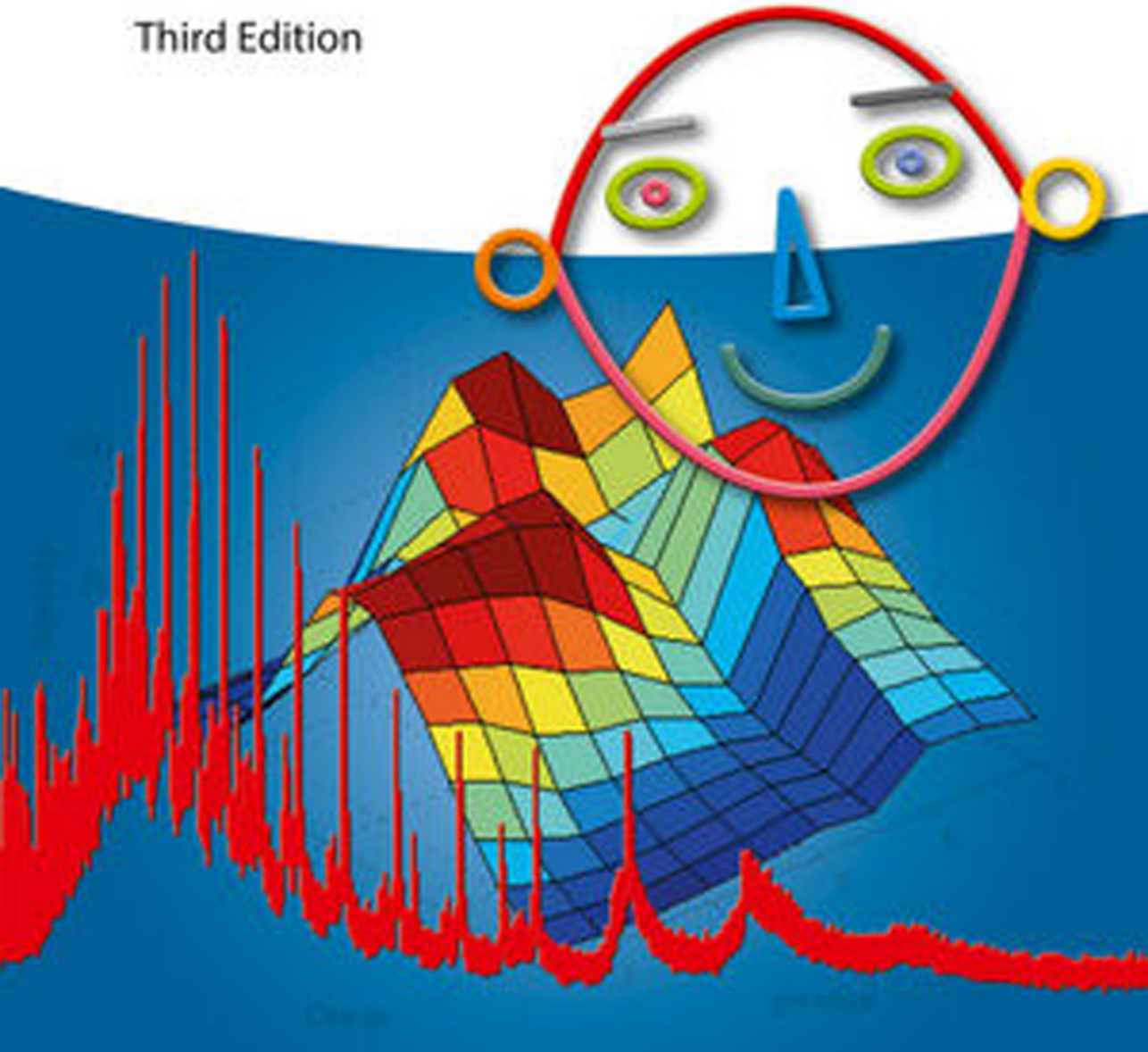
WILEY-VCH

Matthias Otto

Chemometrics

Statistics and Computer Application in Analytical Chemistry

Third Edition



Matthias Otto

Chemometrics

Matthias Otto

Chemometrics

Statistics and Computer Application in Analytical Chemistry

Third Edition

WILEY-VCH
Verlag GmbH & Co. KGaA

Author

Matthias Otto
TU Bergakademie Freiberg
Inst. für Analytische Chemie
Leipziger Str. 29
09599 Freiberg
Germany

Library of Congress Card No.: applied for

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at <http://dnb.d-nb.de>.

Print ISBN: 978-3-527-34097-2
ePDF ISBN: 978-3-527-69936-0
ePub ISBN: 978-3-527-69938-4
Mobi ISBN: 978-3-527-69939-1
oBook ISBN: 978-3-527-69937-7

Cover Design Schulz Grafik-Design, Fußgönheim, Germany
Typesetting SPi Global, Chennai, India

Printing and Binding
Printed on acid-free paper

Contents

List of Abbreviations

ACE	alternating conditional expectations
ADC	analog-to-digital converter
ANFIS	adaptive neuro-fuzzy inference system
ANOVA	analysis of variance
CART	classification and regression trees
CAS	chemical abstracts service
CPU	central processing unit
CRF	chromatographic response function
DAC	digital-to-analog converter
DTW	dynamic time warping
DWT	discrete wavelet transformation
EPA	environmental protection agency
ESI	electrospray ionization
FA	factor analysis
FFT	fast Fourier transformation
FHT	fast Hadamard transformation
FT	Fourier transformation
GC	gas chromatography
HORD	hierarchically ordered ring description
HOSE	hierarchically ordered spherical description of environment
HPLC	high performance liquid chromatography
HT	Hadamard transformation
I/O	input/output
ICA	independent component analysis
IND	indicator function
IR	infrared
ISO	international standardization organization
JCAMP	Joint Committee on Atomic and Molecular Data
KNN	*k*-nearest neighbor method
LAN	local area network
LDA	linear discriminant analysis
LIMS	laboratory information management system

LISP	list processing language
LLM	linear learning machine
MANOVA	multidimensional ANOVA
MARS	multivariate adaptive regression splines
MCR	multivariate curve resolution
MS	mass spectrometry
MSDC	mass spectrometry data center
MSS	mean sum of squares
NIPALS	nonlinear iterative partial least squares
NIR	near infrared
NIST	National Institute of Standards and Technology
NLR	nonlinear regression
NMR	nuclear magnetic resonance
NPLS	nonlinear partial least squares
N-PLS	N-way partial least squares
OLS	ordinary least squares
OPLS	orthogonal partial least squares
PARAFAC	parallel factor analysis
PCA	principal component analysis
PCR	principal component regression
PLS	partial least squares
PRESS	predictive residual sum of squares
PROLOG	programming in logic
RAM	random access memory
RDA	regularized discriminant analysis
RE	real error
ROM	read-only memory
RR	recovery rate
RSD	relative standard deviation
RSM	response surface method
SA	simulated annealing
SEC	standard error of calibration
SEP	standard error of prediction
SIMCA	soft independent modeling of class analogies
SMILES	simplified molecular input line entry specification
SS	sum of squares
SVD	singular value decomposition
SVM	support vector machines
TS	Tabu search
TTFA	target transformation factor analysis
UV	ultraviolet
VIS	visual
VM	vector machines
WT	wavelet transformation

