data is only centered, the method is based on the covariances, and if standardized, it is based on the correlations.

The correct method can be chosen based on the quantities and scales the variables are measuring. If all the variables measure the same quantity, and we want to

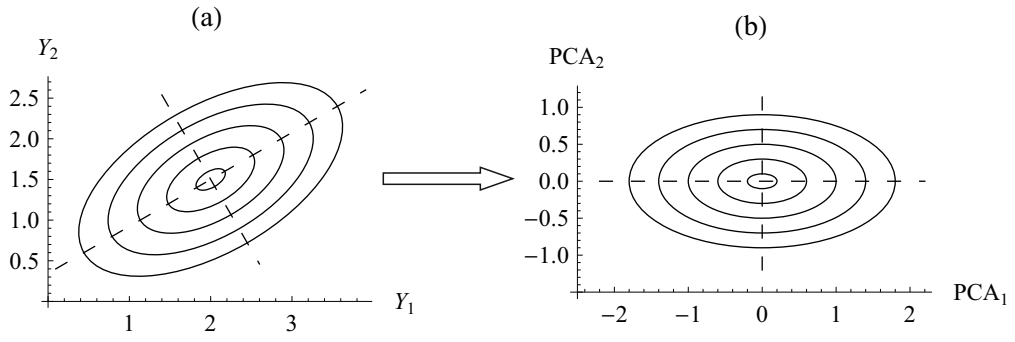


Figure 6.4: Sketch of the PCA in geometrical interpretation.

preserve the information that is in the variances of the variables, we should choose the covariance method. The centering of the data is done using the mean vector $\overline{\boldsymbol{y}}$ which holds the mean values over the observations for each variable, i.e.

$$\overline{\boldsymbol{y}} = (\overline{y}_1, \ldots, \overline{y}_p) = \frac{1}{n}(\sum_i^n y_{i1}, \ldots, \sum_i^n y_{ip}). \tag{6.10}$$

The centered data matrix $\mathbf{X}$ is computed from $\mathbf{Y}$ by:

$$\mathbf{X} = \mathbf{Y} - \mathbf{1}_{n,p}\,\mathrm{diag}(\overline{\boldsymbol{y}}), \tag{6.11}$$

where $\mathbf{1}_{n,p}$ is $n \times p$ matrix full of ones, and $\mathrm{diag}(\cdot)$ is an operator that constructs a diagonal matrix of the values.

However, if the variables measure different quantities and their variances cannot be compared with each other, we should choose the correlation method and use the standardized data matrix. In standardization the centered data is further divided by standard deviations, variable by variable. This can be formulated with the diagonal matrix of inverses of standard deviations, $[\mathbf{V}]_{ii} = 1/s_{ii}$ as

$$\mathbf{X}^* = \mathbf{X}\,\mathbf{V} \tag{6.12}$$

The rest of the PCA procedure is identical for correlation and covariance methods, so we use symbol $\mathbf{X}$ for both the cases. Next, the sample estimate to covariance matrix $\mathbf{S}$ is needed. If (and only if) the data matrix is centered, as with $\mathbf{X}$ here, the sample covariance matrix can be computed as

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}^T\,\mathbf{X}, \tag{6.13}$$

If $\mathbf{X}$ was standardized, $\mathbf{S}$ is actually correlation matrix.

Third step is to compute the eigenvalue decomposition of $\mathbf{S}$. Eigenvalue decomposition is such that

$$\mathbf{S} = \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T, \tag{6.14}$$

where $\mathbf{U}$ is the $p \times p$ matrix of eigenvectors, and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues*. Finally, the data is transformed into PCA space by

$$\mathbf{Z} = \mathbf{X}\,\mathbf{U}. \tag{6.15}$$

An example of PCA transform in shown in Fig. 6.5.

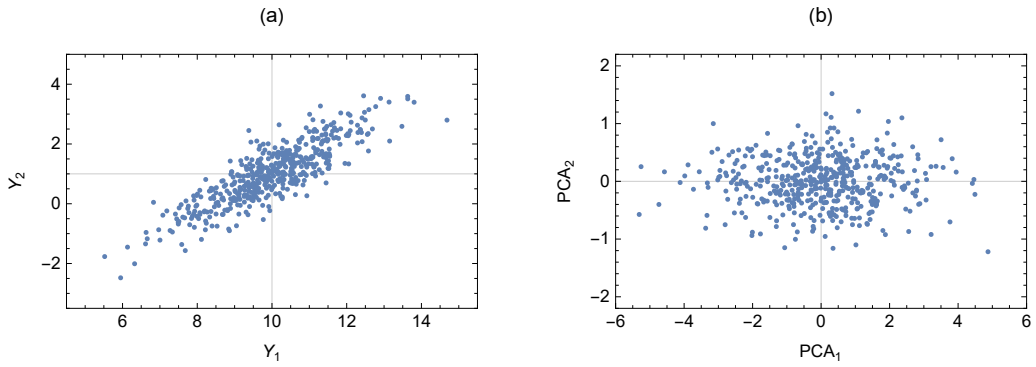


Figure 6.5: Example of PCA transform to 500 observations from two-dimensional multinormal distribution. Original observations are in subfigure (a), and data in PCA space in (b).

Interpretation of principal components

As can be seen from Eq. (6.15), PCA is a linear transform. If $\boldsymbol{u}_j$'s are the eigenvectors in $\mathbf{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_p]$, and $\mathbf{x}_i$ is the row in centered (standardized) data matrix, the value of jth new PCA variable for observation i is

$$z_{ij} = \boldsymbol{x}_i^T\,\boldsymbol{u}_j = x_{i1}u_{1j} + \ldots + x_{ip}u_{pj} \tag{6.16}$$

In that context, the eigenvectors $\boldsymbol{u}_j$ are the new coordinate basis, and map the original variables to the PCA space. The eigenvectors are often called *loadings*. Large absolute values in u_{kj} mean that original variable k has large impact, loading, to PCA variable j. Therefore by plotting eigenvectors one can visually inspect how the original variables influence the PCA variables.

The eigenvalues, i.e. the diagonal values in $\mathbf{\Lambda}$ are the variances of the data in the PCA space. The PCA will preserve the total variance of the data, i.e.

$$\sum_j^p [\mathbf{\Lambda}]_{jj} = \sum_j^p [\mathbf{S}]_{jj} \tag{6.17}$$

In PCA based on the standardized data matrix the total correlation is preserved, so $\sum_j^p [\mathbf{\Lambda}]_{jj} = p$.

*Note that some numerical eigensystem algorithms might return the matrix of eigenvectors so that the eigenvectors are the rows of $\mathbf{U}$. In the equations here we assume that the eigenvectors are the columns of $\mathbf{U}$. You can easily check which way it is by checking if $\mathbf{S} - \mathbf{U}\,\mathbf{\Lambda}\,\mathbf{U}^T = \mathbf{0}$ or if $\mathbf{S} - \mathbf{U}^T\,\mathbf{\Lambda}\,\mathbf{U} = \mathbf{0}$. In the latter case the eigenvalues are the rows, and you need to compute $\mathbf{Z}$ as $\mathbf{Z} = \mathbf{X}\mathbf{U}^T$.

Principal component analysis in variable reduction

One of the applications of PCA is in variable or dimensionality reduction or data compression. The fact that the PCA variables are uncorrelated makes this possible. Unnecessary PCA variables can be removed without affecting the remaining variables. The variances of the PCA variables is used to judge which variables are ”unnecessary”.

Usually the procedure that computes eigenvalues and -vectors already sorts them so that the first eigenvalue is the largest and so forth. The eigenvectors are also sorted because the order of values and vectors must match. If this is not done by the procedure, one should do this manually. So, eigenvalues must be sorted so that $\Lambda_{[1]} \geq \Lambda_{[2]} \geq \cdots \geq \Lambda_{[p]}$. The same ordering must then be applied for eigenvectors, $\mathbf{U} = [\boldsymbol{u}_{[1]}\, \boldsymbol{u}_{[2]}\, \cdots\, \boldsymbol{u}_{[p]}]$.

If there are correlations between the original variables, it is often so that the total variance in the data is redistributed with PCA variables so that the first few PCA variables make up almost all the total variance. The interpretation is that the first few PCA variables with large variances are the ”real signal” and the rest of the PCA variables with variances close to zero are ”random noise”. Variable reduction is based on this.

The portion c of total variance that is reproduced with the first k PCA variables is derived with

$$c = \frac{\sum_j^k \Lambda_j}{\sum_j^p \Lambda_j}. \tag{6.18}$$

Usually the limit for c is set close to 100 %, to 95 % or 99 % for example. When the first k PCA variables can reproduce the required portion, the variable reduction is done by forming $\mathbf{U}^* = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_k]$, i.e. taking only the first k eigenvectors and dropping out the rest. The reduced data $\mathbf{Z}^*$ in PCA space is received by $\mathbf{Z}^* = \mathbf{X}\,\mathbf{U}^*$. The reduced matrix has now only k variables. If the PCA variable reduction is successful, the reduced number of variables k can be significantly smaller than the original number of variables p.

One application for PCA variable reduction is the visualization of high-dimensional data. If the first two or three PCA variables can reproduce a large portion of the total variance, the data can be visualized in 2D or 3D plots in the reduced PCA space. Another is in classification or clustering problems. While PCA is not itself optimized for classification, it can find structures in the data that can be both visualized in low dimensions, and used in classification. An example of this is shown in Fig. 6.6.

6.2.2 Linear discriminant analysis

The PCA is a highly popular method for data analysis and dimension reduction, and is often used prior to clustering or classification analysis to produce (impor-

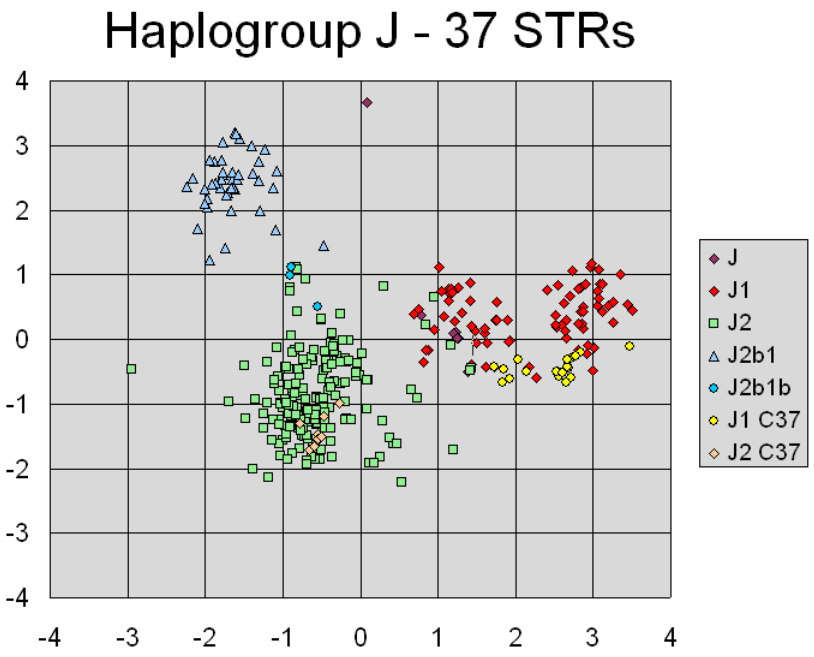


Figure 6.6: PCA example from Wikipedia. A PCA scatterplot of haplotypes calculated for 37 Y-chromosomal STR markers from 354 individuals. PCA has successfully found linear combinations of the different markers, that separate out different clusters corresponding to different lines of individuals' Y-chromosomal genetic descent.

tant) transformed variables that would be more optimal to be clustered. The PCA usually succeeds quite well, but it fails to take into account any prior knowledge about different classes of observations in the data. If these classes are actually known for the data, usually then called the training data, the linear discriminant analysis (LDA) is a very close relative to the PCA but with the ability to acknowledge the classes in the data.

A visual explanation about the difference between these methods is shown in Fig. 6.7. The original data, quite similar as in Fig. 6.5, is shown in the uppermost subplot. The data has two variables with evident linear correlation between them. In this example, we actually know beforehand that the data has two distinct classes of observations, the blue and the red dots.

As seen in the lower left subplot, the PCA will find the direction with the maximal variability in the data, which is the vertical axis in the subplot. However, the vertical axis is useless in classifying the blue and red dots from each other. Remember that the new axis are ordered from the most important to the least important. With only two variables, the second axis, which would separate the classes, is also the least important and would probably be left out in the dimension reduction procedure.

However, as seen from the subplot on the right side, the LDA does the opposite from the PCA and aligns the first and the most important axis along the direction

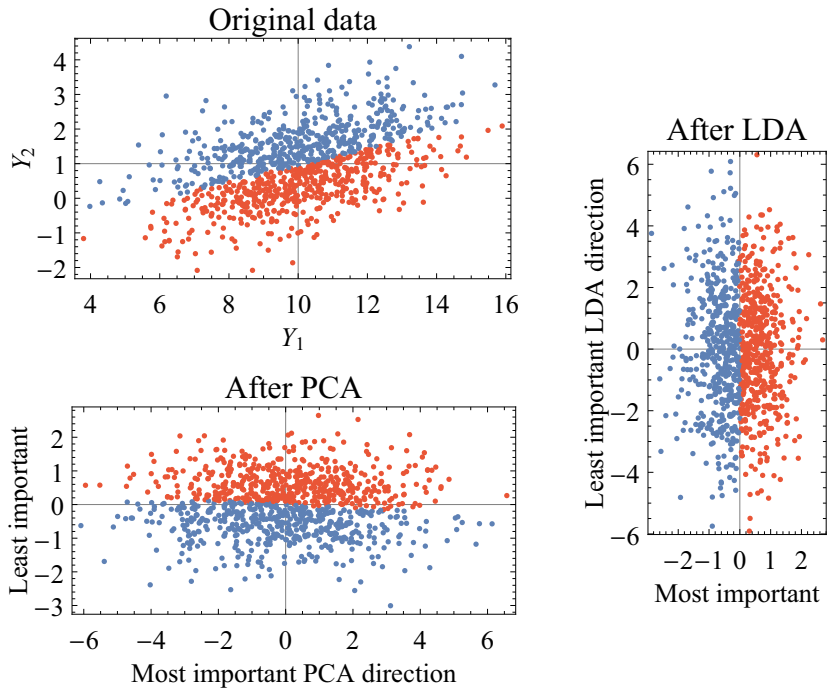


Figure 6.7: The difference between the PCA and LDA when there are known classes in the data.

which gives the most information for separating the classes. This is roughly what the LDA aims to do — find new variables to maximize the separation between the classes in the data.

Implementing linear discriminant analysis

Implementing LDA is somewhat similar to implementing the PCA, but it requires taking into account the different classes in the data. We will need to have the same standardized matrix $\mathbf{X}$ as in Eq. (6.11) or (6.12) for covariance or correlation-based analysis. In addition, we will need the mean vectors $\overline{\boldsymbol{x}_c}$ and covariance matrices $\mathbf{S}_c$ for the standardized observations $\boldsymbol{x}_i$ for each group $c = 1, \ldots, k$.

With these group-based mean vectors and covariances, we compute the within-class covariance matrix

$$\mathbf{W} = \sum_c (n_c - 1)\mathbf{S}_c, \tag{6.19}$$

where n_c is the number of observations for group c, and the between-classes covariance matrix

$$\mathbf{B} = \sum_c n_c (\overline{\boldsymbol{x}_c}\, \overline{\boldsymbol{x}_c}^T). \tag{6.20}$$

From these we can form the LDA projection matrix with the help of eigenvalue

decomposition as[†]

$$\mathbf{L}\mathbf{\Lambda}\mathbf{L}^{-1} = \mathbf{W}^{-1}\mathbf{B}, \tag{6.21}$$

where the matrix $\mathbf{L}$ is not orthogonal anymore, because the matrix to be decomposed is not symmetric as with the PCA. Again, different algorithms might output either $\mathbf{L}$ or $\mathbf{L}^T$, please check this before implementing.

Finally, the LDA-transformed new variables are given by

$$\mathbf{Z} = \mathbf{X}\mathbf{L}. \tag{6.22}$$

Similar to the PCA, the most important variables are ordered as the first columns in the matrix $\mathbf{Z}$, so the dimension reduction can be done by discarding the columns after the most important ones.

6.3 Classification

Classification, also called supervised learning, is a set of techniques to assign new observations into pre-defined classes. To have pre-defined classes one needs a training data set where these classes are already known. The classifier is 'trained' using this data, and then used for future observations without the knowledge of the correct classification. In the 'training' process the parameters of the classifier are estimated.

For classification, a pre-treatment to the data is usually needed, especially if the data has many variables. Dimension reduction such as PCA or LDA is recommended before the classification, since it makes the task for the classification algorithm easier. In an ideal case where the classes are known beforehand, LDA is recommended over PCA which does not exploit the class structure in the data.