Symbols

α	Significance level (risk), separation factor, complexity parameter
A	Area
b	Breadth (width), regression parameter
χ^2	Quantile of chi-squared distribution
cov	Covariance
$\boldsymbol{C}$	Variance covariance matrix
δ	Error, chemical shift
d	Distance measure, test statistic of by Kolmogorov–Smirnov
D	Difference test statistic
D_k	Cook's test statistic
η	Learning coefficient
e	Error
$E(.)$	Expectation value
$\boldsymbol{E}$	Residual matrix
f	Degree of freedom, function
$f(x)$	Probability density function
F	Quantile of Fisher distribution
$F(\cdot)$	Function in frequency space
$f(t)$	Function in the time domain
$\boldsymbol{F}$	Matrix of scores
G	Geometric mean
$G(\cdot)$	Smoothing function in frequency space
$g(t)$	Smoothing function in the time domain
h	Communality
H	Harmonic mean
$\boldsymbol{H}$	Hat matrix, Hadamard transformation matrix
H_0	Null hypothesis
H_1	Alternative hypothesis
$H(\cdot)$	Filter function in frequency space
$h(t)$	Filter function in the time domain
I	Unit step function

$\boldsymbol{I}$	Identity matrix
I_{50}	Interquartile range
$\boldsymbol{J}$	Jacobian matrix
k	Kurtosis
k	Capacity factor
K_A	Protolysis (acid) constant
λ	Eigenvalue, Poisson parameter
$\boldsymbol{L}$	Loading matrix
μ	Population mean
m	Membership function
m_r	Moment of distribution
nf	Neighborhood function
N	Analytical Resolution, plate number
$N(v)$	Noise
p	portion
P	Probability
Q	Quartile, Dixon statistics
r	Correlation coefficient, radius
$\boldsymbol{R}$	Correlation matrix
R_S	Chromatographic resolution
R^2	Coefficient of determination
σ	Standard deviation
s	Estimation of standard deviation, skewness
s_r	Estimation of relative standard deviation
S	Similarity measure
τ	Time lag
t	Quantile of Student distribution
T	Test quantity of Grubbs' test
$\boldsymbol{T}$	Matrix of principal components scores, transformation matrix
$\boldsymbol{U}$	Matrix of left eigenvectors
R	Range, region in CART
$\boldsymbol{V}$	Matrix of right eigenvectors
w	Singular value, weight
W	Warping path
x	(Independent) variable
$\boldsymbol{x}$	Vector of independent variables
$\boldsymbol{X}$	Matrix of dependent variables
$\overline{x}$	Arithmetic mean
ξ	Slack variable
y	(Dependent) variable
y^*	Transformed (filtered) value
z	Standard normal deviate, signal position, objective function

1
What is Chemometrics?

Learning objectives

- To define chemometrics
- To learn how to count with bits and how to perform arithmetic or logical operations in a computer
- To understand the principal terminology for computer systems and the meaning of robotics and automation.

The development of the discipline chemometrics is strongly related to the use of computers in chemistry. Some analytical groups in the 1970s were already working with statistical and mathematical methods that are ascribed nowadays to chemometric methods. Those early investigations were connected to the use of mainframe computers.

The notation *chemometrics* was introduced in 1972 by the Swede Svante Wold and the American Bruce R. Kowalski. The foundation of the International Chemometrics Society in 1974 led to the first description of this discipline. In the following years, several conference series were organized, for example, Computer Application in Analytics (COMPANA), Computer-Based Analytical Chemistry (COBAC), and Chemometrics in Analytical Chemistry (CAC). Some journals devoted special sections to papers on chemometrics. Later, novel chemometric journals were started, such as the *Journal of Chemometrics* (Wiley) and *Chemometrics and Intelligent Laboratory Systems* (Elsevier).

An actual definition of chemometrics is:

> *the chemical discipline that uses mathematical and statistical methods, (a) to design or select optimal measurement procedures and experiments, and (b) to provide maximum chemical information by analyzing chemical data.*

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

The discipline of chemometrics originates in chemistry. Typical applications of chemometric methods are the development of quantitative structure activity relationships and the evaluation of analytical–chemical data. The data flood generated by modern analytical instrumentation is one reason that analytical chemists in particular develop applications of chemometric methods. Chemometric methods in *analytics* is the discipline that uses mathematical and statistical methods to obtain relevant information on material systems.

With the availability of personal computers at the beginning of the 1980s, a new age commenced for the acquisition, processing, and interpretation of chemical data. In fact, today, every scientist uses software, in one form or another, that is related to mathematical methods or to processing of knowledge. As a consequence, the necessity emerges for a deeper understanding of those methods.

The education of chemists in mathematics and statistics is usually unsatisfactory. Therefore, one of the initial aims of chemometrics was to make complicated mathematical methods practicable. Meanwhile, the commercialized statistical and numerical software simplifies this process, so that all important chemometric methods can be taught in appropriate computer demonstrations.

Apart from the statistical–mathematical methods, the topics of chemometrics are also related to problems of the computer-based laboratory, to methods for handling chemical or spectroscopic databases, and to methods of artificial intelligence.

In addition, chemometricians contribute to the development of all these methods. As a rule, these developments are dedicated to particular practical requirements, such as the automatic optimization of chromatographic separations or in prediction of the biological activity of a chemical compound.

1.1 The Computer-Based Laboratory

Nowadays, the computer is an indispensable tool in research and development. The computer is linked to analytical instrumentation; it serves as a tool for acquiring data, word processing, and handling databases and quality assurance systems. In addition, the computer is the basis for modern communication techniques such as electronic mails or video conferences. In order to understand the important principles of computer usage, some fundamentals

are considered here, that is, coding and processing of digital information, the main components of the computer, programming languages, computer networking, and automation processes.

Analog and Digital Data

The use of digital data provides several advantages over the use of analog data. Digital data are less noise sensitive. The only noise arises from round-off errors due to finite representation of the digits of a number. They are less prone to, for instance, electrical interferences, and they are compatible with digital computers.

As a rule, primary data are generated as analog signals either in a discrete or a continuous mode (Figure 1.1). For example, monitoring the intensity of optical radiation by means of a photocell provides a continuous signal. Weak radiation, however, could be monitored by detecting individual photons by a photomultiplier.

Usually, the analog signals generated are converted into digital data by an analog-to-digital converter (ADC) as explained as follows.

Binary versus Decimal Number System

In a digital measurement, the number of pulses occurring within a specified set of boundary conditions is counted. The easiest way to count is to have the pulses represented as binary numbers. In this way, only two electronic states are required. To represent the decimal numbers from 0 to 9, one would need 10 different states. Typically, the binary numbers 0 and 1 are represented electronically by voltage signals of 0.5 and 5 V, respectively. Binary numbers characterize coefficients of the power of 2, so that any number of the decimal system can be described.

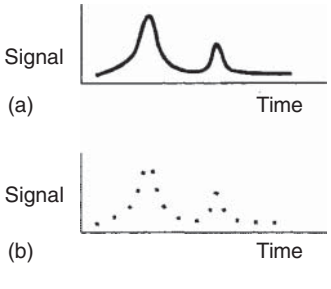


Figure 1.1 Signal dependence on time of an analog (a) and a digital detector (b).

Example 1.1 *Binary Number Representation*

The decimal number 77 is expressed as binary number by 1001101, that is,

1	0	0	1	1	0	1
1×2^6	0×2^5	0×2^4	1×2^3	1×2^2	0×2^1	$1\times2^0=$
64	+0	+0	+8	+4	+0	+1 = 77

Table 1.1 Relationship between binary and decimal numbers.

Binary number	Decimal number
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1101	13
10000	16
100000	32
1000000	64

Table 1.1 summarizes further relationships between binary and decimal numbers. Every binary number is composed of individual *bits* (binary digits). The digit lying farthest to the right is termed the *least significant* digit and the one on the left is the *most significant* digit.

How are calculations done using binary numbers? Arithmetic operations are similar but simpler than those for decimal numbers. In addition, for example, four combinations are feasible:

0	0	1	1
+0	+1	+0	+1
0	1	1	10

Note that for addition of the binary numbers 1 plus 1, a 1 is carried over to the next higher power of 2.

Example 1.2 *Calculation with Binary Numbers*

Consider addition of 21 + 5 in the case of a decimal (a) and of a binary number (b):

a.	21	b.	10101
	+5		101
	26		11010

Apart from arithmetic operations in the computer, logical reasoning is necessary too. This might be in the course of an algorithm or in connection with an expert system. Logical operations with binary numbers are summarized in Table 1.2.

It should be mentioned that a very compact representation of numbers is based on the *hexadecimal number system.* However,

Table 1.2 Truth values for logical connectives of predicates *p* and *q* based on binary numbers.

p	*q*	*p* AND *q*	*p* OR *q*	IF *p* THEN *q*	NOT *p*
1	1	1	1	1	0
1	0	0	1	0	—
0	1	0	1	1	1
0	0	0	0	1	—

1 True and 0 false.

hexadecimal numbers are easily converted into binary data, so the details need not be explored here.

Digital and Analog Converters

Analog-to-Digital Converters (ADCs)

In order to benefit from the advantages of digital data evaluation, the analog signals are converted into digital ones. An analog signal consists of an infinitely dense sequence of signal values in a theoretically infinitely small resolution. The conversion of analog into digital signals in the ADC results in a definite reduction of information. For conversion, signal values are sampled in a predefined time interval and quantified in an *n*-ary raster (Figure 1.2). The output signal is a code word consisting of *n* bits. Using *n* bits, 2^n different levels can be coded, for example, an 8-bit ADC has a resolution of $2^8 = 256$ amplitude levels..

Digital-to-Analog Converters (DACs)

Converting digital into analog information is necessary if an external device is to be controlled or if the data have to be represented by an analog output unit. The resolution of the analog signal is determined by the number of processed bits in the converter. A 10-bit DAC provides $2^{10} = 1024$ different voltage increments. Its resolution is then 1/1024 or approximately 0.1%.

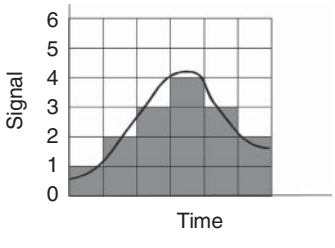


Figure 1.2 Digitization of an analog signal by an analog-to-digital converter (ADC).

Computer Terminology

Representation of numbers in a computer by bits has already been considered. The combination of 8 bits is called a *byte*. A series of bytes arranged in sequence to represent a piece of data is termed a *word*. Typical word sizes are 8, 16, 32, or 64 bits or 1, 2, 4, and 8 bytes.

Words are processed in *registers.* A sequence of operations in a register enables *algorithms* to be performed. One or several algorithms make up a *computer program.*

The physical components of a computer form the *hardware.* Hardware includes the disk and hard drives, clocks, memory units, and registers for arithmetic and logical operations. Programs and instructions for the computer, including the tapes and disks for their storage, represent the *software.*

Components of Computers

Central Processing Units and Buses

A bus consists of a set of parallel conductors that forms a main transition path in a computer.

The heart of a computer is the central processing unit (CPU). In a microprocessor or minicomputer, this unit consists of a highly integrated chip.

The different components of a computer, its memory, and the peripheral devices, such as printers or scanners, are joined by buses. To guarantee rapid communication among the various parts of a computer, information is exchanged on the basis of a definitive word size, for example, 16 bits, simultaneously over parallel lines of the bus. A data bus serves the exchange of data into and out of the CPU. The origin and the destination of the data in the bus are specified by the address bus. For example, an address bus with 16 lines can address $2^{16} = 65536$ different registers or other locations in the computer or in its memory. Control and status information to and from the CPU are administrated in the control bus. The peripheral devices are controlled by an external bus system, for example, an RS-232 interface for serial data transfer or the IEEE-488 interface for parallel transfer of data.

Memory

The microcomputer or microprocessor contains typically two kinds of memory: *random access memory* (RAM) and *read-only memory* (ROM). The term RAM is somewhat misleading and historically reasoned, since random access is feasible for RAM and ROM alike. The RAM can be used to read and write

information. In contrast, information in a ROM is written once, so that it can be read, but not reprogrammed. ROMs are needed in microcomputers or pocket calculators in order to perform fixed programs, for example, for calculation of logarithms or standard deviations.

Larger programs and data collections are stored in *bulk storage devices*. In the beginning of the computer age, magnetic tapes were the standard here. Nowadays CD's, DVD' , and Blu-Ray's are used providing a storage capacity of several gigabytes. In addition, every computer is equipped with a hard disk of at least several gigabytes. The access time to retrieve the stored information is in the order of a few milliseconds.

Input/Output Systems

Communication with the computer is carried out by input–output (I/O) operations. Typical input devices are keyboard, magnetic tapes and disks, and the signals of an analytical instrument. Output devices are screens, printers, and plotters, as well as tapes and disks. To convert analog information into digital or vice versa, the aforementioned ADCs or DACs are used.

Programs

Programming a computer at 0 and 1 states or bits is possible using *machine code.* Since this kind of programming is rather time consuming, higher level languages have been developed where whole groups of bit operations are assembled. However, these so-called *assembler languages* are still difficult to handle. Therefore, high-level algorithmic languages, such as FORTRAN, BASIC, PASCAL, or C, are more common in analytical chemistry. With high-level languages, the instructions for performing an algorithm can easily be formulated in a computer program. Thereafter, these instructions are translated into machine code by means of a *compiler.*

For logical programming, additional high-level languages exist, for example, LISP (List Processing language) or PROLOG (Programming in Logic). Further developments are found in the so-called *Shells,* which can be used directly for building expert systems.

Networking

A very effective communication between computers, analytical instruments, and databases is based on networks. There are local nets, for example, within an industrial laboratory as well as

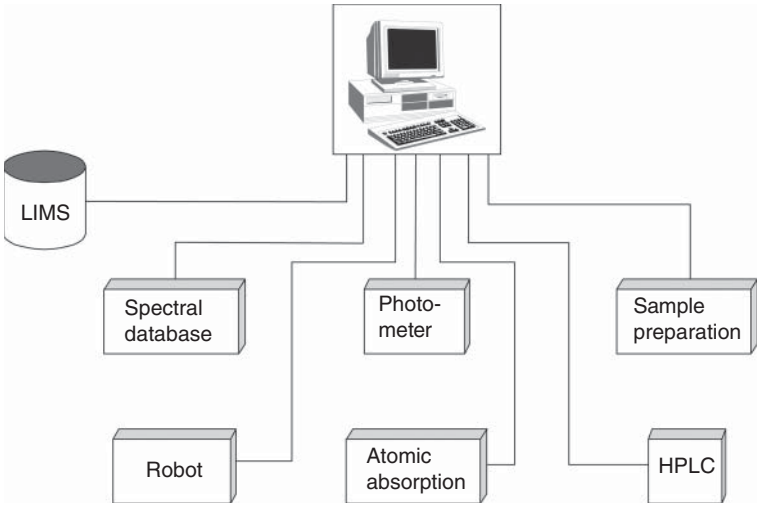


Figure 1.3 Local area network (LAN) to connect analytical instruments, a robot, and a laboratory information management system (LIMS).

national or worldwide networks. Local area networks (LANs) are used to transfer information about analysis samples, measurements, research projects, or in-house databases. A typical LAN is demonstrated in Figure 1.3. It contains a laboratory information management system (LIMS), where all information about the sample or the progresses in a project can be stored and further processed (cf. Section 7.1).