However, sometimes classification is done so that the classes are not known beforehand. In the analysis PCA or similar is first applied to the training set without the class information. From the result a visual and subjective analysis is done to define the classes, and then the classifier is build upon these.

In what follows, three classification algorithms are introduced and applied for the classical Iris flower data set of three Iris flower species and the measures of their sepal and pedal flower dimensions. The data with four variables is first treated with LDA, and the two most important new variables are extracted, as seen in Fig. 6.8.

The first 'algorithm' to be introduced is a heuristic division of the variable space, often with some simple linear borders, to different class areas/volumes. This division is subjective and often done just 'by eye' by the researcher. Still, this kind of taxonomic systems are quite popularly used. An example of the variable space division for the Iris data is shown in Fig. 6.9.

[†]Note on the matrix inversion $\mathbf{W}^{-1}$. If there are more variables than observations per class in

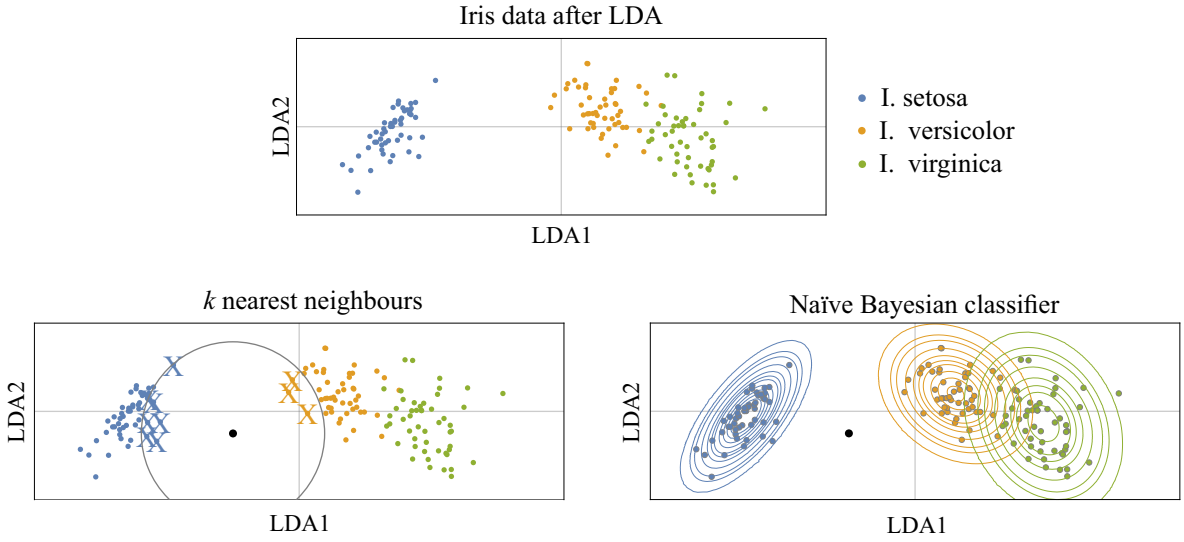


Figure 6.8: The Iris dataset after LDA dimension reduction on the top panel. On the lower left panel, a sketch of the k-nearest neighbor algorithm and on the right, the naïve Bayesian classifier algorithm.

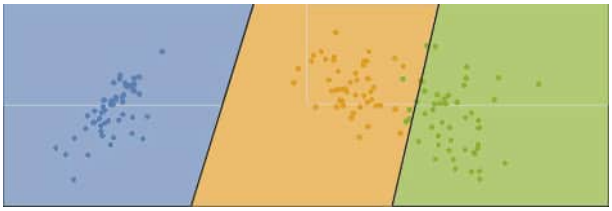


Figure 6.9: An example of a heuristic classification of the Iris data (after LDA transform) and the resulting areas of the variable space for the three classes.

6.3.1 Nearest-neighbor method

The nearest-neighbor (N-N) method is also quite simple but also robust method for classification. The simple procedure is to find n closest points in the training set for the new observation (see Fig. 6.8). The class frequencies within these n points are computed, and the most frequent class is assigned for the new observation. The most simple version searches only for the one nearest neighbor and classifies to the same class as this. This N-N method using only one point will actually create a Voronoi division of the variable space, as shown in Fig. 6.10.

If some kind of dimension reduction / variable transform technique is used before the N-N classification, the normal Euclidian distance should be a proper distance measure with this algorithm. However, if the original variables are used, one might want to apply more suitable distance measures such as the Mahalanobis or Man-

your data, the matrix $\mathbf{W}$ cannot be inverted. In that case, you can replace the matrix inversion with the so-called pseudoinverse or Moore-Penrose inverse $\mathbf{W}^{+}$.

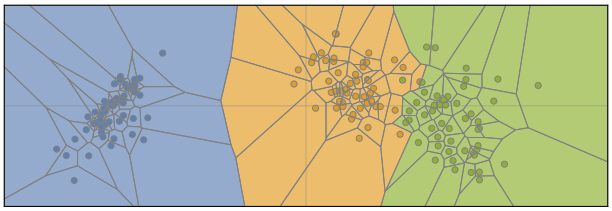


Figure 6.10: The Iris data after the LDA transform and the Voronoi division in the variable space corresponding to the one-nearest-neighbor classification areas.

hattan distances when searching for the nearest neighbors.

6.3.2 Naïve Bayes classifier

The probabilistic classifier is based on the probability distribution assumption on the data, and if the *a priori* probabilities are also considered, this method is called the naïve Bayesian classifier (NBC). In most cases, it is assumed that the data in each class follows a multinormal distribution with class mean vectors $\boldsymbol{\mu}_c$ and covariance matrices $\mathbf{S}_c$. With new observation $\boldsymbol{x}$, the probability p_c to belong to class c is computed simply from the assumed (multinormal) probability distribution

$$p_c = \mathrm{f}_c(\boldsymbol{x}; \boldsymbol{\mu}_c, \mathbf{S}_c). \tag{6.23}$$

If Bayesian classification is sought for, the *a priori* probabilities a_c, which can be computed for example from the training data frequencies, are also considered as

$$p_c = a_c \, \mathrm{f}_c(\boldsymbol{x}; \boldsymbol{\mu}_c, \mathbf{S}_c). \tag{6.24}$$

Finally, the most probable class is selected, i.e., the class c with the highest value of p_c. This is the probabilistic or the naïve Bayesian classifier. The multinormal distributions are shown in Fig. 6.8 for the Iris data, and the areas in the variable space in Fig. 6.11.

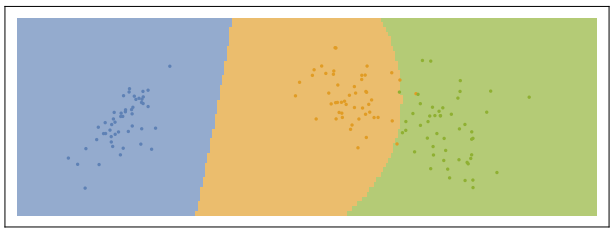


Figure 6.11: The Iris data after the LDA transform and the variable space division corresponding to the probabilistic classifier.

6.4 Clustering

While classification is supervised learning with pre-defined class information, clustering is a non-supervised method without the prior knowledge of the possible groups in the data. The clustering algorithms works by choosing groups for each observation by minimizing a chosen measure of "group conformance" while maximizing the difference between the groups in some sense. This is usually done for different number of groups, and the recommended number of groups is chosen so that it optimizes the ratio between "within-group" and "between-groups" variances. In clustering, as with classification, a pre-treatment to the data (PCA or similar) is recommended, or the distance measure can be chosen to be some other than the Euclidean distance. In Fig. 6.12, the Iris data is divided into 2 to 5 groups.

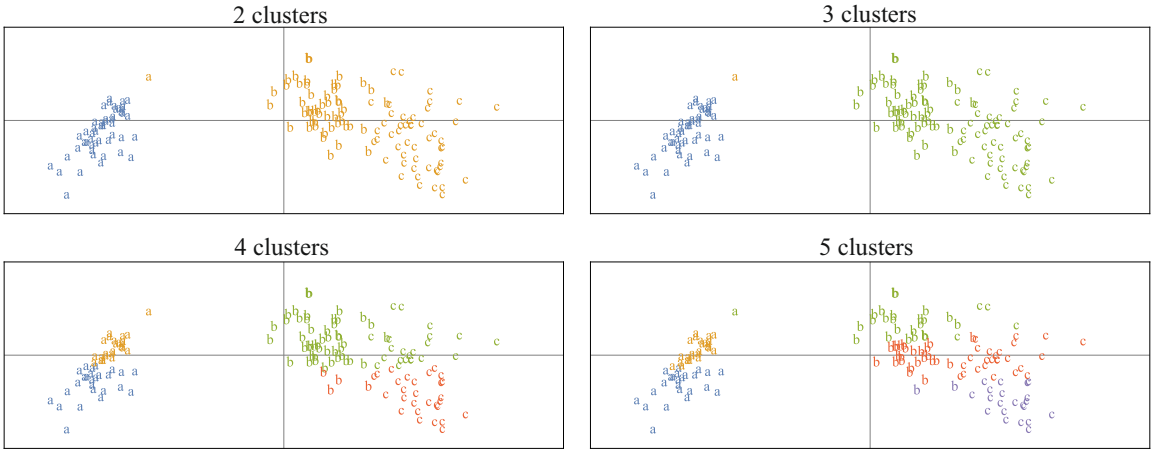


Figure 6.12: The Iris data after the LDA transform clustered into 2 to 5 groups. The letters a–c mark the real groups, and the colors the clusters found by the algorithm.

With the example in Fig. 6.12 one can see that at least the default clustering algorithm of the Mathematica software using the 'k-means' method cannot find the original flower subspecies from the data. The *setosa* species is found quite well when dividing into 2 or 3 clusters, but *versicolor* and *virginica* cannot be separated that well.

Clustering can be done in a tree-like structure where the observations are group together one-by-one and finally ending into one group. This structure can be plotted into *dendogram* or clustering tree, as in Fig. 6.13.

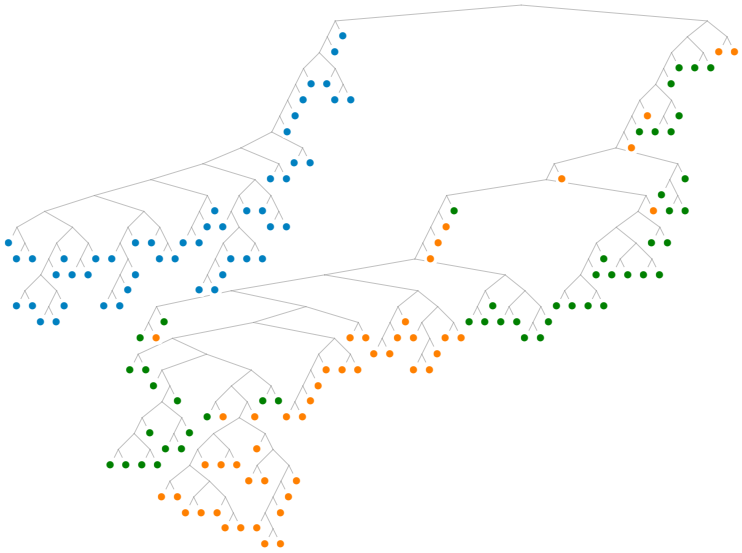


Figure 6.13: Clustering tree representation of the sequential grouping of the Iris flower data observations. The grouping is started from the bottom of the tree, and finally the two main groups are joined into one at the top of the tree. The colors of the points are the same as in Fig. 6.8

Chapter 7

Bayesian inference

7.1 Introduction

Bayesian inference (BI) gives the theoretical basis to Bayesian (statistical) methods the same way as frequentist (statistical) inference is the basis for frequentist (statistical) analysis. There are some philosophical and technical differences between frequentist (i.e. classical) and Bayesian approaches, but actually many parts of the inference are done similarly.

The philosophical difference is in the way the unknown parameters are interpreted. In frequentist inference the parameter is an unknown but a fixed constant, while in BI the parameter itself is a random variable. In what follows we do not concentrate on the philosophical differences that much, but give guidance on the technical procedure and theory behind BI.

The one formula behind the whole Bayesian standpoint is, of course, the Bayes formula as in Eq. (1.9). In parameter estimation, the idea is to use Bayes formula as:

$$\mathrm{P}(\text{parameters}|\text{data}) = \frac{\mathrm{P}(\text{parameters})\,\mathrm{P}(\text{data}|\text{parameters})}{\mathrm{P}(\text{data})} \tag{7.1}$$

Let us write it here for continuous variables using pdf's:

$$\mathrm{f}_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{\mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\,\mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\mathrm{f}_{\boldsymbol{Y}}(\boldsymbol{y})} = \frac{\mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\,\mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})}{\int_\Omega \mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\,\mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\,d\boldsymbol{\theta}} \tag{7.2}$$

We explicitly write out here the random variables the different pdf's are referring to, but in what follows we will often shorten it, e.g. $\mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta}) = \mathrm{f}(\boldsymbol{y}|\boldsymbol{\theta})$.

From the way Eq. (7.2) is written, one can immediately recognize the application to parameter estimation. The left side is the pdf of the unknown parameter vector $\boldsymbol{\theta}$, given that we have observed the data $\boldsymbol{y}$. The left side is called the posterior distribution of the parameters. The numerator of the right side(s) is from the chain

rule, it has both the prior distribution for the parameter, $\mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$ and the distribution of data given the parameters, $\mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$.

An important point in BI is that the denominator of Eq. (7.2) is often unnecessary to be known. The denominator is the (unconditional) distribution of the data. Definition $\mathrm{f}_{\boldsymbol{Y}}(\boldsymbol{y}) = \int_{\Omega} \mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, \mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{\theta}$ uses the formula of total probability and integrates over the parameter space Ω. However, the denominator is constant with respect to θ. In fact, the role of the denominator is only to scale the resulting formula to a proper pdf, i.e., to ensure that the area of the pdf is one, $\int \mathrm{f}_{\boldsymbol{\Theta}|\boldsymbol{Y}}(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta} = 1$.

In many applications the knowledge of a properly scaled posterior distribution is not important. If you compare to the task of maximum likelihood parameter estimation with frequentist approach, one is only interested of the maximization of $\mathrm{f}(\boldsymbol{y}; \boldsymbol{\theta})$, i.e. the probability density of data with given parameter value $\boldsymbol{\theta}$ (so, classical $\mathrm{f}(\boldsymbol{y}; \boldsymbol{\theta})$ equals Bayesian $\mathrm{f}(\boldsymbol{y}|\boldsymbol{\theta})$). In comparable BI case it is enough to know the unscaled posterior, $\mathrm{f}_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\, \mathrm{f}_{\boldsymbol{Y}|\boldsymbol{\Theta}}(\boldsymbol{y}|\boldsymbol{\theta})$. There is even closer connection to classical inference — if unscaled posterior is enough, we can use the likelihood function instead of the pdf. So, the version of the Bayes formula that is usually applied in BI is

$$\mathrm{f}(\boldsymbol{\theta}|\boldsymbol{y}) \;\propto\; \mathrm{f}(\boldsymbol{\theta})\, \mathrm{f}(\boldsymbol{y}; \boldsymbol{\theta}) \;\propto\; \mathrm{f}(\boldsymbol{\theta})\, \mathrm{L}(\boldsymbol{\theta}; \boldsymbol{y}) \tag{7.3}$$

7.2 Prior distributions

When comparing Eq. (7.3) to traditional maximum likelihood problems, one can see that the main difference is the presence of the prior distribution. Selecting a priori pdf is a subjective decision that should be somehow justified by the researcher. In principle, any pdf can be used as a priori pdf, or the prior does not even need to be a proper pdf, but there are some common approaches to the problem.

7.2.1 Conjugate prior distributions

Especially in times before efficient computers and easy-to-use software, the concept of conjugate prior (*liitännäispriori*) was important, since it allowed analytical, closed-form formulas to be derived. In short, a conjugate prior $\mathrm{f}(\boldsymbol{\theta})$ is such a distribution that the posterior $\mathrm{f}(\boldsymbol{\theta}|\boldsymbol{y})$ has the same distribution family as the prior. The selection of a conjugate prior is always related to the probability model of the data, $\mathrm{f}(\boldsymbol{y}|\boldsymbol{\theta})$.

The attractive benefit in using conjugate prior is that the results can be easily computed and interpreted, and the influence of both the data and the choice of parameters of prior distribution, i.e. *hyperparameters*, to the posterior parameters is clear. For example, if we conduct n independent Bernoulli trials with parameter (probability of success) π, and receive k positive outcomes, the likelihood model for the data is $\mathrm{L}(\pi; k) = \pi^k (1-\pi)^{1-k}$. Now, the Beta distribution is the conjugate

prior for Bernoulli data. That means that if $\pi \sim \mathcal{B}(\alpha, \beta)$, then the posterior is also Beta-distribution but with some other parameters. Without calculating anything ourselves we can check from literature that the posterior is

$$\pi \,|\, k \sim \mathcal{B}(\alpha + k, \beta + n - k) \tag{7.4}$$

The complete Bayesian analysis of the case is now done and the result is compressed into the distribution and its parameters. With conjugate priors it can be straightforward to interpret the effect of data and hyperparameters in the prior distribution to the properties of the posterior distributions. For example, the expected value *a priori* in the Bernoulli-Beta example is $\mu_{\text{prior}} = \frac{\alpha}{\alpha+\beta}$. After the data has been collected, the *a posteriori* expected value is $\mu_{\text{posterior}} = \frac{\alpha+k}{\alpha+\beta+n}$.

The simpleness of the conjugate prior approach is at the same time its shortcoming. The subjective choice of prior distribution is the key point in BI. In this era of efficient computing tools a conjugate prior should be used only if the prior would suit the case anyway, not just because the result is easy to derive and interpret. Lists of likelihood models with their prior distributions can be found in the literature, for example in Wikipedia. For the most common model of normal likelihood the prior distribution for the expectation parameter μ is also the normal distribution, and for variance σ^2 it is the inverse gamma distribution.

7.2.2 Uninformative prior distributions

Another common approach, or rather a framework of approaches, is the use of uninformative or vague priors. This means that if the researcher does not have any particular information of the parameter *a priori* the observations, the uncertainty should be described in the prior. The idea is straightforward, but the practice might not be so simple to implement.

It is easy to think that if there is no knowledge of the location parameter, μ for normal model for example, all the values of μ should be equally probable, $\mathrm{f}(\mu) \propto c$. So, the uninformative prior for μ should be the uniform distribution.

The first immediate problem is that the uniform distribution over the real axis is not a proper distribution since it does not integrate to one, it is a so-called improper prior. If the prior distribution is improper, the posterior is often also an improper distribution. However, in many BI analysis this problem can be avoided by using the form in Eq. (7.3) and deriving computational results by Monte Carlo or Markov chain Monte Carlo sampling. The recommended uninformative prior for scale parameter (i.e., variance) is of the form $\mathrm{f}(\sigma^2) \propto 1/\sigma^2$

If the improper prior is not a problem, the reparametrization of the model might arise new problems. Reparametrization means that the original parameter of the model is transformed by some function. In many physical models it is possible to change from one set of parameters to another. For example astronomical coordinates can be defined in several ways. The reparametrization will also transform the

shape of the prior distribution. It can easily happen that 'uniform' distribution in one parametrization will transform into something quite non-uniform in another parametrization.

It can be thought that the prior information should be invariant under parameter transformations. The prior that implements this principle is the Jeffreys prior. It has the form

$$\mathrm{f}(\boldsymbol{\theta}) \propto \sqrt{\det(\mathbf{I}(\boldsymbol{\theta}))}, \tag{7.5}$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the so-called Fischer information matrix for the parameter $\boldsymbol{\theta}$. The Fischer information is the expectancy of the Hessian matrix $\mathbf{H}$ of the models second partial derivatives mentioned in Eq. (2.6). While Jeffreys prior solves the reparametrization problem, it is not always evident if the Jeffreys prior will describe the uncertainty in a meaningful way. With normal distribution and location parameter this is not the case, since the Jeffreys prior for that model is $\mathrm{f}(\mu) \propto c$.

Other common choices for uninformative priors, or at least for priors with very small amount of information, are proper distributions with very large variances so that they are 'almost flat' but still integrate to one. For example, with normal likelihood model the normal distribution itself is a conjugate distribution for the expectancy μ. If normal distribution has hyperparameter (i.e., parameter of the distribution of the parameter) σ_0^2 that is very large, the prior is almost flat but the posterior is still a proper normal distribution.

7.2.3 Informative or subjective prior distributions

A criticism towards the use of uninformative priors is that, first, sometimes it can be difficult to actually express the lack of information as seen above. Second, BI with uninformative priors will actually give more or less the same result as the traditional frequentist approach since the results will only depend on the likelihood function of the data. Third, choosing an uninformative prior is also a subjective choice. Therefore, the most rewarding case for BI is when there actually is *a priori* information about the parameter and when that information can be represented in the form of a (prior) distribution.

With the subjective choice of the prior distribution, a sensitivity analysis would often be a good idea. If the variance of the prior pdf is small, a lot of observations are needed to shift the posterior estimate away from the prior. The sensitivity of the posterior to observations is weak. If the variance of the prior is large, already a few observations can overdrive the prior information in the posterior, and the sensitivity to observations is strong. Often it needs some numerical tests to assure that the sensitivity is on the right level. An example of two priors, observations, and posteriors is shown in Fig. 7.1.

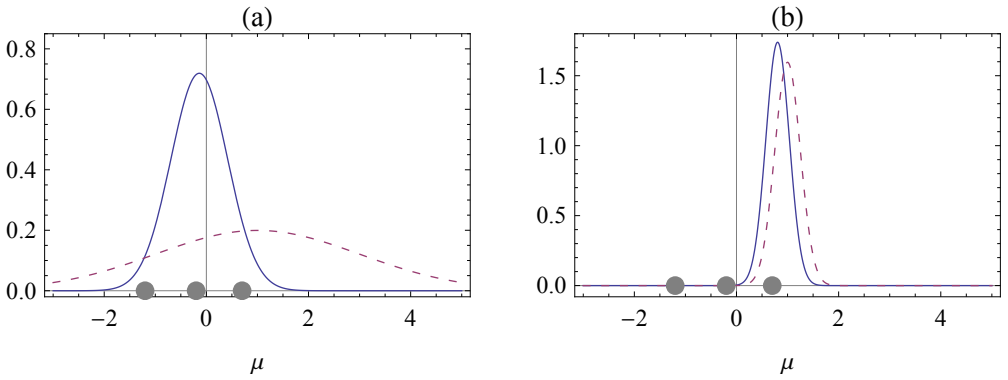


Figure 7.1: Three observations from normal distribution, normal prior (red dashed line) and posterior (blue solid line) for the parameter μ. In (a) the prior variance is large, and in (b) it is small.

7.3 Parameter estimation

Derivation of the point-estimates to the (unknown) model parameters $\boldsymbol{\theta}$ within Bayesian framework is based on either Eq. (7.2) or Eq. (7.3). There are three common choices for parameter estimate $\hat{\boldsymbol{\theta}}$: the posterior median, the posterior mean, and the maximum a posteriori (MAP) estimates. The analytical derivation of posterior median and mean estimates require the knowledge of the proper posterior distribution (Eq. (7.2)), because e.g. the posterior mean is calculated as

$$\text{Posterior mean } \hat{\boldsymbol{\theta}} = \int_{\Omega} \boldsymbol{\theta}\, \mathrm{f}(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{7.6}$$

With Markov chain Monte Carlo (MCMC) methods we will see that the explicit formulation of the proper posterior distribution is not always necessary, and posterior mean or median estimates can be computed from samples.

With MAP estimate, however, the proper form of posterior distribution is not needed. Maximization of Eq. (7.2) can be equally well done using only Eq. (7.3). Note the similarities with the MLE estimate which is computed in the similar manner, only without the prior distribution.

The fact that there are three equally justified and popular methods for parameter estimation in Bayesian framework is somehow typical for BI. There is a certain amount of subjectivity in every Bayesian analysis, and the best practice is to write out all the choices made, so that other researchers can reproduce the results and follow the formulations if needed.

7.3.1 Bayesian interval estimation

With frequentist ML inference the uncertainty about the ML estimate is described with confidence intervals. The similar construction in BI is the credible interval.

Because in BI it is natural to speak about probability of the parameter, the credible interval is defined as

$$\mathrm{P}(\Theta \in (\theta_1, \theta_2) \,|\, \boldsymbol{y}) = \int_{\theta_1}^{\theta_2} \mathrm{f}(\theta|\boldsymbol{y})\, d\theta = 1 - \alpha \tag{7.7}$$

The problem with the equation above is that it does not define the limits θ_1 and θ_2 unambiguously. There are two different extra conditions that can be used to define the interval properly. The first one is the *equal tail credible interval* where we require that the tail probabilities are the same:

$$\int_{-\infty}^{\theta_1} \mathrm{f}(\theta|\boldsymbol{y})\, d\theta = \int_{\theta_2}^{\infty} \mathrm{f}(\theta|\boldsymbol{y})\, d\theta = \frac{\alpha}{2} \tag{7.8}$$

The second possibility is that we require the posterior densities inside the credible interval to be larger than any density value outside the interval. This is called the *highest posterior density region*:

$$\begin{gathered} \theta_1 \text{ and } \theta_2 \text{ so that } \quad \mathrm{f}(\theta|\boldsymbol{y}) \geq \mathrm{f}(\theta^*|\boldsymbol{y}), \\ \text{when } \theta_1 \leq \theta \leq \theta_2 \text{ , and } \theta^* < \theta_1 \text{ or } \theta_2 < \theta^* \end{gathered} \tag{7.9}$$

For symmetric unimodal distribution these intervals will coincide.

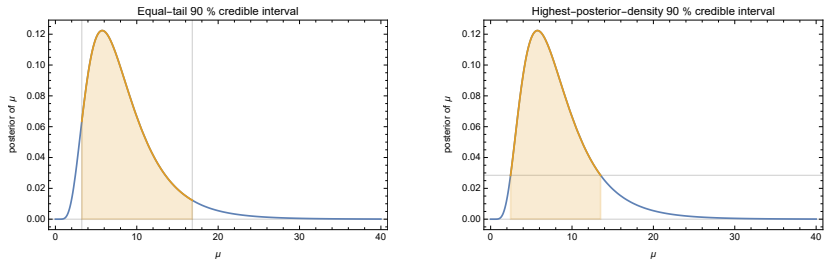


Figure 7.2: Equal-tail (on the left) and highest posterior density (on the right) 90 % credible intervals for parameter μ when its posterior density is log-normal.

Chapter 8

Monte Carlo methods

Monte Carlo (MC) methods are statistical methods that are based on the computer-generated random numbers. Random numbers can be used directly to asses some features of complicated random model, or they can be used in drawing randomized samples of existing data. The latter case is called resampling. When Monte Carlo is used to create the so-called Markov chains, the method is called MCMC and is presented in next chapter. In any case, the inference with MC (or MCMC) is based on the averaged descriptive statistics of the data that results from the MC procedure.

8.1 Random number generation

Before MC methods can be used, we must have procedure that can generate random numbers from the desired distribution. In some cases the probability model can consist of an algorithm that is difficult to describe with parametric distribution. In that case, the algorithm itself can be used to create samples that obey the unknown distribution with some random input to the algorithm. In the common case, however, we know the parametric distribution from which we want to create random numbers.

Even the creation of uniform random integers is somewhat complicated if we want the *pseudorandom numbers* to come from a sequence that will seem random. First of all, the length of the period, i.e. the length of unique sequence of integers, should be large. Second, the integers should pass any test of uniformity. Third, there should not be detectable autocorrelation in the sequence. All the uniform number generators in modern computing environments should be reliable nowadays. For example, the Mersenne twister algorithm was developed in 1997 and has very good random properties. However, the rise of parallel computing has new challenges for random number generators, since also the parallel threads of the code should have random sequences that are uncorrelated with the other threads.

The pseudorandom number generators always generate random integers. The conversion to uniform real numbers between 0 and 1, $U \sim \mathcal{U}(0,1)$, is done by dividing the random integer by the largest possible integer (plus one) in the random sequence. Usually the generators can return the lower limit of 0.0 (naturally, with very low probability), but not the upper limit of 1.0.

Since the uniform distribution should be well implemented in almost all systems, and it is hard to improve the implementation yourself, we will concentrate on the creation of continuous random numbers from more complicated distributions using the uniform numbers as an input.

8.1.1 Inversion method

The most general algorithm for random numbers is the inversion method or inverse transform method. It is based on the following deduction. Let U be random number from $\mathcal{U}(0,1)$. Let us compute $\mathrm{F}^{-1}(U)$, where $\mathrm{F}^{-1}(\cdot)$ is the inverse cumulative probability function of the desired distribution. If we compute the cumulative probability of $\mathrm{F}^{-1}(U)$ being less than x, we can see that

$$\begin{aligned} \mathrm{P}\left(\mathrm{F}^{-1}(U) \leq x\right) &= \mathrm{P}\left(\mathrm{F}\left(\mathrm{F}^{-1}(U)\right) \leq \mathrm{F}(x)\right) \text{ [by applying F to both sides]} \\ &= \mathrm{P}\left(U \leq \mathrm{F}(x)\right) = \mathrm{F}(x) \end{aligned} \quad (8.1)$$

because U is uniform, so the probability of $U \leq y$ is y when y is between zero and one. A graphical example is shown in Fig. 8.1.

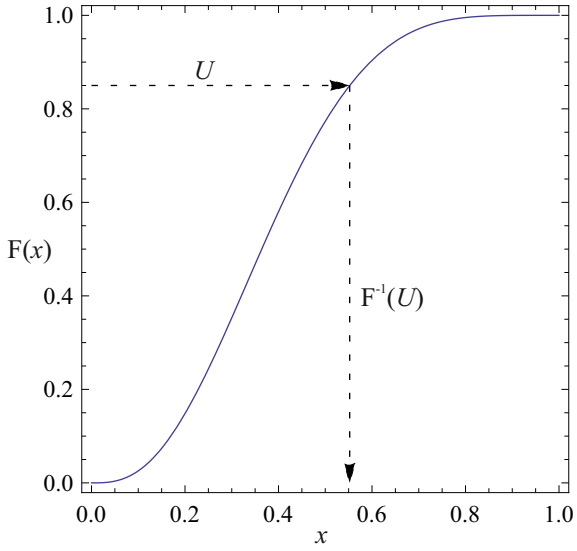


Figure 8.1: Example of the inverse transform method with Beta(3,5)-distribution.

The inverse transform method is valid, in theory, for all distributions. The problem is that the inverse cdf does not exist in closed from for all the distributions, for

example the cdf and the inverse cdf for normal distribution are not closed-form functions.

8.1.2 Normal distribution

Random numbers from the standard normal distribution can be generated with the special transformation, the Box-Muller algorithm. For two uniform numbers U_1, U_2 it holds that the transformation

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2), \quad X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2) \tag{8.2}$$

produces X_1 and X_2 that are independent and have standard normal distribution. In Sec. 6.1.1 we introduced how to create correlated values from multinormal distribution, but for two-dimensional multinormal distribution there is a shortcut with the Seppo Mustonen -algorithm. It is the same transform to X_1 as the Box-Muller, but X_2 is computed by

$$X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2 + \arcsin(\rho)), \tag{8.3}$$

where ρ is the correlation coefficient between X_1 and X_2.

Other special transforms exists, and some of them are based on the way the distribution is originally derived. For example, as we know that sum of squared normal variables has χ^2-distribution, random numbers from χ^2 can be derived simply just by first creating normal random numbers and the summing their squares. Often these kind of transformations are inefficient when the parameters require a lot of source variables per one outcome.

8.1.3 Accept-Reject method

The Accept-Reject method is based on the creation of random coordinates uniformly inside an area (in 2-D) that bounds the pdf of the target distribution. If the coordinate is inside the area bounded by the target pdf, it's x-coordinate is accepted as a random number from the distribution. If not, it is rejected and a new coordinate is created.

The most simple application of the accept-reject method is the 'box-counting' version where random coordinates are created inside the rectangular area that holds the target pdf inside. For this to work, the target pdf must have finite support. For example, the Beta distribution is defined between 0 and 1 — example of box-counting accept-reject algorithm for Beta distribution is shown in Fig. 8.2.

The box-counting comes less effective in multiple dimensions and in cases where the support of the distribution is very wide, because the number of the rejected point grows. The effectiveness can be improved by finding an envelope that has

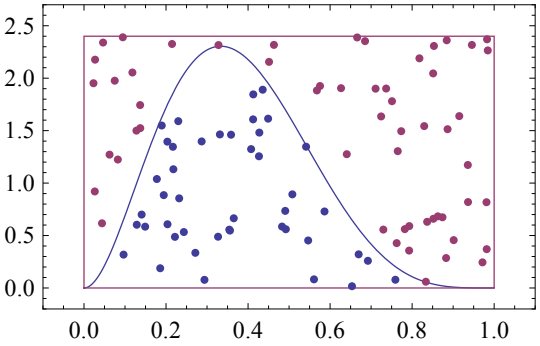


Figure 8.2: Beta(3,5)-distribution and box-counting accept-reject algorithm. The x-coordinates of the blue points have the desired distribution.

smaller reject-area outside the target pdf than the rectangle. In general, any 'envelope pdf' $g(\cdot)$ can be used in accept-reject method, if only we can find constant c so that

$$\mathrm{f}(x) \leq c\,\mathrm{g}(x) \quad \forall x \tag{8.4}$$

If this condition can be fulfilled, the X that is generated from the envelope pdf g can be accepted if

$$U \leq \frac{\mathrm{f}(X)}{c\,\mathrm{g}(X)}, \tag{8.5}$$

where U is from $\mathcal{U}(0,1)$, and rejected otherwise. The box-counting is a simple version of this where the envelope is also a uniform distribution, so that $c\,\mathrm{g}(x) = c$. If the envelope is very close to the target distribution, only a small fraction of the random numbers must be rejected. For the envelope method to work we naturally need to have such a distribution g that it is easy to create random numbers from it.