Worldwide networking is feasible, for example, via Internet or CompuServe. These nets are used to exchange electronic mails (e-mail) or data with universities, research institutions, or industry.

Robotics and Automation

Apart from acquiring and processing analytical data, the computer can also be used to control or supervise automatic procedures. To automate manual procedures, a *robot* is applied. A robot is a reprogrammable device that can perform a task more cheaply and effectively than a human being.

Typical geometric shapes of a robot arm are sketched in Figure 1.4. The anthropomorphic geometry (Figure 1.4a) is derived from the human torso, that is, there is a waist, a shoulder, an elbow, and a wrist. Although this type of robot is mainly found in the automobile industry, it can also be used for manipulation of liquid or solid samples.

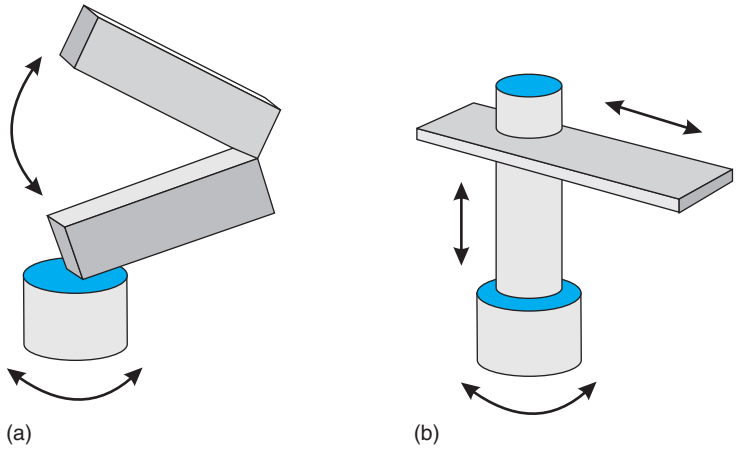


Figure 1.4 Anthropomorphic (a) and cylindrical (b) geometry of robot arms.

In the chemical laboratory, the cylindrical geometry dominates (Figure 1.4b). The revolving robot arm can be moved in horizontal and vertical directions. Typical operations of a robot are as follows:

- *Manipulation* of test tubes or glassware around the robotic work area
- *Weighing,* for the determination of a sample amount or for checking unit operations, for example, addition of solvent
- *Liquid handling,* in order to dilute or add reagent solutions
- *Conditioning* of a sample by heating or cooling
- *Separations* based on filtrations or extractions
- *Measurements* by analytical procedures, such as spectrophotometry or chromatography
- *Control and supervision* of the different analytical steps.

Programming of a robot is based on software dedicated to the actual manufacture. The software consists of elements to control the peripheral devices (robot arm, balance, pumps), to switch the devices on and off, and to provide instructions on the basis of logical structures, for example, IF – THEN rules.

Alternatives for automation in a laboratory are *discrete analyzers* and *flowing systems.* By means of discrete analyzers, unit operations such as dilution, extraction and dialyses can be automated. Continuous flow analyzers or flow injection analyses serve similar objectives for automation, for example, for the determination of clinical parameters in blood serum.

The transfer of manual operations to a robot or an automated system provides the following advantages:

- High productivity and/or minimization of costs
- Improved precision and trueness of results
- Increased assurance for performing laboratory operations
- Easier validation of the different steps of an analytical procedure.

The increasing degree of automation in the laboratory leads to more and more measurements that are available online in the computer and have to be further processed by chemometric data evaluation methods.

1.2 Statistics and Data Interpretation

Table 1.3 provides an overview of chemometric methods. The main emphasis is on statistical–mathematical methods. Random data are characterized and tested by the descriptive and inference methods of statistics, respectively. Their importance increases in connection with the aims of quality control and quality assurance. Signal processing is carried out by means of algorithms for smoothing, filtering, derivation, and integration. Transformation methods such as the Fourier or Hadamard transformations also belong in this area.

Efficient experimentation is based on the methods of experimental design and its quantitative evaluation. The latter can be performed by means of mathematical models or graphical representations. Alternatively, sequential methods are applied, such as the simplex method, instead of these simultaneous methods of experimental optimization. There, the optimum conditions are found by systematic search for the objective criterion, for example, the maximum yield of a chemical reaction, in the space of all experimental variables.

Table 1.3 Chemometric methods for data evaluation and interpretation.

Descriptive and inference statistics
Signal processing
Experimental design
Modeling
Optimization
Pattern recognition
Classification
Artificial intelligence methods
Image processing
Information and system theory

To find patterns in data and to assign samples, materials, or, in general, objects, to those patterns, multivariate methods of data analysis are applied. Recognition of patterns, classes, or clusters is feasible with projection methods, such as principal component analysis or factor analysis, or with cluster analysis. To construct class models for classification of unknown objects, we will introduce discriminant analyses.

To characterize the information content of analytical procedures, information theory is used in chemometrics.

1.3 Computer-Based Information Systems/Artificial Intelligence

A further subject of chemometrics is the computer-based processing of chemical structures and spectra.

There, it might be necessary to extract a complete or partial structure from a collection of molecular structures or to compare an unknown spectrum with the spectra of a spectral library.

For both kinds of queries, methods for representation and manipulation of structures and spectra in databases are needed. In addition, problems of data exchange formats, for example, between a measured spectrum and a spectrum of a database, are to be decided.

If no comparable spectrum is found in a spectral library, then methods for spectra interpretation become necessary. For interpretation of atomic and molecular spectra, in principle, all the statistical methods for pattern recognition are appropriate (cf. Section 1.2). In addition, *methods of artificial intelligence* are used. They include methods of logical reasoning and tools for developing expert systems. Apart from the methods of classical logic in this context, methods of approximate reasoning and of *fuzzy logic* can also be exploited. These interpretation systems constitute methods of *knowledge processing* in contrast to data processing based on mathematical–statistical methods.

Knowledge acquisition is mainly based on expert knowledge, for example, the infrared spectroscopist is asked to contribute his knowledge in the development of an interpretation system for infrared spectra. Additionally, methods are required for automatic knowledge acquisition in the form of *machine learning*.

Methods based on fuzzy theory, neural nets, and evolutionary strategies are denoted as *soft computing*.

The methods of artificial intelligence and machine learning are not restricted to the interpretation of spectra. They can also be used to develop expert systems, for example, for the analysis of drugs or the synthesis of an organic compound.

Novel methods are based on biological analogs, such as neural networks and evolutionary strategies, for example, genetic algorithms. Future areas of research for chemometricians will include the investigation of *fractal structures* in chemistry and of models based on the theory of *chaos*.

General Reading

1. Sharaf, M.A., Illman, D.L., and Kowalski, B.R. (1986) *Chemometrics*, Chemical Analysis Series, vol. **82**, John Wiley & Sons, Inc., New York.
2. Massart, D.L., Vandeginste, B.G.M., Deming, S.N., Mi-chotte, Y., and Kaufmann, L. (1988) *Chemometrics – a Textbook*, Elsevier, Amsterdam.
3. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, 4 Volumes, Elsevier, Amsterdam.
4. Varmuza, K. and Filzmoser, P. (2009) *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, Boca Raton, FL, Berlin.

Questions and Problems

1. Calculate the resolution for 10-, 16-, and 20-bit analog-to-digital converters.
2. How many bits are stored in an 8-byte word?
3. What is the difference between procedural and logical programming languages?
4. Discuss typical operations of an analytical robot.