The Gamma distribution is one example of a distribution that can be simulated by the envelope accpet-reject algorithm efficiently. The trick is that Gamma(α,β)-variables are easy and fast to create if α is an integer. For other α, the Gamma distribution with integer α can be used as an envelope with suitable choice of β and constant c. Example is shown in Fig. 8.3.

8.2 Resampling methods

The resampling methods are procedures that recycle the existing data in some random manner, i.e. draw random (re)samples of the data. If the original sample is a good representation of the unknown sample space, then the random resamples also estimate the properties of the sample space well. Resampling methods have troubles with small and biased samples, but then again, this is true for more or less all the statistical methods. We will introduce bootstrap, permutation tests and cross-validation here. The so-called jackknife is also a resampling method for vari-

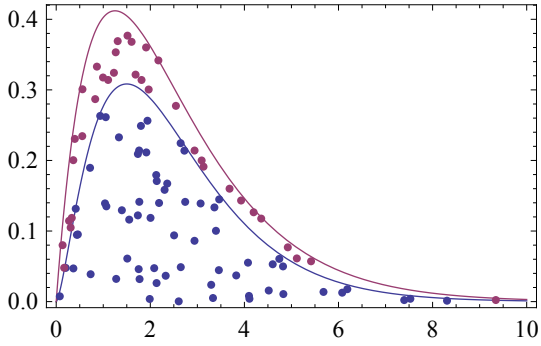


Figure 8.3: Gamma(2.5,1) distribution and a suitable envelope for envelope accept-reject algorithm.

ance estimation, but the bootstrap is more general and preferable in many cases, so the jackknife method is not dealt here.

8.2.1 Bootstrap

The bootstrap method was developed by Bradley Efron in 1979 as an extension to jackknife. The name refers to phrase "pull oneself up from one's bootstraps", and suits the method quite well. The initial situation for bootstrap is that we have only one random sample of the interesting phenomena, $\boldsymbol{y}$, and no other information. However, the sample should represent the total sample space. If so, we could draw new samples $\boldsymbol{y}_i^*$ from $\boldsymbol{y}$, and they should also represent the sample space. These resamples should be drawn *with replacement* from the original sample, and have the same size.

From the original sample we can compute a value for an estimator of interest, $\hat{\theta}$. With bootstrap we can asses the uncertainty, e.g. the variance or the confidence intervals, of the estimator. If we compute the same estimator value for every bootstrap sample, θ_i^*, the empirical distribution of θ_i^*'s should estimate the true distribution of $\hat{\theta}$. The inference about $\hat{\theta}$ can be made based on the empirical distribution by descriptive statistics.

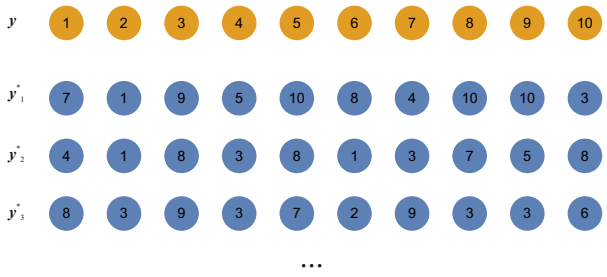


Figure 8.4: Sample with return.

For example, we have a sample of 10 numbers from the exponential distribution

in Fig. 8.5. The mean $\overline{y}$ is 1.09. Without knowing that the underlying distribution is exponential, one could compute the variance or the confidence interval to the mean using normal approximation. The resulting CI will be symmetrical about the mean. However, exponential distribution is skewed to right, and thus the real CI is not symmetrical.

The histogram in Fig. 8.5 is drawn from the 40,000 means computed from 40,000 bootstrap samples of $\boldsymbol{y}$. Their distribution is slightly skewed to right, as it should be. The bootstrap CI can be computed from the ordered values of $\overline{y}_i^*$. For 95 % CI we will take the 1,000th (2.5 %) value and the 39,000th (97.5 %) value of sorted bootstrap means, and end up with a CI of $(\overline{y}^*_{[1000]}, \overline{y}^*_{[39000]}) = (0.507, 1.879)$. This CI is shown in the figure with gray vertical lines, and it is clearly nonsymmetric around the mean with red vertical line.

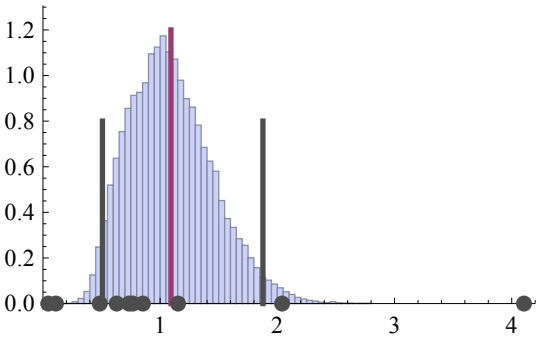


Figure 8.5: Ten samples (gray dots), their mean (red line), histogram of 40,000 bootstrap means and bootstrap CI for the mean (gray lines).

The great advantage with bootstrap is that variances or CI's can be derived for any estimator as easily as for mean, for example for median. The number of bootstrap samples should be large in order to smooth out the finite sampling effects. What is 'large' depends on the problem, but with modern computers the speed is usually not an issue, so 10,000, 50,000 or even 100,000 could be used as the number of bootstrap samples.

Bootstrap can be used also in regression problems (LM and NLM), but then the bootstrap sampling should be used for *residuals* instead of the original data. The procedure is such that first the standard LM or NLM is fitted, and estimates $\boldsymbol{b}$, fitted values $\hat{y}_i$ and residuals e_i are received. Then, bootstrap dataset is formed by adding randomly chosen residual value e_j to each $\hat{y}_i$, thus creating new dataset with $(\boldsymbol{x}_i, \hat{y}_i + e_j)$. The same regression analysis is computed, and bootstrap values $\boldsymbol{b}_k^*$ are received. This is repeated, and inference is based on the distribution of $\boldsymbol{b}_k^*$'s.

8.2.2 Cross-validation

Resampling methods are often quite simple and straightforward ideas that are easy to implement if only the computing power is not an issue. This is true with the

bootstrapping, and is especially true with the cross-validation.

Cross-validation is practical with methods involving some kind of prediction, and the accuracy of the prediction interest us. For example, LM or NLM can be used to predict the values of the dependent variable for given explanatory variable. The CI of the prediction can be computed using the residual variance of the model, i.e. the observed errors. However, the residual variance gives too optimistic estimate. The model is fitted to exactly the same observations from which the residuals are computed, and this introduces *overfitting* if we consider new observations. Actually, this can be taken into account analytically in LM, but in NLM or in general linear models this is not possible.

Another example is a classification procedure. Let us say that we have a dataset and we use that to form (i.e. train) our classification scheme. We can try to estimate the error rate the classifier does by letting it classify our training data, but again, the estimate will be too optimistic because the classification is tuned with exactly these data.

The solution is to leave out one part of the data from the model estimation, and use the model to predict the values for the left-out data. The prediction error is then computed using the errors computed with the left-out data. The usual problem is, of course, that we seldom have huge amounts of data available, and the fitted model will perform worse when estimated with smaller training data than with all the available data. The cross-validation, especially with the so-called *leave-one-out* procedure, is the best compromise between large training set and realistic error estimation. In leave-one-out, one repeatedly leaves one observation out from the training set, estimates the model, and computes the prediction error for the one left-out observation. This is then repeated for all the observations, or at least for a large number, and the mean prediction error is computed from these numbers.

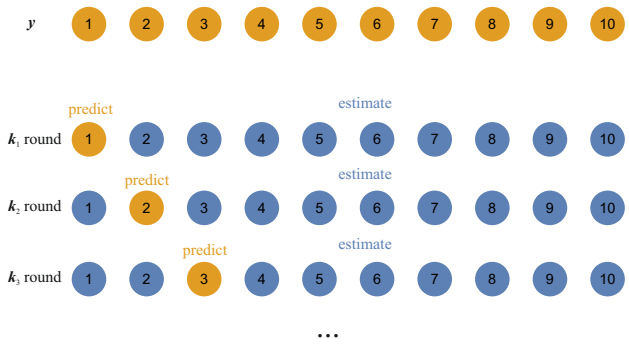


Figure 8.6: Leave-one-out crossvalidation.

Cross-validation can be done for larger dedicated data than one (k-fold cross-validation), but usually the leave-one-out is the most accurate estimate.

8.2.3 Permutation tests

Permutation tests can be used in cases where we have two or more datasets, and the null hypothesis claims that these should come from the same distribution. A test for medians can be used as an example. Let us have two sets, $\boldsymbol{y}$ and $\boldsymbol{x}$, and we want to test if the medians of the groups are the same with certain statistical significance. The null hypothesis for this test is that the medians are the same.

If the null hypothesis is correct, we could divide the data randomly into new groups $\boldsymbol{y}_i^*$ and $\boldsymbol{x}_i^*$ with the sizes $n_y$and n_x. The difference between the medians is recorded. Again, if the null hypothesis is correct, the difference between the medians between the original sets, $\hat{d} = m_y - m_x$, should be 'common' in the set of all median differences d_i^* computed from the randomly divided groups. If not, the original division was somehow 'special' and the probability of receiving such groups and such difference in median is very small. In the latter case, the null hypothesis can be rejected.

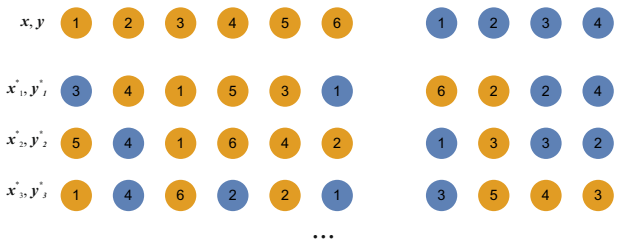


Figure 8.7: Random groupings.

The decision between 'common' and 'special' can be based on the distribution of d_i^*'s in similar manner as in traditional test theory. The p-value of the test is the proportion of d_i^*'s that are as large or larger than our $\hat{d}$. If the p-value is small, i.e. less than 5 %, the null hypothesis can be rejected.

An example of this kind of permutation test for the medians of two groups is shown in Fig. 8.8. The two datasets have both sizes of 30, and they come from exponential distributions with intensity λ of 1.0 (group 1) or 2.0 (group 2). The difference between the medians is 0.353. With 40,000 random permutations of the groups we can find that only 2.16 % of the median differences in randomly divided groups have values larger than 0.353. Therefore, the p-value for one-sided test ($H_1 : m_1 > m_2$) is 2.16 % and for two-sided test ($H_1 : m_1 \neq m_2$) 4.32 %. In both cases, the null hypothesis can be rejected — the groups do not have equal medians.

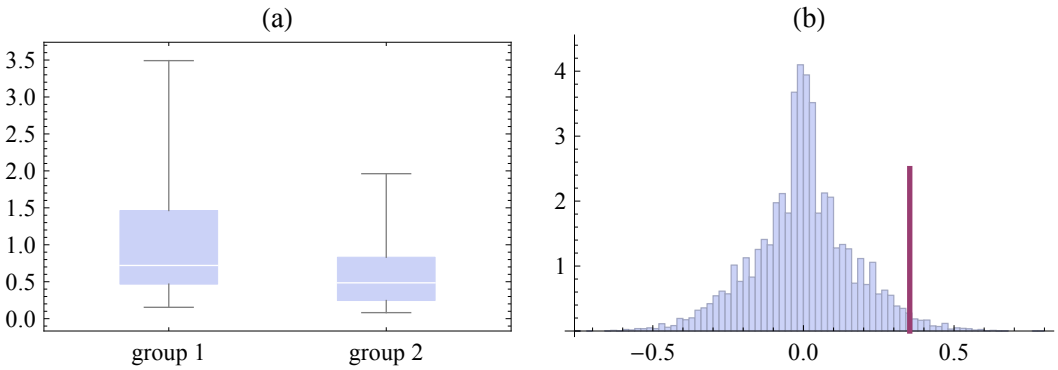


Figure 8.8: Permutation test for equal medians. Box-and-whiskers plot of the two groups in (a), and histogram of the permuted median differences in (b), where red vertical line show the observed median difference.

Chapter 9

Markov chain Monte Carlo methods

In this chapter we will continue with the Monte Carlo methods, but with a particulate family of MC metohds, that is, Monte Carlo Markov chain (MCMC). The MCMC methods have become very popular over the few recent decades with the improved power of computers, and because they offer a quite generic tool that is especially suitable for Bayesian estimation problems with no closed-form solution.

The material in this chapter is based mainly on two references, first, the book by Robert & Casella, *Monte Carlo statistical methods* (1999, Springer texts in statistics), and second, the lecture notes by Solonen, Haario, and Laine for the course *Statistical Analysis in Modeling* (2014, Lappeenranta University of Technology).

Note that in this chapter we deal quite often with some non-specified distribution, which I try to mark with the letter 'f', so the distribution is $\mathcal{F}$ and its probability density function is f. Furthermore, in most of the methods we need an auxiliary (often called instrumental/proposal) distribution marked with 'g', so distribution $\mathcal{G}$ and pdf g. And, because these distributions appear in the algorithms a lot, it is sometimes more convenient to write shortly something like $\mathrm{f}(x|y)$, which should be understood as the pdf of the conditional distribution, $\mathrm{f}_{X|Y}(x, y)$, or as the distribution $X|Y \sim \mathcal{F}(X, Y)$.

Before actually going to MCMC, I will introduce two 'regular' MC methods that have lead the way towards the MCMC and the Metropolis-Hastings algorithm.

9.1 Monte Carlo towards MCMC

Importance sampling is a technique which allows us to sample from a distribution $\mathcal{F}$ without actually being able to simulate $\mathcal{F}$ directly. The second example, the simulated annealing algorithm, is leading us to the Metropolis-Hastings algorithm in MCMC.

9.2 Importance sampling

Importance sampling is related to the accept-reject method of creating random numbers from a distribution $\mathcal{F}$ with the help of an envelope distribution $\mathcal{G}$ (see Sec. 8.1.3), but in this technique, all the proposed random numbers are accepted, only with different weights.

The algorithm is based on the fact that for any function h of the random variable X, the expected value can be computed as

$$\mathrm{E}_f[\mathrm{h}(x)] = \int \mathrm{h}(x)\mathrm{f}(x)dx = \int \mathrm{h}(x)\frac{\mathrm{f}(x)}{\mathrm{g}(x)}\mathrm{g}(x)dx. \tag{9.1}$$

In the last form we have simply multiplied and divided with the pdf g of an instrumental distribution $\mathcal{G}$. For the equality to hold, the support of g (set X of values $x \in X$ where $\mathrm{g}(x) > 0$) has to be at least the support of f.

Now the last form in the previous equation means that we can simulate a sample $x_1, \ldots, x_n$ from distribution $\mathcal{G}$, and estimate

$$\mathrm{E}_f[\mathrm{h}(x)] \approx \frac{1}{n}\sum_{i=1}^{n} \mathrm{h}(x_i)\frac{\mathrm{f}(x_i)}{\mathrm{g}(x_i)}. \tag{9.2}$$

Especially if $\mathrm{h}(x) = x$, we are estimating the mean of the distribution $\mathcal{F}$.

The importance sampling is useful when the distribution $\mathcal{F}$ is impossible or very costly to simulate from, but the distribution $\mathcal{G}$ is not. There is a practical limitation to $\mathcal{G}$, the pdf g should have 'heavier' tails than f, that is, the ratio f/g should not approach to ∞. If the tails of g would be 'lighter' than of f, then with rare cases of large $|x_i|$, the weights $\mathrm{f}(x_i)/\mathrm{g}(x_i)$ would be very large, and the variance of the mean estimator would behave badly.

An example of the importance sampling is shown here with the distribution $\mathcal{F}$ being the Fisher's z-distribution, which is the the distribution of a logarithm-transformed F-distribution (the distribution of the ratio of two independent χ^2-variables) variable. The pdf of Fisher's z-distribution is already a bit complicated:

$$\mathrm{f}(x; n, m) = \frac{2\, n^{n/2}\, m^{m/2}}{\mathrm{B}(n/2, m/2)} \frac{e^{nx}}{(ne^{2x} + m)^{(n+m)/2}}, \tag{9.3}$$

where B is the Beta function. The inverse cdf needed for the general method for creating random numbers involves the inverse of the regularized Beta function, which is not a closed-form function. Even the expected value of the distribution involves special functions.

To approximate the mean and pdf of $\mathcal{F}$, we use Cauchy distribution as the instrumental distribution $\mathcal{G}$. The inverse of the Cauchy cdf with parameters a, b is $\mathrm{F}^{-1}(u; a, b) = a + b\tan(\pi(u - 1/2))$, so it is easy to simulate. Furthermore, the tails

of the Cauchy distribution are heavy, so we can trust that $\mathrm{f}(x_i)/\mathrm{g}(x_i)$ is limited to a finite value.

The target pdf and the instrumental pdf are shown in Fig. 9.1 for Fisher's z with $n = 2, m = 10$, and for Cauchy with $a = 0, b = 1/2$. In the same figure, the sample mean $\frac{1}{n}\sum x_i \frac{\mathrm{f}(x_i)}{\mathrm{g}(x_i)}$ with Cauchy instrumental distribution is computed for five chains of simulations with n from 10 to 10,000.

Figure 9.1: In the left, the target distribution Fisher z with $n = 2, m = 10$ and the instrumental distribution Cauchy with $a = 0, b = 1/2$. In the right, the convergence of the mean of the Fisher z-distribution estimated with the importance sampling algorithm. The correct mean value (-0.237) is marked with the horizontal gray line.

By choosing different functions h, quite many different properties can be estimated with importance sampling. For example, probabilities such as $\mathrm{P}(X < c)$ (so, cfd values) can be estimated by choosing $\mathrm{h}(x) = \mathbb{I}_{x<c}$ (indicator function).

The usability of importance sampling is, however, a bit hindered by the fact that the full f needs to be known, up to any normalizing constant. And, the instrumental distribution g should be chosen so that the tails are heavier compared to f, which is not always a straightforward task.

9.3 Simulated annealing

Importance sampling can be thought as Monte Carlo integration, but simulated annealing is a method for Monte Carlo optimization. The algorithm is interesting for MCMC, since Metropolis-Hastings MCMC algorithm was developed from the simulated annealing algorithm.

Simulated annealing optimization fits well for large-dimensional global optimization problems, and does not require any information about the derivatives of the function to be optimized. The algorithm is the following:

- With the target function h to be minimized, choose x_0
- At round $i, i = 1, \ldots$
  - generate x from symmetric instrumental distribution $\mathcal{G}(x_{i-1})$

- Take

$$x_i = \begin{cases} x & \text{with probability } \min\left(\exp(-\Delta h_i/T_i), 1\right) \\ x_{i-1} & \text{otherwise} \end{cases} \tag{9.4}$$

- Repeat

The symmetric instrumental distribution $\mathcal{G}(x_{i-1})$ will give us a random movement from the previous chain value x_{i-1}. The symmetry means that both directions x and $-x$ are equally probable, e.g., $\mathrm{g}(x; x_{i-1}) = \mathrm{g}(-x; x_{i-1})$. A simple choice is the uniform distribution around the previous value, $[x_{i-1} - c, x_{i-1} + c]$. The width of the instrumental distribution must be large enough for the chain to easily move around the realistic search area, but not too large so that it will not 'shoot off' too often, which would lead to inefficiency of the algorithm.

The function $T_i := T(i)$ is the 'temperature' of the system. It must be decreasing as the simulation goes forward. It has been shown that a choice of $T(i) = k/\log(1+i)$ with sufficiently large k will guarantee good properties for the algorithm.

Finally, as the 'time' i goes forward and the 'temperature' T_i is decreasing, the probability of 'bad' moves x_i away from the (local or global) minimum of h will be more and more difficult, because $\Delta h_i = \mathrm{h}(x) - \mathrm{h}(x_{i-1})$ are scaled with T_i, so the probability limit $\exp(-\Delta h_i/T_i) \to 0$.

The global optimization is achieved by the ability of the algorithm to make 'bad' moves from time to time, making sure it can escape local minima and converge into global minimum. However, one can try to run the algorithm a few times with different starting points x_0 to make sure that the global minimum is really found.

In the example shown in Fig. 9.2 there is a quite nasty function

$$\mathrm{h}(x, y) = (y\sin(20x) + x\sin(20y))^2 \cosh(x\sin(10x)) + \cosh(y\cos(20y))(x\cos(10y) - y\sin(10x))^2 \tag{9.5}$$

with several local minima that is optimized using the simulated annealing algorithm. Two step sizes for the instrumental uniform distribution are tested, and two values for the temperature $T(i) = k/\log(1+i)$. We can see how the chain of successive values in the optimization either concentrate around the valley with the global minima, or explore more the narrow valleys away from the center, depending on the choice of parameters.

9.4 The Metropolis-Hastings algorithm for MCMC

The MCMC method is quite generally defined in Robert & Casella (1999): "A Markov chain Monte Carlo method for the simulation of a distribution $\mathcal{F}$ is any method producing an ergodic Markov chain $(X^{(t)})$ whose stationary distribution is $\mathcal{F}$." Before

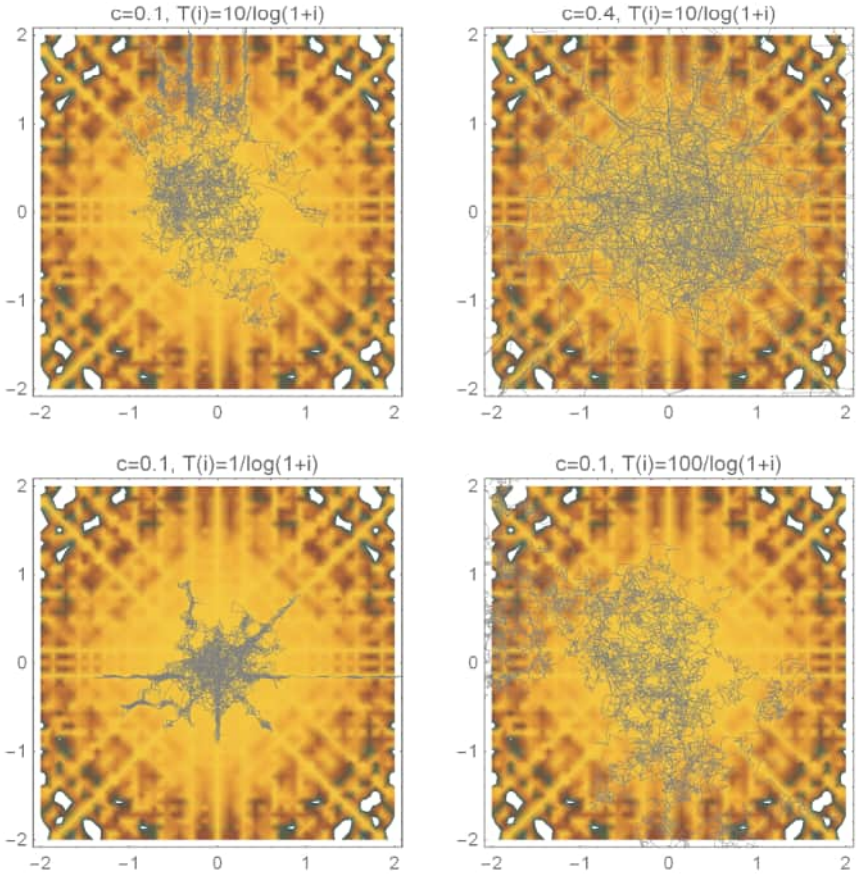


Figure 9.2: A complex function and the simulated annealing minimization chain on gray with four different combinations of step size c and temperature decrease function $T(i)$.

moving to the most popular MCMC algorithm, the Metropolis-Hastings (M-H), we will shortly introduce what is Markov chain so that we can understand a bit the definition above.

9.4.1 Markov chain

The mathematics behind Markov chains are quite involved, so to our purposes they just work, like magic. But for them to work they need to have some properties such as ergodicy, so let us try to list these required properties. First, Markov chains in MCMC are discrete random processes $(X^{(t)}), t = 1, \ldots$. The markovian property of the chain is that the next value of the chain, $X^{(t+1)}$ depends on the previous values only by the current value $X^{(t)}$.

The chain is often associated with a *transition kernel* K where $\mathrm{K}(x, y)$ marks the probability density to move from x to y in the chain. In another words, the kernel is the conditional probability distribution $\mathcal{G}$ so that $\mathrm{K}(x, y) = \mathrm{g}(y|x)$.

The ergodicy of the chain means that it will converge to a stationary distribution $\mathcal{F}$.

It can be proven that this requires the chain to be Harris recurrent, i.e., the expected number of visits of the chain to a arbitrary set A is infinite. In practice, this means that every possible value of $\mathcal{F}$ is accessible by the chain, regardless of the starting value of the chain.

If we can construct a (Markov) chain such that the limiting stationary distribution of $(X^{(t)})$ is $\mathcal{F}$, then we can estimate for any function h

$$\int \mathrm{h}(x)\mathrm{f}(x)dx = \mathrm{E}_\mathrm{f}[\mathrm{h}(x)] \approx \frac{1}{T}\sum_1^T \mathrm{h}(x^{(t)}). \tag{9.6}$$

Note that the 'x' here will usually be the unknown parameter (vector) $\boldsymbol{\theta}$ of our probability model or, in Bayesian analysis, of our posterior density $\mathrm{f}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \mathrm{f}(\boldsymbol{\theta})\mathrm{L}(\boldsymbol{\theta};\boldsymbol{y})$. From now on, we will use the symbol θ, not x.

9.4.2 General Metropolis-Hastings algorithm

To complete the magic in the previous section we need a way to construct a Markov chain that has the required properties and, most of all, has the ability to have an arbitrary distribution $\mathcal{F}$ as the limiting distribution. Such a magic can be done with the Metropolis-Hastings (M-H) algorithm. The general version of that is the following:

Algorithm *General Metropolis-Hastings*

- Choose $\theta^{(0)}$
- At round $t, t = 1, \ldots, T$
  - Generate θ from distribution $\mathcal{G}_{\theta|\theta^{(t-1)}}$
  - Take

$$\theta^{(t)} = \begin{cases} \theta & \text{with probability } \min\left(\frac{\mathrm{f}(\theta)}{\mathrm{f}(\theta^{(t-1)})}\frac{\mathrm{g}(\theta^{(t-1)}|\theta)}{\mathrm{g}(\theta|\theta^{(t-1)})}, 1\right) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$$

- Repeat

The popularity of the M-H algorithm, especially with Bayesian analysis, lies now in the ratios of f's and g's, because any common normalizing factors not depending on the parameter θ can be canceled from $\mathrm{f}(x)$'s and from $\mathrm{g}(x|y)$'s. So, in the Bayesian framework, the pdf 'f' of the target distribution can be the proportional part of the posterior distribution, $\mathrm{f}(\theta)\mathrm{L}(\theta;\boldsymbol{y})$, and the transition kernel pdf g can be anything that produces an ergodic chain.

From this M-H algorithm description one can notice the relationship to the simulated annealing algorithm. In both algorithms the 'bad' moves can be accepted

with a positive probability. In M-H, this probability value is given by the ratio $\mathrm{f}(\theta)/\mathrm{f}(\theta^{(t-1)})$. If the proposed new value θ is 'more probable', then the ratio is above one and will be accepted (well, the transition probability ratio $\mathrm{g}(\theta^{(t-1)}|\theta)/\mathrm{g}(\theta|\theta^{(t-1)})$ has to be taken also into account). But even if the ratio is below one and the proposed value is 'less probable', it can be accepted with a positive probability. The reason why the simulated annealing is not strictly a MCMC algorithm is in the decreasing temperature of the system, which makes the chain non-homogeneous, which is needed for the ergodicy.

9.4.3 Independent Metropolis-Hastings algorithm

The MCMC in general form is presented above. It seems quite straightforward with the only ambiguous part being the choice of distribution $\mathcal{G}_{x|y}$. One particular choice, leading to the so-called *independent Metropolis-Hastings*, is to have $\mathcal{G}$ so that it is not conditional, i.e., $\mathcal{G}_x$.

Algorithm *Independent Metropolis-Hastings*

- As general M-H, but select $\mathrm{g}(x|y) = \mathrm{g}(x)$. This means that the acceptance condition in general M-H simplifies into

$$\theta^{(t)} = \begin{cases} \theta & \text{with probability } \min\left(\frac{\mathrm{f}(\theta)}{\mathrm{f}(\theta^{(t-1)})}\frac{\mathrm{g}(\theta^{(t-1)})}{\mathrm{g}(\theta)}, 1\right) \\ \theta^{(t-1)} & \text{otherwise} \end{cases}$$

This algorithm is particularly similar to the importance sampling (see Sec. 9.2). The proposal (i.e., instrumental) distribution is the same, but the weights of the accepted samples are a different. In independent M-H, there are no weights in such, but some chain values are repeated (giving them larger weights) if the proposed chain movement is not accepted.

Similar consideration for the proposal distribution $\mathcal{G}$ holds with independent M-H — the g should be able to visit all the values from the support of f, and preferably with a reasonable probability (compare with tail weights f/g in importance sampling).

As an example, we repeat the sampling from Fisher's z-distribution with $n = 2$, $m = 10$ as in Sec. 9.2, and use the Cauchy distribution with $a = 0$, $b = 1/2$ as the proposal distribution. One important benefit of MCMC is that now the chain $(\theta^{(t)})$ should have the distribution $\mathcal{F}$ without any weights. This means that we can not only estimate any function $\mathrm{h}(\theta)$ with $\theta \sim \mathcal{F}$, but also the distribution $\mathcal{F}$ itself.

To speed up the computation, we can clean $\mathrm{f}(x)$ for the Fisher z-distribution from anything not depending on x, so we can use $\mathrm{f}(\theta; n, m) \propto \exp(n\theta)\,(m+\exp(2\theta)\,n)^{\frac{1}{2}(-n-m)}$.

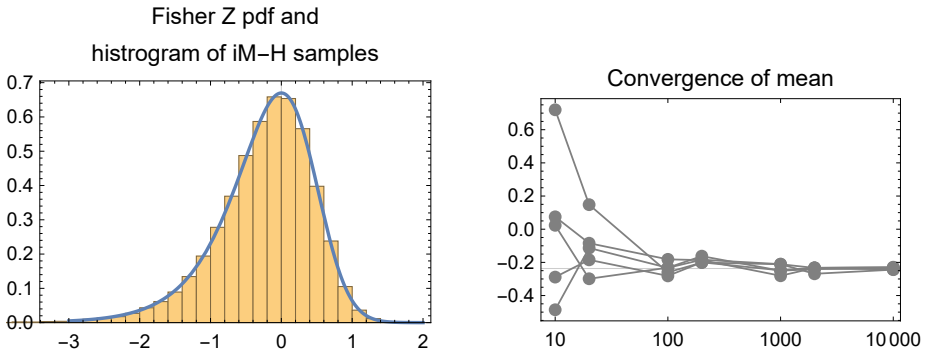


Figure 9.3: In the left, the target distribution Fisher z with $n = 2, m = 10$ and a histogram of one 100,000 sample independent M-H chain using the Cauchy$(0, 1/2)$ as the proposal distribution. In the right, the convergence of the mean of the Fisher z-distribution estimated with the independent M-H. The correct mean value (-0.237) is marked with the horizontal gray line.

9.4.4 Random walk Metropolis-Hastings

Probably the most used version of the M-H algorithms is the random walk Metropolis-Hastings. This is because the construction of the algorithm does not require detailed knowledge of the target distribution $\mathcal{F}$. In fact, one basically needs only to know the support of $\mathcal{F}$, and the proportional form $\mathrm{f}(x) \propto \mathrm{f}^*(x)$.

Algorithm *Random walk Metropolis-Hastings*

- As general M-H, but with symmetric random-walk proposal distribution g:

$$\mathrm{g}(\theta|\theta^{(t-1)}) = \mathrm{g}(\theta - \theta^{(t-1)}) = \mathrm{g}(\theta^{(t-1)} - \theta) = \mathrm{g}(\theta^{(t-1)}|\theta)$$

- The acceptance probability simplifies to

$$\min\left(\frac{\mathrm{f}(\theta)}{\mathrm{f}(\theta^{(t-1)})}\frac{\mathrm{g}(\theta^{(t-1)}|\theta)}{\mathrm{g}(\theta|\theta^{(t-1)})}, 1\right) = \min\left(\frac{\mathrm{f}(\theta)}{\mathrm{f}(\theta^{(t-1)})}, 1\right)$$

In practice, the symmetric g can be implemented as a random movement from the previous chain value, i.e., random walk. With a small random number ξ, the random-walk g is such that

$$\theta = \theta^{(t-1)} + \xi. \tag{9.7}$$

The random number ξ can be drawn from, e.g., uniform distribution over $[\theta^{(t-1)} - c, \theta^{(t-1)} + c]$, or from Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

Let us go through the use of random walk M-H in Bayesian framework. Let us model the photon count in a detector with an exponential distribution $\mathcal{E}xp(\lambda)$. The parameter λ describes the number of events (i.e., photons arriving) on a unit time interval. Furthermore, we have some previous knowledge saying that the distribution of λ could be modeled with the log-normal distribution $\mathcal{LN}(\alpha, \beta)$ with $\alpha = 1.5$ and $\beta = 0.75$.