2
Basic Statistics

Learning objectives

- To introduce the fundamentals of descriptive and inference statistics
- To highlight important distributions such as normal, Poisson, Student's t, F, and χ^2
- To understand the measures for characterizing location and dispersion of a data set
- To discuss the Gaussian error propagation law
- To learn statistical tests for comparison of data sets and for testing distributions or outliers
- To distinguish between one- and two-sided statistical tests at the lower and upper end of a distribution
- To estimate the effect of experimental factors on the basis of univariate and multivariate analyses of variance.

In analytical chemistry, statistics are needed to evaluate analytical data and measurements and to preprocess, reduce, and interpret the data.

As a rule, analytical data are to some degree *uncertain.* There are three sources of uncertainty:

- Variability
- Measurement uncertainty
- Vagueness.

A high degree of *variability* of data is typically observed with data from living beings, reflecting the rich variability of nature. For example, the investigation of tissue samples provides a very variable pattern of individual compounds for each human individual.

Measurement uncertainty is connected with the impossibility of observing or measuring to an arbitrary level of precision and

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

without systematic errors (bias). This is the type of uncertainty the analyst has to consider most frequently.

Vagueness is introduced by using a natural or professional language to describe an observation, for example, if the characterization of a property is uncertain. Typical vague descriptions represent sensory variables, such as sweet taste, raspberry-colored appearance, or aromatic smell.

For description of the uncertainty due to variability and measurement uncertainty, statistical methods are used. Vague circumstances are characterized by fuzzy methods (cf. Section 8.3).

2.1 Descriptive Statistics

Sources of uncertainty in analytical measurements are *random* and *systematic* errors. Random errors are determined by the limited precision of measurements. They can be diminished by repetitive measurements. To characterize random errors, probability-based approaches are used where the measurements are considered as random, independent events.

Apart from *descriptive* statistics, there exists *inference* statistics (cf. Section 2.2).

Systematic errors (bias) represent a constant or multiplicative part of the experimental error. This error cannot be decreased by repetitive measurements. In analytics, the trueness of values, that is, the deviation of the mean from the true value, is related to a systematic error. Appropriate measurements with standards are used to enable recognition of systematic errors in order to correct them at later measurements.

Measurements that are dependent on each other provide correlated data. Typically, time-dependent processes, such as the time series of glucose concentrations in blood, are of this type of data. Correlated measurements cannot be characterized by the same methods used for description of random independent observations. They require methods of time series analysis where it is assumed that the measurements are realizations of a stochastic process and where they are statistically dependent (Section 3.2).

Distribution of Random Numbers

In the following, we consider random and uncorrelated data. The distribution of random data can be determined from their frequency in a predefined interval, also called *class*. As an example, we consider (Table 2.1) repetitive measurements of a sample solution in spectrophotometry. Partitioning the continuous variable into 12 classes, the frequency of the observations in each class is

To decide on the *number of classes*, their class width, w, is used. In dependence on the number n of single values and the range R_n (Eq. (2.18)), it is valid:

$$w = \frac{R_n}{\sqrt{n}} \text{ for } 30 < n \leq 400$$

$$w = \frac{R_n}{20} \text{ for } n > 400$$

Table 2.1 Spectrophotometric measurements (absorbances) of a sample solution from 15 repetitive measurements.

Measurement	Value	Measurement	Value
1	0.3410	9	0.3430
2	0.3350	10	0.3420
3	0.3470	11	0.3560
4	0.3590	12	0.3500
5	0.3530	13	0.3630
6	0.3460	14	0.3530
7	0.3470	15	0.3480
8	0.3460		

Table 2.2 Frequency distribution of measurements from Table 2.1.

Range	Frequency	Relative frequency (%)
0.3300 – 0.3333	0	0
0.3333 – 0.3367	1	6.67
0.3367 – 0.3400	0	0
0.3400 – 0.3433	3	20.00
0.3433 – 0.3467	2	13.33
0.3467 – 0.3500	4	26.67
0.3500 – 0.3533	2	13.33
0.3533 – 0.3567	1	6.67
0.3567 – 0.3600	1	6.67
0.3600 – 0.3633	1	6.67
0.3633 – 0.3667	0	0
0.3667 – 0.3700	0	0

obtained (Table 2.2). Graphically, the frequency of observations is represented in a histogram (Figure 2.1).

Gaussian Distribution

If the number of repetitive measurements is increased to infinity and the class width is simultaneously decreased, then a bell-shaped distribution for the frequency of the measurements is obtained. It is called a *Gaussian* or *normal distribution* and is illustrated in Figure 2.1 by a solid line.

The terms probability density function, density function, and frequency function are used *synonymously.*

The Gaussian distribution is expressed mathematically by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \; e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.1}$$

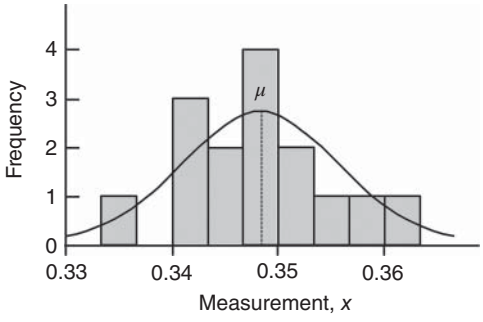


Figure 2.1 Histogram for the measurements of Table 2.1 and the theoretical distribution function according to a Gaussian distribution (*solid line*).

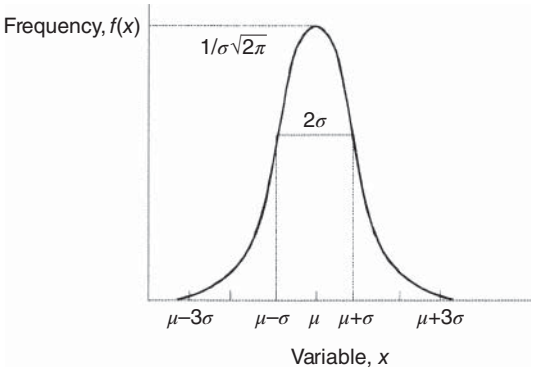


Figure 2.2 Probability density distribution function of the Gaussian distribution according to Eq. (2.1) with the mean μ and the standard deviation σ.

where $f(x)$ is the frequency or probability density function, σ the standard deviation, μ the mean, and x the measurement (variable).

The mean, μ, characterizes the location of the data on the variable axis. The standard deviation, σ, and its square, the variance σ^2, describe the *dispersion* of data around their mean (cf. Figure 2.2). The Greek letters are used by the statistician for true parameters of a population. Since only a limited number of measurements is available, the location and variance parameters must be estimated. The estimates are labeled by Latin letters or by a hat, for example, $\hat{y}$. For estimations of the parameters of a Gaussian distribution, we obtain

Dispersion of data is also termed *variation*, *scatter*, or *spread*.

$$f(x) = \frac{1}{s\sqrt{2\pi}} e^{-\frac{(x-\overline{x})^2}{2s^2}} \tag{2.2}$$

where s is the estimate of the standard deviation and $\overline{x}$ the estimate of the mean.

Estimation of the arithmetic mean is calculated for n repetitive measurements by

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.3}$$

The standard deviation is estimated by

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}} \tag{2.4}$$

Moments of a Distribution

Mean and variance can be derived from the moments of a distribution. In general, the rth central moment calculated about the mean is

$$m_r(x) = \int_{-\infty}^{\infty} (x-\mu)^r f(x)\ \mathrm{d}x \tag{2.5}$$

For the individual moments, one obtains

- First moment: *mean*

$$m_1(x) = \int_{-\infty}^{\infty} xf(x)\ \mathrm{d}x = \mu \tag{2.6}$$

- Second central moment: *variance*

$$m_2(x) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x)\ \mathrm{d}x = \sigma^2 \tag{2.7}$$

- Third moment: *skewness s* (asymmetry of distribution)

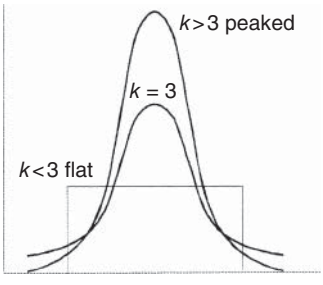
$$s = \frac{m_3(x)}{\sigma^3} \tag{2.8}$$

For symmetric distributions, the value of the skewness is zero. Left-peak asymmetry reveals a value larger than zero and a right-peaked shape smaller than zero.