The posterior distribution $f(\lambda|\alpha,\beta) \propto \mathrm{f}_p(\lambda;\alpha,\beta)\mathrm{L}(\lambda;\boldsymbol{x})$, where f_p is the prior distribution (log-normal), and L is the likelihood function of exponential-distributed data vector $\boldsymbol{x}$. In this example the posterior could be solved analytically, but it is not in the form of any 'common' distribution. However, the estimates such as the posterior mean cannot be solved analytically.

With data $\boldsymbol{x} = (0.254, 0.360, 0.0372, 0.340, 0.252, 0.105, 0.111, 0.222, 0.162, 0.0307)$ the prior distribution and the likelihood-function (correctly scaled to a proper pdf) are shown in Fig. 9.4. Now, using the random-walk M-H algorithm we can create a chain of values $(\lambda^{(t)})$ that should have the correct posterior distribution. One example chain is shown in Fig. 9.5.

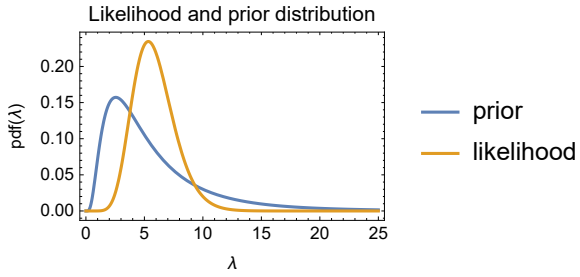


Figure 9.4: The likelihood function for the exponential model, and the a priori pdf with log-normal distribution for the parameter λ.

9.4.5 MCMC diagnostics

The M-H algorithm with suitable proposal distribution should converge to the target distribution $\mathcal{F}$ if you have infinite time. However, with finite (computer) time, you should somehow make sure that your results have already converged. There is no definite proof for that, but some steps of checks that you should at least make.

First of all, you should try different starting values for the chain, i.e., run several chains. There is a so-called *burn-in period* with the random chains when the chain starts from an initial point and finds its way towards the target distribution. This burn-in period should be discarded from the data when doing analysis based on the chain values. In Fig. 9.6 there are three chains with different starting points. One can see that the chains approach to same distribution only after some, say 3,000, steps.

The mixing of values is one interesting property to be followed with the convergence. It means that the chain should visit different values and ranges often enough. This also means that the *acceptance ratio*, the ratio of accepted movements to all the proposed movements, should not be too low (chain stuck) or too high (chain still converging). In Fig. 9.7 one can see three chains with different random walk step sizes c. The acceptance ratios are 99 % ($c = 0.1$), 97 % ($c = 0.25$), and 87 % ($c = 1$).

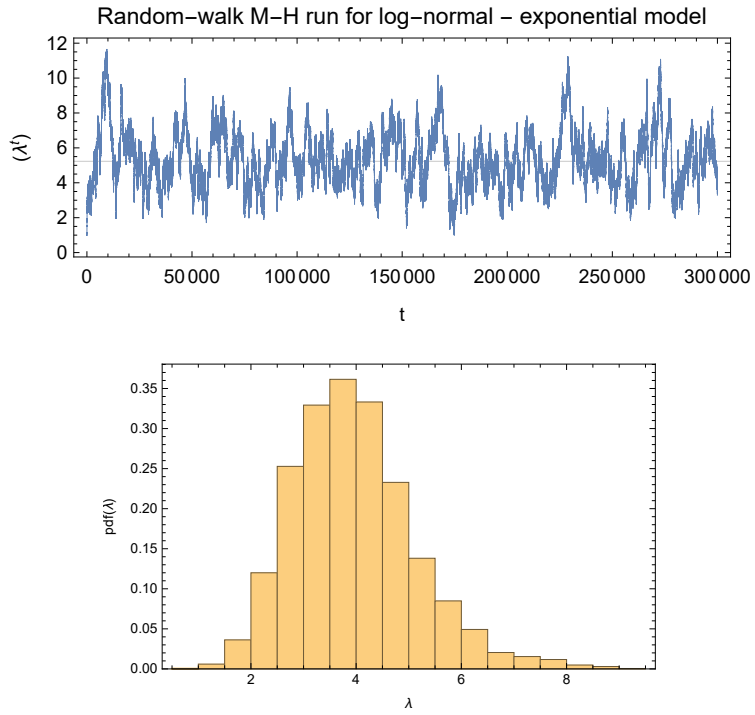


Figure 9.5: On top, the random-walk M-H chain for the parameter λ with 300,000 samples. The correct mean of 5.23 is shown in the figure with the gray horizontal line. On bottom, the histogram of the chain λ values as the estimate of the unknown target distribution.

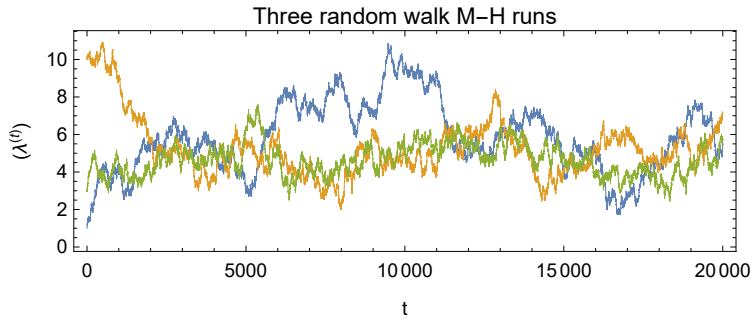


Figure 9.6: Three chains with different starting points for random walk M-H.

The smallest step size causes too slow mixing and too high acceptance ratio, while the largest step size is inefficient with too small acceptance ratio.

Other properties to follow include the chain autocorrelation, which should not be too long. The autocorrelation ρ for distance t, ρ_t, can be computed as

$$\rho_t = \frac{\left(\sum_{i=1}^{T-t}(\theta^{(i)} - \overline{\theta})(\theta^{(i+t)} - \overline{\theta})\right)/(T-t)}{\left(\sum_{i=1}^{T}(\theta^{(i)} - \overline{\theta})^2\right)/T} \tag{9.8}$$

and should approach 0 when the distance grows. The length of the chain should

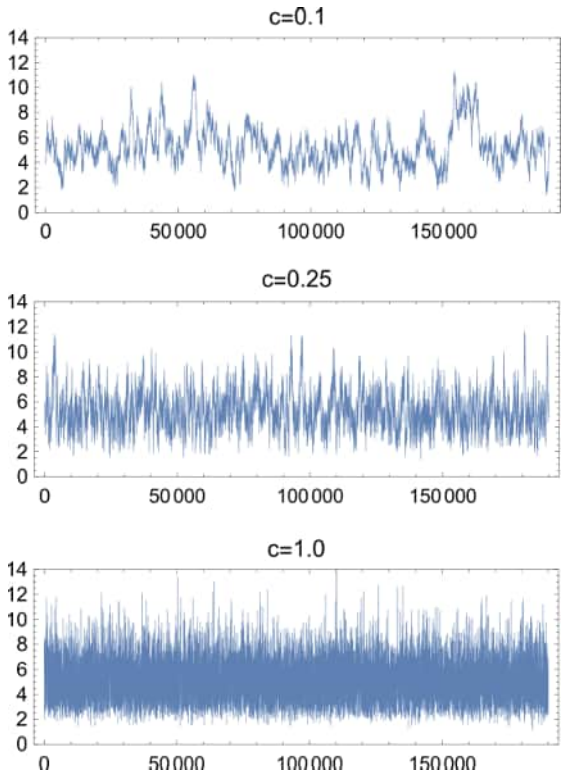


Figure 9.7: Three different step sizes for random walk M-H.

be remarkably larger than the length where its autocorrelation approaches 0. For the three chains above, the autocorrelation lengths are shown in Fig. 9.8. For step size $c = 0.1$ the autocorrelation is above 0 for up to 8,000 samples, so the autocorrelation length is quite large. The chain with $c = 1.0$ has a very short autocorrelation length, but also small acceptance ratio. The chain with $c = 0.25$ has quite reasonable autocorrelation length of ∼1,000 samples and a large acceptance ratio.

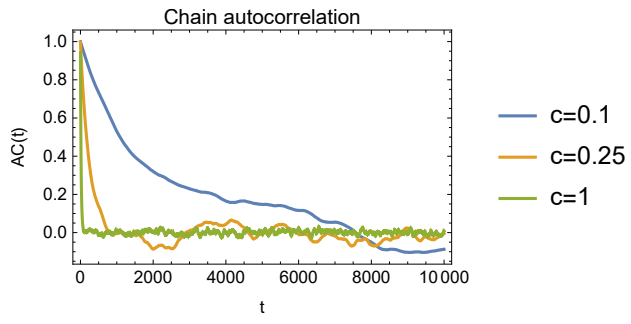


Figure 9.8: Chain autocorrelation with three different step sizes for random walk M-H.

9.4.6 Metropolis-Hastings with regression models

In regression models we have the systematic part of the regression model, let us call that $\mathrm{r}(\boldsymbol{x};\boldsymbol{\beta})$ to distinguish from target distribution f. In linear regression the function r is the linear matrix equation $\mathbf{X}\boldsymbol{\beta}$, and in nonlinear regression it is any function r.

The random part for regression comes with the residuals $\boldsymbol{\epsilon}$. If we assume these residuals to follow a normal distribution, our probability model is then

$$\boldsymbol{Y} \sim \mathcal{N}_n(\mathrm{r}(\boldsymbol{x};\boldsymbol{\beta}), \boldsymbol{\Sigma}), \tag{9.9}$$

where $\boldsymbol{\Sigma}$ is the (unknown) covariance matrix, usually assumed to be $\sigma^2\mathbf{I}$. The likelihood function for the model parameters $\boldsymbol{\beta}$ and σ^2 is now

$$\mathrm{L}(\boldsymbol{\beta},\sigma^2) = (2\pi\sigma^2)^{-n/2}\,\exp\left(-\frac{1}{2\sigma^2}\,\sum_i^n\,(y_i - \mathrm{r}(\boldsymbol{x}_i;\boldsymbol{\beta}))^2\right) \tag{9.10}$$

The prior distributions for $\boldsymbol{\beta}$ and σ^2 can be usually thought to be independent of each other, so $\mathrm{f}_p(\boldsymbol{\beta},\sigma^2) = \mathrm{f}_p(\boldsymbol{\beta})\mathrm{f}_p(\sigma^2)$. With this information we can write the posterior distribution as

$$\mathrm{f}(\boldsymbol{\beta},\sigma^2|\boldsymbol{y}) \propto \mathrm{f}_p(\boldsymbol{\beta})\mathrm{f}_p(\sigma^2)\mathrm{L}(\boldsymbol{\beta},\sigma^2). \tag{9.11}$$

In practice, the M-H update round for the vector $(\boldsymbol{\beta},\sigma^2)$ can be done *component-wise*, updating only one coordinate at time. This increases the overall acceptance ratio in problems with many coefficients β_i.

Algorithm for *component-wise* regression random walk Metropolis-Hastings (but, see forward for practical alternative version):

- Choose $\boldsymbol{\beta}^{(0)}$ and $(\sigma^2)^{(0)}$
- At round $t, t = 1,\ldots,T$
  - Initialize vector $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = \boldsymbol{\beta}^{(t-1)}$
  - Generate σ^2 from random-walk proposal distribution
  - Take

$$(\sigma^2)^{(t)} = \begin{cases} \sigma^2 & \text{with probability } \min\left(\frac{\mathrm{f}_p(\sigma^2)\,\mathrm{L}(\boldsymbol{\beta},\sigma^2)}{\mathrm{f}_p((\sigma^2)^{(t-1)})\,\mathrm{L}(\boldsymbol{\beta},(\sigma^2)^{(t-1)})},1\right) \\ (\sigma^2)^{(t-1)} & \text{otherwise} \end{cases}$$

  - For $i = 1,\ldots,k$
    - Copy current $\boldsymbol{\beta}$ to $\boldsymbol{\beta}'$. Generate component β_i' from random-walk proposal distribution
    - Take

$$\beta_i = \begin{cases} \beta_i' & \text{with probability } \min\left(\frac{\mathrm{f}_p(\boldsymbol{\beta}')\,\mathrm{L}(\boldsymbol{\beta}',(\sigma^2)^{(t)})}{\mathrm{f}_p(\boldsymbol{\beta})\,\mathrm{L}(\boldsymbol{\beta},(\sigma^2)^{(t)})},1\right) \\ \beta_i & \text{otherwise} \end{cases}$$

- Repeat
- Update $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}$

- Repeat

While the algorithm above is correct, the probabilities and their ratios in 'Take' steps might suffer from some numerical instabilities due to very small numbers being multiplied and divided. An alternative version would use log-likelihoods and log-transformed priors. For step with $(\sigma^2)^{(t)}$, let use define

$$a = \log\left(\mathrm{f}_p(\sigma^2)\right) + \mathrm{l}(\boldsymbol{\beta}, \sigma^2), \text{ and } b = \log\left(\mathrm{f}_p((\sigma^2)^{(t-1)})\right) + \mathrm{l}(\boldsymbol{\beta}, (\sigma^2)^{(t-1)}). \quad (9.12)$$

Then, testing with probability $\min\left(\exp(a-b), 1\right)$ is equal to the original test. Similarly for the step with β_i, use

$$a = \log\left(\mathrm{f}_p(\boldsymbol{\beta}')\right) + \mathrm{l}(\boldsymbol{\beta}', (\sigma^2)^{(t)}), \text{ and } b = \log\left(\mathrm{f}_p(\boldsymbol{\beta})\right) + \mathrm{l}(\boldsymbol{\beta}, (\sigma^2)^{(t)}), \quad (9.13)$$

and the same test of $\min\left(\exp(a-b), 1\right)$. For normal model, the log-likelihood is

$$\mathrm{l}(\boldsymbol{\beta}, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{S(\boldsymbol{\beta})}{2\sigma^2} \quad (9.14)$$

with $S(\boldsymbol{\beta})$ as $\sum_i^n \left(y_i - \mathrm{r}(\boldsymbol{x}_i; \boldsymbol{\beta})\right)^2$.

Advanced MCMC

There is a vast collection of different small improvements to the random walk M-H algorithm for cases where the convergence is poor using the basic form of the algorithm. *Adaptive MCMC* uses multidimensional normal distribution as the proposal distribution. The adaptation is achieved by estimating the covariance matrix of the proposal distribution from the previous values of the chain.

Other small tweaks to the proposal distribution include using a population of possible parameter values, and computing 'typical' movements somehow from them.

9.5 Gibbs sampling for MCMC

Gibbs sampling (GS) can be treated as a special case of M-H algorithm. GS is suitable for MCMC simulation of multidimensional parameter vector $\boldsymbol{\theta}$ in a stepwise manner, similar to component-wise M-H. The distinct features of GS are, that the proposed values are always accepted, and that the full conditional distributions of $\mathcal{F}$ needs to be known. By full conditional distribution we mean the pdf's

$$\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p \sim \mathrm{f}(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p) = \mathrm{f}(\theta_i|\boldsymbol{\theta}\backslash\theta_i). \quad (9.15)$$

If the full conditionals are known, the *Gibbs sampling* algorithm is:

- Choose multivariate $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$
- At round $t, t = 1, \ldots, T$
  - Update $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$
  - Loop over $i = 1, \ldots, p$
    - Generate $\theta_i^{(t)}$ from $\mathrm{f}(\theta_i | \boldsymbol{\theta}^{(t)} \backslash \theta)$
  - Repeat
- Repeat

Gibbs sampling is quite suitable for Bayesian multivariate regression problems if suitable prior distributions are selected for the parameters. For example, by selecting (multivariate) normal prior to $\boldsymbol{\theta}$, the conditional posterior distributions are (1D) normal distributions. Another typical application are the so-called *hierarchical models*.

Chapter 10

Artificial neural networks

In this section we will take a look at the quite recent field of artificial neural networks (ANNs). The rapid evolution has also led to many terms describing the field, so let us try to define some of them first. The overall top concept is artificial intelligence (AI). Inside AI, one concept is machine learning (ML), where an algorithm tries to learn features from the data. The field of ML includes many 'traditional' statistical techniques such as classification, clustering, and dimension reduction (see, e.g., PCA and LDA), but also artificial neural networks.

The ANNs have been around for already some time, but the recent advances in computing power (multi-core processors, highly parallel graphics processing units) have finally made them practical in real use. With the advances in computing, also a new concept or term, deep learning (DL), has been invented. Typically, DL should refer to the use of ANNs in the algorithm, and perhaps the word 'deep' comes from the fact that these 'deep neural networks' can have several layers of artificial neurons.

It is somewhat common that when new fields are discovered, they try to merge also existing techniques under their umbrella. This can be seen with ML and DL, largely driven by computer scientists. The traditional statistical methods are included within the ML concept. This can sometimes be quite confusing, and it might not be clear which kind of problems can be handled with ANNs. To my understanding, ANNs can be used in *supervised* learning problems. In supervised problems we need to have a set of learning data where the data has been *labeled*, i.e., each case in the learning data has the correct label/category/output attached. However, one can often encounter sources where ANNs are discussed together with *unsupervised* problems, but when looking deeper, the unsupervised problems are actually dealt with (traditional multivariate statistics) clustering techniques, and not with ANNs. There are some *semi-supervised* methods using ANNs that are between supervised and non-supervised, but in any case, ANNs need to get some kind of feedback on their performance to tune their parameters and improve.

From a very general mathematical perspective, ANNs are very flexible *non-linear*

function estimators. They can be tuned (i.e., taught) to mimic almost any kind of linear or non-linear function with multidimensional input and one-dimensional output.

10.1 Components of artificial neural networks

The main components of ANN are shown in Fig. 10.1. The figure presents a *feed-forward* ANN with two actual (i.e., *hidden*) layers of neurons or nodes.

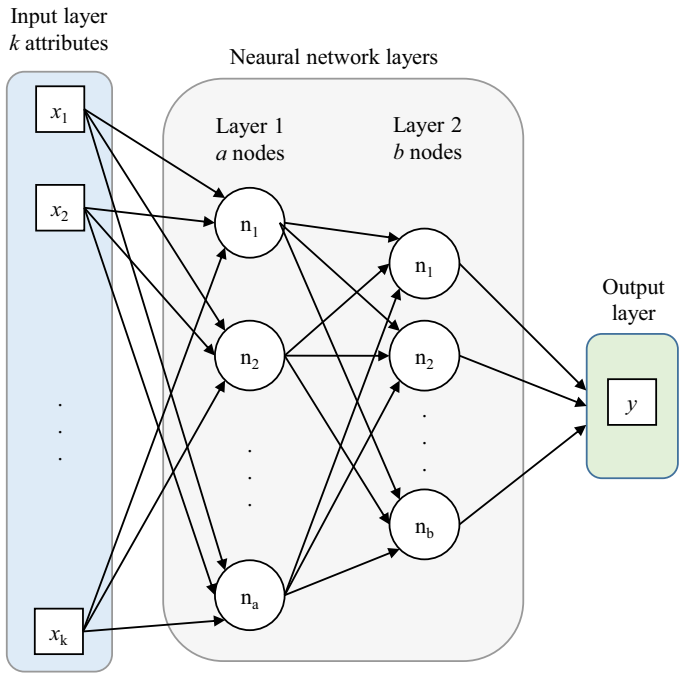


Figure 10.1: A feed-forward artificial neural network architecture with two hidden layers.

10.1.1 Nodes

The nodes or neurons in the ANN are, in general, non-linear functions with multiple input and single output. Typically, the nodes take all the outputs of the previous layers nodes as input, and produce a single, real-valued output. However, in convolutional neural networks (CNN, see later), both the input and output can be multi-dimensional tensors. But, in non-convolutional ANNs, the node is mathematically of form

$$\mathrm{n}_i(\vec{x}) = \mathrm{f}(\vec{x} \cdot \vec{w}_i + b_i) \ , \tag{10.1}$$

where $\vec{x}$ is the input from the previous layer, and f is the so-called *activation* or *propagation* function (see later).

The parameters of the ANN are the *weights* $\vec{w}_i$ and the *bias* parameters b_i. Although we are omitting the index for the layer, each node in each layer has its own weights and bias parameters. The number of weights for individual node is the dimension of the previous layer (in fully-connected network). For example, if we have input vector of length 10 and two hidden layers with dimensions of 5 and 3, we have $5 \times 10 + 3 \times 5 = 65$ unknown weights and $5 + 3 = 8$ unknown bias parameters that we need to fit (see about *training the network*). Since this example ANN would be considered as a very small network, one can see that the amount of free parameters in ANNs are easily very large.

Input and output nodes

The input and output layers do not really consist of nodes in the sense that there can not be any free parameters with the input or output nodes that would need to be estimated (i.e., trained). Both layers can, however, contain predefined operations on the data if needed.

The input nodes are just the values of the input variables. So, the dimension of the input layer is the dimension of the data vector for each case. The input nodes are sometimes called *features*, since they can be different features measured or extracted from the object to be classified. In typical ANNs, the dimension of the input vector can be large, easily some tens or hundreds.

The output layer should usually contain only one value, either a predicted numeric value (compare with regression) or a predicted class (compare with classical classification). A 'decoder' can be used with the output value in case of classification to map the numerical value into class label.

Convolutional neural networks

When dealing with images, for example, the data dimensions can get huge. In a 1024×1024 image with three color channels there are over 3 million values. If one would use fully connected layer after this kind of input layer and use, say, 100 nodes, there would be over 300 million weight parameters to be estimated with the first layer alone.

The convolutional neural networks (CNN's) tackle this problem by introducing limited 'views' or *receptive fields* from the nodes to the earlier layer data. The spatial structure of the data (e.g., image) is being exploited, and the view from a single node can take, for example, a small rectangular area around one particular pixel in the image and perform a simple operation on these pixel values. The convolution here is when the same identical *filter*, operation on the view data, is moved across the whole image to produce input for all the nodes in a layer. The CNN's also apply pooling operations which will combine multiple node/pixel values into one, therefore reducing the dimensionality.

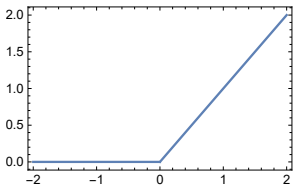
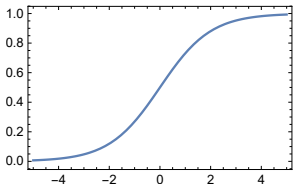
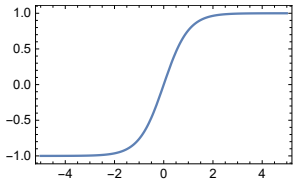
The convolutional and pooling layers in the CNN are extracting different simplified 'features' from the image. When the feature extraction phase is done, there is typically a final, fully-connected traditional ANN layer before the classification.

10.1.2 Activation functions

In each node, there is the linear part with weights and bias parameters, and the activation function f. The activation function can be any $\mathbb{R}^1 \to \mathbb{R}^1$ function, but if f is not non-linear, the network would actually just simplify into one large linear operation on the input data, and would not perform well. The other practical limitation for the activation function is that it should be differentiable, i.e., have analytical derivative. This limitation is needed when training the network (see later).

The activation functions are typically functions that take the input from the whole real axis, and limit/suppress/compress the output. In this sense, the activation function will regularize the behavior of the linear part in the node. Some of the typical activation functions are presented in Table 10.1. The ReLU will clip out all negative values, while sigmoid and tanh will compress $\mathbb{R}^1$ into $(0, 1)$ or $(-1, 1)$. Usually, the choice of the activation function will not be crucial for the ANN performance.

Table 10.1: Typical activation functions in ANNs'.

Rectified linear unit (ReLU)	$\mathrm{f}(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$	
Logistic (sigmoid)	$\mathrm{f}(x) = \frac{1}{1+e^{-x}}$	
Hyperbolic tangent (tanh)	$\mathrm{f}(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	

Special activation functions

In addition to the $\mathbb{R}^1 \to \mathbb{R}^1$ activation functions applied to each individual node, ANNs can have some special activation functions that are applied to all the nodes

collectively. Namely, these are so-called *softmax* and *maxout* functions. The purpose of these functions is in classification. The softmax function for node i is of form

$$\mathrm{f}_i(\vec{x}) = \frac{e^{x_i}}{\sum_j e^{x_j}}, \tag{10.2}$$

so the softmax scales the exponent of the output of node i with the sum of all the exponents in the output. Effectively, this convertes the outputs into something that can be treated as 'probability' — all the values are in $(0, 1)$ and they sum up to 1.

The maxout function simply outputs the maximum of all the node outputs. Alternatively, it can output the index of the node giving the maximum output.

In classification task, softmax and maxout are typically used as the activation functions of the last hidden layer before the output. If there are k classes, one should have also k nodes in the last hidden layer. After softmax activation, the output values of these nodes can be interpreted as the probabilities of the classes with given input. Then, maxout can be used to give the class with the highest probability.

10.1.3 Connections

The ANNs are typically fully-connected networks, meaning that all the outputs from the previous layer will be inputted to all the nodes of the next layer. The weights $\vec{w}_i$ (see Eq. (10.1)) in the following layer can effectively discard/nullify inputs that the ANN decides to be of no information. However, if one likes to increase the robustness of the ANN in case of possible missing data in the input, one can randomly break some connections between the nodes when training the network.

Typically, ANNs are feed-forward networks, meaning that the flow of input/output only go from the previous layer to the next layer. There are also *recurrent neural networks* where there can be loops in the graph. These loops can introduce temporal dynamic behavior, and some short-term 'memory' in the network.

10.2 Training and operation of artificial neural networks

A typical ANN operates on high-dimensional input data, and has several hidden layers between the input and output layers. As all the nodes in all the layers have the linear operation (see Eq. (10.1)) with several unknown parameters, the ANN will have a very large number of parameters to be estimated. In ML, the estimation process is usually called *learning* or *training*.

10.2.1 Training the network

The training phase of ANN is the one which consumes a lot of computations, and requires preferably a lot of training data. The training is essentially a minimization problem. The values of the unknown weight and bias parameters are estimated so that the prediction error of the network is minimized. The training data with existing labels on the cases is needed so that the prediction error can be computed.

Since the ANN is a complex non-linear system, the minimization needs to be done numerically. In each round of the minimization process, the weights are updated and the prediction error is computed. This process is accelerated with the use of gradient-based optimization algorithms, which is the reason why the activation functions need to be differentiable.

In ANNs, one round in the optimization algorithm is called *epoch*. One needs to train the network with large enough number of epochs so that the network weight and bias parameters are converged. On the other hand, training too long can result in overfitting. Overfitting is especially a problem with ANNs, since there are so many free parameters and the ANN is very flexible to fit almost anything.

As the training data can be large, the optimization round can be quite demanding computationally. Some optimization algorithms, such as Adam which is a stochastic gradient-based algorithm, are not using all the training data at each round of optimization. Instead, they take smaller *batches* from the data, and evaluate the gradient of the function adaptively based on earlier values and the value from the batch. These algorithms have parameters such as the learning rate to control how much the algorithms should trust on local (linear) gradients when moving to next value.

Because of the large number of parameters, there should be enough training data available. It is unrealistic to assume that hundreds or thousands of parameter values could be estimated very reliably with some tens of observations. This is especially the case with ANNs if one wants to gain better predictions with the use of *validation data*. The prediction error can be computed using the training data. However, this error rate is known to underestimate the true error, since the model is fitted to exactly this particular data. If part of the training data is set aside on each minimization round, the error rate when predicting this validation set can be used as a more reliable estimate. Especially when the error rate for the validation data becomes much worse than the error rate in the training data, the model might suffer from overfitting. The validation data is still different from the final *test data* (see next), so having enough training data enables to have representative sets of training, validation, and test sets. Validation data and the model performance during training can be used to tune the size and shape of the network, and the parameters in the training process, such as the number of rounds (epochs) in training algorithm and the value of the learning parameter.

10.2.2 Evaluating the performance of the network

The final evaluation of the ANN accuracy should be done using the *test data*, a part of the labeled training data that is put aside before starting the ANN learning phase. In cases where one has limited amount of training data available, one can also do k-fold cross-validation by repeatedly leaving a small part of the original data for testing, and learning the ANN several times with random division of the data.

The three possible scenarios of dividing the available data in ANN training and evaluation are shown in Fig. 10.2. In the case a), there is enough data so that it can be divided into training, validation, and testing. In the case b), the available data set is not very large, so one might want to skip the division into training and validation. Finally in the case c), there is serious lack of data. Therefore a k-fold cross-validation is done with a small testing data on each round, and by averaging the final accuracy over the runs. Please note that in the AI community, usually only the case a) with data divided into training, validation, and testing is accepted as a valid method for ANN training and use.

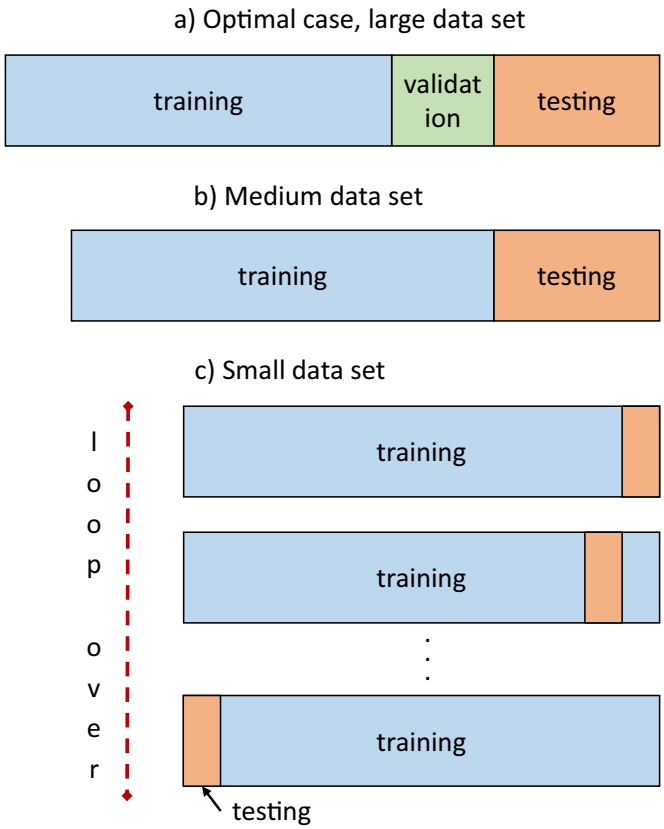


Figure 10.2: Three cases of the division of the available data for ANNs.

The final accuracy of the ANN (in classification) is the fraction of cases in the test data where the label (i.e., class) is predicted correctly by the network. However, in

many cases it might be interesting to know if different classes are predicted equally well, or if there are some problems with some particular classes. The *confusion matrix* is a popular way to present the classification results class-by-class. In confusion matrix, the correct classes are organized as rows, and the predicted classes as columns of a rectangular matrix. In Fig. 10.3 there is an example plot of a confusion matrix. One can use colors to highlight the more populated cells in the matrix. The correctly-classified cases are in the diagonal of the matrix.

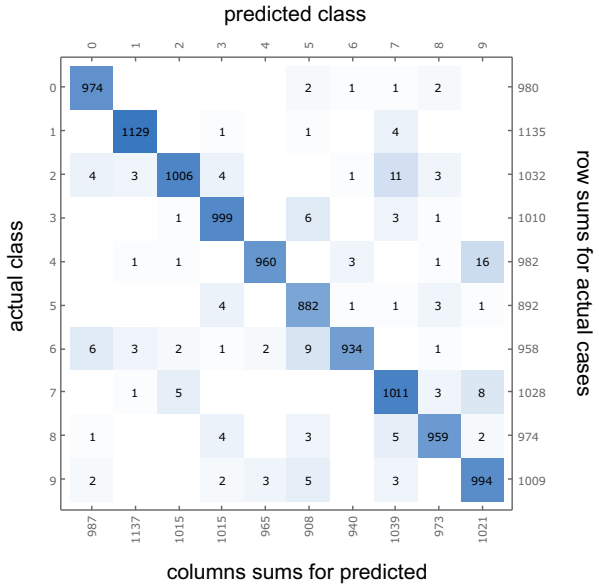


Figure 10.3: An example of a confusion matrix plot for evaluating the result of a classification task.

10.3 Computational issues with artificial neural networks

The training phase of the ANNs present a heavy task on the computing side. Gradient-based optimization is done in a task with typically hundreds of parameters and thousands of multi-dimensional data points. The computer science and hardware advances enabling the recent rise of ANNs are the use of stochastic gradient-based optimization algorithms that can be ran on graphics processing units (GPUs).

The stochastic gradient-based optimization can work with a smaller *batch* of the data on each round instead of computing the gradient using all the training data on each round. The gradient on each round will be a stochastic estimate of the true gradient, but computed a lot faster because of the smaller amount of data in the batch. One particularly popular and efficient recent algorithm for this is the Adam

algorithm, which an adaptive version of stochastic gradient and weight updating algorithm.

The GPUs come to the picture with the fact that during the network and gradient value evaluation, a large number of relatively simple operations are done over a fixed-sized batches of data. The GPUs are massively parallel but simple processors, and are suitable for this kind of tasks.