- Fourth moment: *kurtosis k* (measure for excess)

$$k = \frac{m_4(x)}{\sigma^4} \tag{2.9}$$

A flat distribution has values smaller than, and a peaked one values larger than, zero. For the normal distribution $k = 3$, that is, for a peaked normal distribution, $k > 3$. Frequently, the peakedness is defined by k values related to the normal distribution, that is, by the expression $k' = k - 3$.



Other Distributions

Apart from the distributions used for hypothesis testing, that is, F-, t-, and χ^2-distributions, to be considered in Section 2.2, there are further models for the distribution of random numbers. Those are the lognormal, uniform, binomial, and Poisson distributions.

The *Poisson distribution* is quite important for the analyst to characterize countable but rare events. Such events are typical for procedures based on counting rates, for example, if a photomultiplier is applied in optical spectrometry or a proportional counter in X-ray analysis.

A *variate* specifies a random variable.

The Poisson distribution is based on the probability density function for discrete values of a variate. This is termed a *probability function*. For each value of this function, $f(x)$, a probability for the realization of the event, x, can be defined. It is calculated according to a Poisson distribution by

$$f(x) = \frac{\lambda^x \ \mathrm{e}^{-\lambda}}{x!} \tag{2.10}$$

The parameter λ represents the mean and the variance of the distribution similarly, that is, $\lambda = \mu = \sigma^2$.

Figure 2.3 demonstrates the Poisson distribution for λ values of 1 and 5, that is, the distributions for the cases that on average 1 event or 5 events are observed, respectively. The breadth of curves, also determined by λ, is only dependent on the total number of events or the counting rate. The higher the counting rate, the larger will be the variance or standard deviation of the counting process. For $x = n$ events, the variance $\sigma^2 = n$, or for the standard deviation, the Poisson distribution becomes

$$\sigma = \sqrt{n} \tag{2.11}$$

This is true for the absolute standard deviation. In contrast, the relative standard deviation is diminished with increasing counting

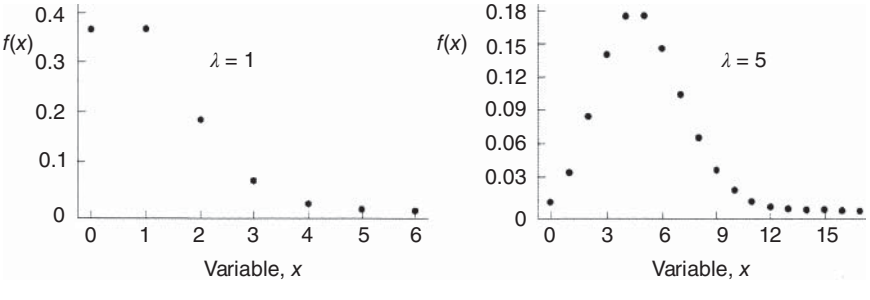


Figure 2.3 Probability function of the Poisson distribution according to Eq. (2.10) for two values of the parameter λ.

rate since (cf. Eq. 2.16)

$$\sigma_r = \frac{\sigma}{n} = \frac{\sqrt{n}}{n} = \frac{1}{\sqrt{n}} \tag{2.12}$$

Central Limit Theorem

The most important distribution is the normal distribution. This conclusion can be drawn from the *central limit theorem.*

The distribution of a sum, y, calculated from $i = 1$, p variables, x_i (Eq. (2.13)), with the means μ_i and the variances σ_i^2 tends to a normal distribution with the mean $\Sigma_i \mu_i$ and the variance $\Sigma_i \sigma_i^2$, if p approaches infinity, independent of the distributions of the individual variables, x_i.

$$y = x_1 + x_2 + \cdots + x_p \tag{2.13}$$

The central limit theorem is illustrated in Figure 2.4. The distribution of the population considered follows a binomial distribution. From this population, several samples are drawn with 2, 4, and 25 samples in each group. Then the means of those

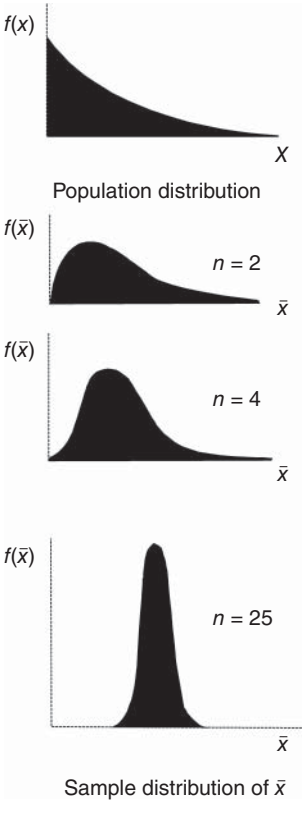


Figure 2.4 Illustration of the central limit theorem for the distribution of means taken from a binomial distribution X.

The probability density function of a *binomial distribution* is

$$f(x) = \binom{n}{x} \; p^x (1-x)^{n-x}$$

with mean np and variance $np(1-p)$. Here n is the number of trials and p is the probability of success.

groups are formed and plotted as a distribution of the means. Although the investigated population is binomially distributed, the distribution of the means leads to a normal distribution.

Location Parameter

A data set can be characterized by the following quantities: *frequency, location, variance, skewness, kurtosis, quantile,* and *rank.*

The only location quantity considered thus far was the arithmetic mean (Eq. (2.3)). For some problems, different location parameters are more appropriate.

Geometric Mean

For lognormally distributed data, often the geometric mean is reported, since it is valid:

$$G = \sqrt[n]{x_1 x_2 \ \ldots \ x_n} \text{ or } \log G = \frac{\sum_{i=1}^{n} \log \ x_i}{n} \tag{2.14}$$

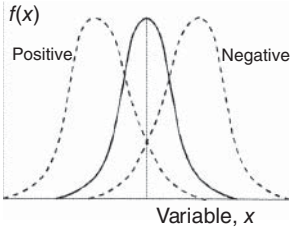
Harmonic Mean

Median and interquartile range represent a rank order statistic. A *rank* indicates the position of an object with respect to other objects by means of an ordinal number.

The harmonic mean is another parameter for characterization of the most central and typical value of a set of data. Its definition reads

$$H = \frac{n}{\sum_{i=1}^{n} \left(\frac{1}{x_i}\right)} \tag{2.15}$$

When the geometric mean exists, it lies between the harmonic and arithmetic means, that is, $H \leq G \leq \overline{x}$.



In the case of symmetric distributions, the mean, $\overline{x}$, the median, and the mode are identical. For *positive* values of the skewness, $\overline{x}$ > median > mode is valid; *negative* skewness values result in $\overline{x}$ < median < mode.

Median

A very robust location measure is the median. For an odd number of values, the median is the middle-order statistic. It lies at the position $(n+1)/2$. For an even number of measurements, the median is calculated from the average of the $(n/2)$th and $(n/2+1)$th order statistics. The median is not dependent on outliers, and this is an advantage compared to the arithmetic mean.

Quartile

Looking from a different perspective, the median represents that point that divides the total frequency into two halves, that is, 50% of the data are found above and below the median, respectively. This point is also termed the *middle quartile* $Q(0.5)$ or Q_2. The quartiles Q_1 and Q_3 are then called the *lower* and *upper quartiles*, respectively. The lower quartile contains 25% of all measurements,

the upper one 75%. A percentile p% divides the data range into hundreds.

Dispersion Measures

As the most common dispersion measure, we have already used the standard deviation. Frequently, this measure is reported as relative standard deviation:

$$s_r = \frac{s}{\overline{x}} \text{ or in percent } s_r(\%) = s_r 100 \tag{2.16}$$

This quantity is also termed the *coefficient of variation*.

The standard error characterizes the averaged error of the mean for n observations:

$$s_{\overline{x}} = \frac{s}{\sqrt{n}} \tag{2.17}$$

The data difference between the maximum, x_{max}, and minimum value, x_{min}, is termed the *range*, R:

$$R = x_{max} - x_{min} \tag{2.18}$$

This quantity describes the range, which contains 100% of all observations. If a range is of interest that contains just 50% of all the observations, then the *interquartile range* should be calculated:

$$I_{50} = Q_3 - Q_1 \tag{2.19}$$

The interquartile range is obtained from the difference between the lower and upper quartiles.

The *mode* is the most frequently occurring value in a set of data. In the case of categorized data, the mode is equivalent to the class with the highest frequency.

A *quantile* divides a set of observations in two groups, such that one fraction falls above and the complementary fraction below, the value specified by the quantile. The most frequently applied quantiles are quartiles and percentiles.

Confidence Interval

The confidence interval characterizes the range about the mean of a random variable, in which an observation can be expected with a given probability P or risk $\alpha = 1 - P$. As a statistical factor, the t value from Student's distribution should be used in the case of a normal distribution (cf. Section 2.2). The confidence interval for the mean, $\overline{x}$, is calculated for f degrees of freedom by

$$\Delta x = t(1 - \alpha/2; f)\, s_{\overline{x}} \tag{2.20}$$

Inserting the standard deviation for the mean according to Eq. (2.17) reveals:

$$\Delta x = \frac{t(1 - \alpha/2; f)\, s}{\sqrt{n}} \tag{2.20a}$$

The following example provides an overview of the quantities of descriptive statistics discussed.

Example 2.1 *Descriptive Statistics*

Table 2.3 summarizes the results for calculating the descriptive statistics for the spectrophotometric data given in Table 2.1.

Table 2.3 Descriptive statistics for the spectrophotometric measurements in Table 2.1.

Sample number, n	15
Arithmetic mean, $\overline{x}$	0.3486
Median	0.347
Geometric mean, G	0.3485
Variance s^2	0.000053
Standard deviation, s	0.00731
Relative standard deviation, $\%s_r$	2.096
Standard error, $s_{\overline{x}}$	0.00189
Confidence interval, Δx at $\alpha = 0.05$	0.00405
Minimum value, x_{min}	0.335
Maximum value, x_{max}	0.363
Range, R	0.028
Lower quartile, Q_1	0.343
Upper quartile, Q_3	0.353
Interquartile range, I_{50}	0.01
Skewness	0.259
Kurtosis	0.0452

Descriptive Statistics Graphically Illustrated: Box-and-Whisker Plots

Graphically, the most important details of descriptive statistics of data can be represented in a box-and-whisker plot or, for short, box plot (Figure 2.5). Along the variable axis, here the ordinate, a box is drawn, with the lower and upper quartile being the bottom and top of the box, respectively. The width of the box has no meaning.

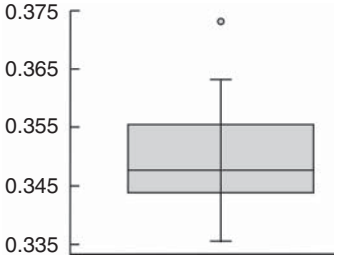


Figure 2.5 Box-and-whisker plot for the data in Table 2.1 with an additional outlier at an absorbance of 0.373.

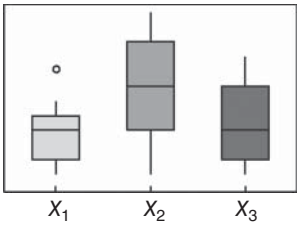
The whiskers are obtained as follows: the upper adjacent value is equal to the upper quartile plus 1.5 times the interquartile range:

$$\text{Upper adjacent value} = Q_3 + 1.5(Q_3 - Q_1) \tag{2.21}$$

The lower adjacent value is obtained from the lower quartile minus the 1.5 times the interquartile range:

$$\text{Lower adjacent value} = Q_1 - 1.5(Q_3 - Q_1) \tag{2.22}$$

Values outside the adjacent values are considered outliers. They are plotted as individual points. Box-and-whisker plots are not restricted to illustrating the univariate statistics of a single variable. Plots of several variables enable their different distribution characteristics easily to be compared.



Error Propagation

The uncertainty of analytical measurements originates from different sources. Among those are the following:

- Sampling
- Instrumental deviations, for example, a wrongly calibrated balance
- Reagent impurities
- Measuring conditions, for example, influence of temperature or humidity
- Matrix effects
- Round-off errors
- Contamination from the environment or between individual samples
- Operator effects
- Random influences.

In order to estimate the uncertainty of analytical results, the propagation of error cannot be considered alone for a single effect, for example, the reproducibility of the instrumental measurement, but *all* sources of uncertainty are to be taken into account for all steps of the analytical procedure.

The uncertainty of an analytical result is obtained from the Gaussian law of error propagation. Assuming a general analytical observation, y, that is dependent on m factors x_i according to a function f:

$$y = f(x_1, x_2, \ldots, x_m) \tag{2.23}$$

The uncertainty of the final result based on the deviation from the mean can then be described in dependence on the uncertainty

of the factors by

$$dy = f(dx_1, dx_2, \dots, dx_m) \tag{2.24}$$

As a local approximation of the unknown function *f*, the partial derivatives with respect to all factors are formed by calculating the total differentials:

$$dy = \left(\frac{\delta y}{\delta x_1}\right)_{x_2,\dots,x_m} dx_1 + \left(\frac{\delta y}{\delta x_2}\right)_{x_1,\dots,x_m} \times dx_2 \quad \dots + \left(\frac{\delta y}{\delta x_m}\right)_{x_1,x_2,\dots} dx_m \tag{2.25}$$

To link this quantity to the variance, the deviations are squared on both sides of Eq. (2.24):

$$(dy)^2 = \left[\left(\frac{\delta y}{\delta x_1}\right)_{x_2,\dots,x_m} dx_1 + \left(\frac{\delta y}{\delta x_2}\right)_{x_1,\dots,x_m} \times dx_2 \quad \dots + \left(\frac{\delta y}{\delta x_m}\right)_{x_1,x_2,\dots} dx_m\right]^2 \tag{2.26}$$

In squaring Eq. (2.26), two types of terms emerge from the right-hand side of the equation: squared terms and crossed terms for example,

$$\left(\frac{\delta y}{\delta x_1}\right)^2 dx_1^2 \text{ and } \left(\frac{\delta y}{\delta x_1}\right)\left(\frac{\delta y}{\delta x_2}\right) dx_1 dx_2$$

The square terms must always be considered, since they come out as positive terms independent of the sign of the partial derivatives. The cross terms may be either positively or negatively signed. As long as the factors are independent of each other, the cross terms will approximately cancel out so that they can be neglected. If the factors are dependent on each other, then the cross terms are to be included in calculating the uncertainty of the whole procedure.