The low-level computer implementation of the optimization algorithms and the parallelization into heterogeneous CPU/GPU environment is not an easy task. Therefore there is only a handful of lowest-level libraries that implement the ANN operations on the hardware. These include TensorFlow (Google), PyTorch, mlpack, and implementations by Microsoft and Amazon. Then there are several libraries on top of these which offer an improved usability in different computing environments, such as Keras for Python, or the toolboxes in Matlab.

10.4 An example of asteroid spectral classification with artificial neural network

To further introduce the ANN field, let us have an example of the workflow of one ANN classification task. The composition of asteroids is typically estimated from their reflectance spectra in visual-near infrared wavelengths, since other methods are generally not feasible. One can divide the spectral types very roughly into silicate-rich S, carbonous C, and 'other' X category. Typical spectra are shown in Fig. 10.4.

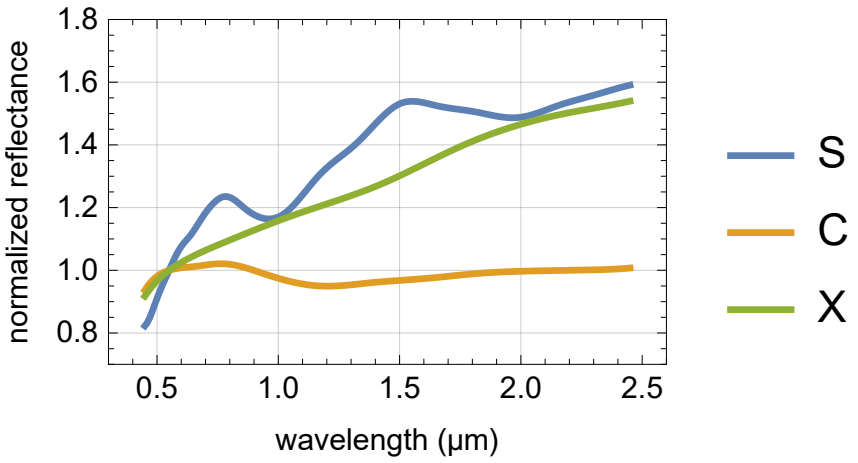


Figure 10.4: Examples of the reflectance spectra of a S-, C-, and X-type asteroids.

We have a dataset with 311 S-types, 62 C-types, and 52 X-types. The spectral resolution is such that there is always 201 values between the wavelengths of 0.45 and

2.45 μm. So, the input layer size is 201. Let us form an ANN with two hidden layers. The first layer will extract features from the spectral values, and the second layer will evaluate the probabilities for each class from these features. We have three classes, so the size of the second layer is three. Let us try having six features, at maximum, computed from the spectra, so the size of the first layer is six.

The activation function of the first layer can be sigmoid, and the second classification layer needs to have softmax activation function. The output layer might need a 'decoder' function to map the numerical labels back to class labels. The ANN could be summarized as in Fig. 10.5. Even though the network is quite small, it has $201 \times 6 + 6 + 6 \times 3 + 3 = 1233$ parameters (weights $\vec{w}_i$ and biases b_i) to be estimated.

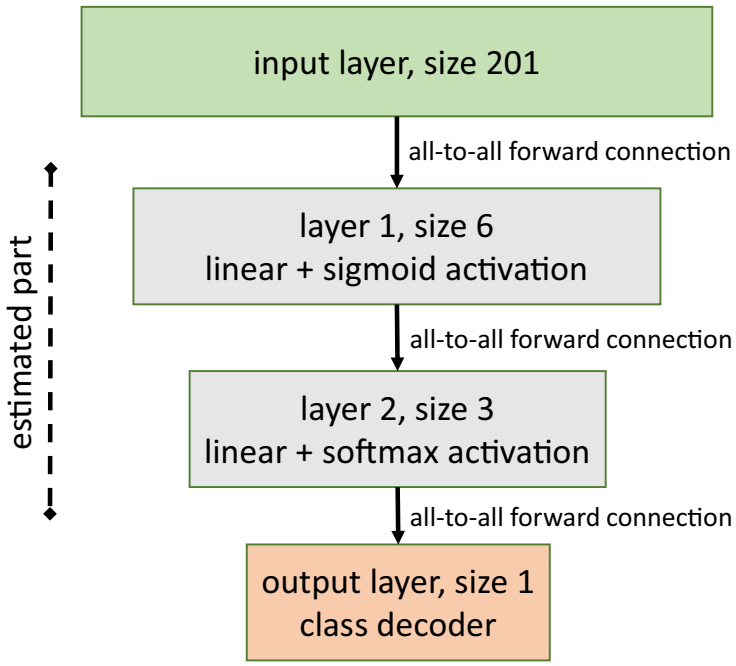


Figure 10.5: The ANN for asteroid spectral classification.

The size of the data set with known labels (i.e., asteroid classes) available for us is 425 cases: 311 S-types, 62 C-types, and 52 X-types. Let us divide the data so that 15 % is left for testing, and also 15 % of the training data is left for validation. Next step is the training of the network, which can take typically some tens of seconds or few minutes, depending on your computer and environment speed. One possible output graph of the trained network is shown in Fig. 10.7 using Mathematica software.

Finally, the trained network can be tested with the test data set. The overall accuracy, i.e., the percentage of the correctly classified cases of all the cases, is about 95 %, which is quite promising. We can look at the results by asteroid types from the confusion matrix plot in Fig. 10.7. We can see that all the S-type asteroids are correctly classified. From C-types, only 1 out of 11 is wrongly classified, but the

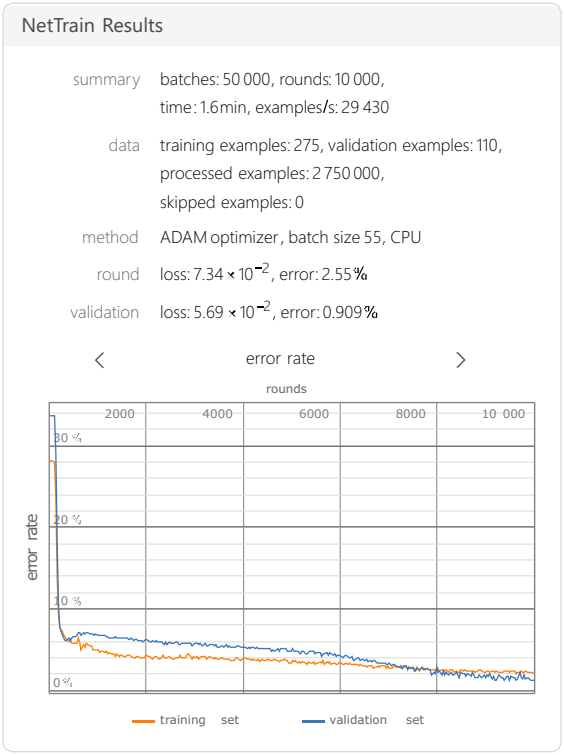


Figure 10.6: Mathematica output summarizing the training process of the network.

X-types are more challenging. Only 50 % of X-types are correctly classified. This is understandable, since X-type here is a heterogeneous collection of types not really fitting into S or C. All in all, the ANN seems to perform very well.

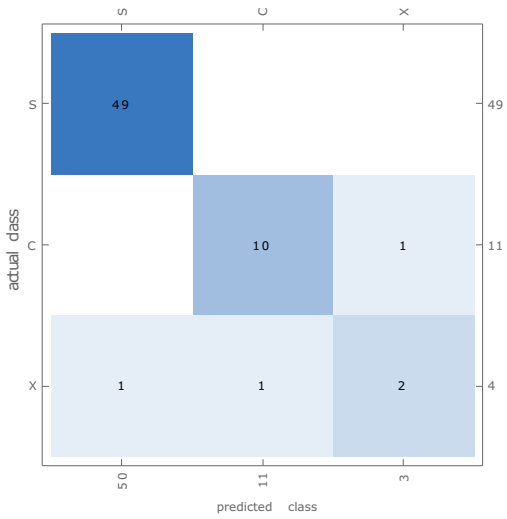


Figure 10.7: The confusion matrix plot of the asteroid classification ANN result.

Chapter 11

Appendix

11.1 Normal and related distributions

Pdf's, cdf's and inverse cdf's for normal, t, χ^2, and $\mathcal{F}$-distributions, formulated using special functions.

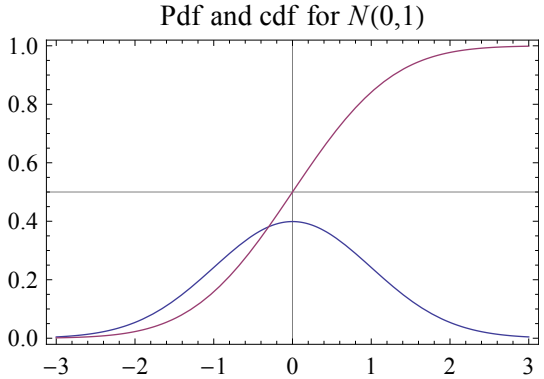
Standard normal distribution

$$\mathrm{f}(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) \tag{11.1}$$

$$\mathrm{F}(y) = \int_{-\infty}^{y} \mathrm{f}(x)dx = \frac{1}{2}\left(1 - \mathrm{erfc}\left(-\frac{y}{\sqrt{2}}\right)\right) \tag{11.2}$$

$$\mathrm{F}^{-1}(p) = \{y : \mathrm{F}(y) = p\} = -\sqrt{2}\,\mathrm{erfc}^{-1}(2p) \tag{11.3}$$

where erfc is the complementary error function, and erfc^{-1} its inverse function.



Standard normal distribution, pdf and cdf.

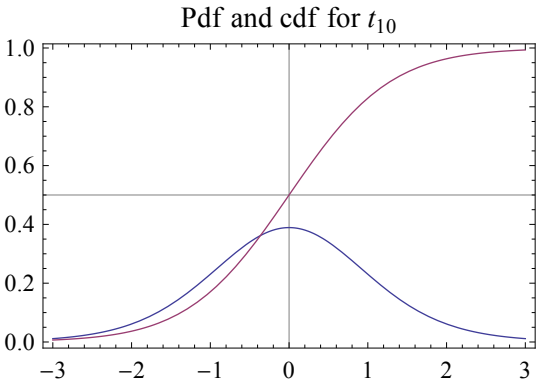
t-distribution

$$\mathrm{f}(y) = \frac{1}{\sqrt{\kappa}\,\mathrm{B}(\kappa/2, 1/2)} \left(\frac{\kappa}{\kappa + y^2}\right)^{\frac{\kappa+1}{2}} \tag{11.4}$$

$$\mathrm{F}(y) = \int_{-\infty}^{y} \mathrm{f}(x)dx = \frac{1}{2}\mathrm{I}\left(\frac{\kappa}{y^2+\kappa}, \frac{\kappa}{2}, \frac{1}{2}\right), \text{ if } y \leq 0, \text{ and} \tag{11.5}$$

$$\frac{1}{2}\left(1 + \mathrm{I}\left(\frac{y^2}{y^2+\kappa}, \frac{1}{2}, \frac{\kappa}{2}\right)\right), \text{ if } y > 0$$

where κ is the degrees of freedom for the distribution, B is the Euler beta function, and $\mathrm{I}(z, a, b)$ is the regularized incomplete beta function.



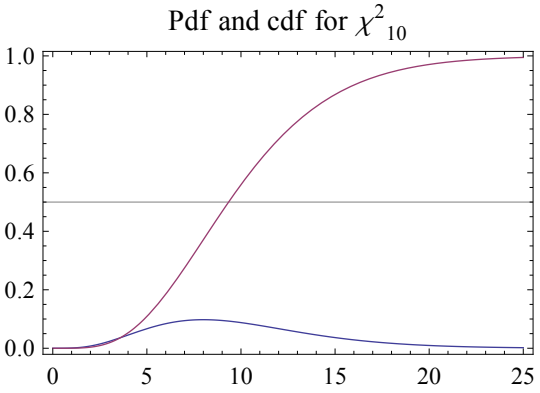
Student's t-distribution with 10 degrees of freedom, pdf and cdf.

χ^2-distribution

$$\mathrm{f}(y) = \frac{2^{-\kappa/2}\exp(-y/2)y^{\frac{\kappa}{2}-1}}{\Gamma\left(\frac{\kappa}{2}\right)} \tag{11.6}$$

$$\mathrm{F}(y) = \int_{-\infty}^{y} \mathrm{f}(x)dx = \mathrm{Q}\left(\frac{\kappa}{2}, 0, \frac{y}{2}\right) \tag{11.7}$$

where κ is the degrees of freedom for the distribution, Γ is the Euler gamma function, and $\mathrm{Q}(a, z_0, z_1)$ is the generalized regularized incomplete gamma function.



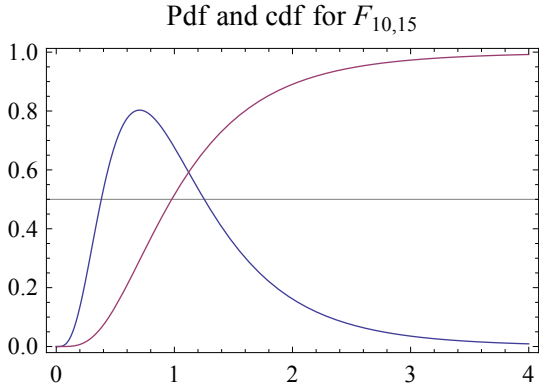
χ^2-distribution with 10 degrees of freedom, pdf and cdf.

$\mathcal{F}$-distribution

$$\mathrm{f}(y) = \frac{\kappa_1^{\kappa_1/2}\kappa_2^{\kappa_2/2}y^{\frac{\kappa_1}{2}-1}(\kappa_2+\kappa_1 y)^{\frac{1}{2}(-\kappa_1-\kappa_2)}}{\mathrm{B}\left(\frac{\kappa_1}{2},\frac{\kappa_2}{2}\right)} \tag{11.8}$$

$$\mathrm{F}(y) = \int_{-\infty}^{y}\mathrm{f}(x)dx = \mathrm{I}\left(\frac{y\kappa_1}{y\kappa_1+\kappa_2},\frac{\kappa_1}{2},\frac{\kappa_2}{2}\right) \tag{11.9}$$

where κ_1 and κ_2 are the degrees of freedom for the distribution, B is the Euler beta function, and $\mathrm{I}(z,a,b)$ is the regularized incomplete beta function.



$\mathcal{F}$-distribution with $\kappa_1 = 10, \kappa_2 = 15$, pdf and cdf.

11.2 Matrix algebra

In what follows we introduce some simple properties of matrix algebra that should be useful with the material in this course. First, some rules regarding matrix transpose:

$$(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T+\mathbf{B}^T \qquad (\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T \qquad (\mathbf{A}^T)^T = \mathbf{A} \tag{11.10}$$

$$(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1} \qquad \det(\mathbf{A}^T) = \det(\mathbf{A}) \tag{11.11}$$

$$\text{If } \mathbf{A} \text{ symmetric, then } \mathbf{A}^T = \mathbf{A} \tag{11.12}$$

$$\text{If } \mathbf{A} \text{ orthogonal, then } \mathbf{A}^T = \mathbf{A}^{-1} \text{ and } \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{11.13}$$

and matrix inverse:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \qquad (\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \qquad \det(\mathbf{A}^{-1}) = \det(\mathbf{A})^{-1} \tag{11.14}$$

$$\text{If } \det(A) = 0, \text{ then } \mathbf{A} \text{ is singular and cannot be inverted} \tag{11.15}$$

$$\text{If } \mathbf{A} \text{ is invertible, then columns of } \mathbf{A} \text{ are linearly independent} \tag{11.16}$$

$$\text{If } \mathbf{A} \text{ is invertible, then } \mathbf{A}^T \text{ is invertible} \tag{11.17}$$

If matrix $\mathbf{A}$ is diagonal, all the entries outside the diagonal $[\mathbf{A}]_{ii}$ are zero. Diagonal matrix can be noted by listing its diagonal elements, $\mathbf{A} = \lceil a_{11}\ a_{22}\ \cdots\ a_{nn} \rfloor$. For diagonal matrices inverse and determinant are easy to calculate:

$$\mathbf{A}^{-1} = \left\lceil \frac{1}{a_{11}}\ \frac{1}{a_{22}}\ \cdots\ \frac{1}{a_{nn}} \right\rfloor \tag{11.18}$$

$$\det(\mathbf{A}) = \prod_i a_{ii} \tag{11.19}$$

Basic rules regarding expectation and covariance operators with matrices:

$$\mathrm{E}(\mathbf{A}Y) = \mathbf{A}\,\mathrm{E}(Y) \qquad \mathrm{cov}(\mathbf{A}Y) = \mathbf{A}\,\mathrm{cov}(Y)\,\mathbf{A}^T \tag{11.20}$$