The computation of the uncertainty on the basis of the variances is carried out after appropriate reshaping (normalization to $n-1$ measurements) without accounting for the crossed terms according to

$$s_y^2 = \left(\frac{\delta y}{\delta x_1}\right)^2 s_{x_1}^2 + \left(\frac{\delta y}{\delta x_2}\right)^2 s_{x_2}^2 \dots + \left(\frac{\delta y}{\delta x_m}\right)^2 s_{x_m}^2 \tag{2.27}$$

Frequently, the uncertainty is given as the standard deviation s_y, that is, the square root of Eq. (2.27).

The propagation of error is exemplified in Table 2.4 for typical cases. As can be seen, the variance for the observation y is

Table 2.4 Examples of error propagation for different dependencies of the analytical observation, *y*, on the factors, *x*.

Relationship	Calculation of uncertainty
$y = x_1 + x_2$ $y = x_1 - x_2$	$s_y^2 = s_{x_1}^2 + s_{x_2}^2$
$y = x_1 \cdot x_2$ $y = x_1/x_2$	$\frac{s_y^2}{y^2} = \left(\frac{s_{x_1}}{x_1}\right)^2 + \left(\frac{s_{x_2}}{x_2}\right)^2$
$y = x^a$	$\frac{s_y^2}{y^2} = \left(a\,\frac{s_x}{x}\right)^2$
$y = \log_{10} x$	$s_y^2 = \left(0.434\,\frac{s_x}{x}\right)^2$
$y = \text{antilog}_{10} x$	$\frac{s_y^2}{y^2} = (2.303 s_x)^2$

Table 2.5 Important areas according to the error integral.

Limits at the *x*-axis		Area under the total area *P* (%)	Tail area α (%)
x_1	x_2		
$\mu - 1\sigma$	$\mu + 1\sigma$	68.3	31.7
$\mu - 2\sigma$	$\mu + 2\sigma$	95.4	4.6
$\mu - 3\sigma$	$\mu + 3\sigma$	99.7	0.3
$\mu - 1.96\,\sigma$	$\mu + 1.96\sigma$	95.0	5.0
$\mu - 2.58\,\sigma$	$\mu + 2.58\sigma$	99.0	1.0
$\mu - 3.29\,\sigma$	$\mu + 3.29\sigma$	99.9	0.1

dependent on the *absolute* values of the variances of the individual quantities, if the factors obey an additive or subtractive relationship. For multiplicative or divisive relationships, the final variance is determined by the *relative* variances.

For reporting an error interval with a given statistical certainty, the expression in Eq. (2.27) is to be multiplied by a factor *k,* for example, $k = 2$ for a 95% probability (cf. Table 2.5) to find the results in this interval.

Uncertainty and Error

At the end of this section, the difference between error and uncertainty is to be stated. The *error* describes the difference between a measured and a true or expected value. It is expressed by a single value for a given measurement. In principle, an error can be corrected for.

The *uncertainty* of a result characterizes a range and is valid for a group of measurements or for all measurements considered. Correction of uncertainty is basically not feasible.

2.2 Statistical Tests

In Section 2.1, we used statistics only for the description of data. In many cases, however, it is necessary to draw conclusions from comparisons of data at a given statistical significance. These test methods are part of inferential statistics. For testing hypotheses, we need to learn about some more distributions, such as the *t-, F-,* and χ^2-distributions.

The Standard Normal Distribution

For testing hypotheses, the probability density function of the Gaussian distribution is standardized to an area of 1 and a mean of 0. This is done by introducing a deviate (*the standard normal variate*), z, which is obtained from the deviation of the observations, x, from the mean, μ, related to the standard deviation, σ:

$$z = \frac{x - \mu}{\sigma} \tag{2.28}$$

Transforming all x values into values of the deviate z results in a density function of the s*tandard normal distribution* according to the following model:

$$f(z) = \frac{1}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{z^2}{2}} \tag{2.29}$$

As seen already in Figure 2.2, the probability of the occurrence of an observation decreases with increasing variance or standard deviation, σ. The probability for which an observation is contained in a given range of z is denoted by P (cf. Figure 2.6).

This probability can also be derived from the distribution curve – the error integral $F(x)$ – of the Gaussian distribution (Figure 2.7). An analog consideration leads to the risk $\alpha = 1 - P$.

Important ranges for the error integral are given in Table 2.5 in connection with the related percentage and tail areas in units of the standard deviation.

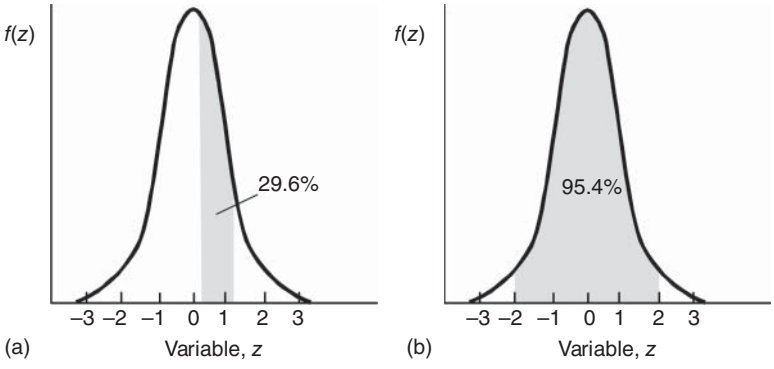


Figure 2.6 Examples for the integration of the Gaussian distribution in the ranges for the deviate z from 0.25 to 1.25 (a) and for $z = \pm 2$ (b).

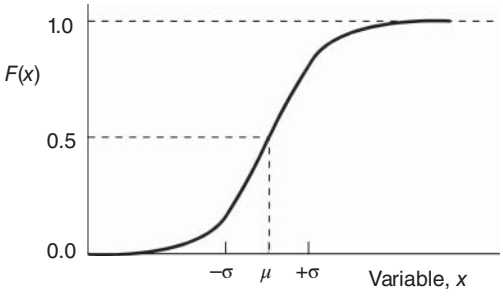


Figure 2.7 Distribution function (error integral) for the Gaussian distribution.

Example 2.2 *Normal Distribution*

Determination of phenol in wastewater revealed normally distributed values with a mean of $\mu = 0.6\,\mu g\,l^{-1}$ and a standard deviation of $\sigma = 0.04\,\mu g\,l^{-1}$. How large is the probability that in a subsequent measurement, the phenol concentration is contained in the range $0.61 - 0.65\,\mu g\,l^{-1}$?

According to Eq. (2.28), the value for the deviate z is calculated at the upper and lower limits by

$$z_1 = \frac{0.61 - 0.6}{0.04} = 0.25 \quad z_2 = \frac{0.65 - 0.6}{0.04} = 1.25$$

For the areas under the Gaussian curve, Table A.2 gives values(the curve covers values between 0 and $+z$) of $F(z_1) = 0.0987$ and $F(z_2) = 0.3944$. The difference between the areas provides the probability for the occurrence of the measurements, that is, the probability is $0.3944 - 0.0987 = 0.296$ or 29.6% (cf. Figure 2.6a).

Frequently employed areas under the Gaussian curve correspond to probabilities of 95% and 99% and the corresponding risk values of 5% and 1% (cf. Table 2.5). They are commonly used as significance levels in hypothesis testing.

The aforementioned probability-based considerations may serve as the most important foundations for derivation of statistically assured decisions. In general, in inferential statistics, the first step is the statement of a hypothesis, the significance of which is tested against a given risk α.

Testing Hypotheses

In general, the test of a hypothesis consists of five steps. First, the *null hypothesis,* H_0 and *alternative hypothesis,* H_1 are defined. Second, a test statistic is chosen. Third, the significance level is specified. Fourth, a decision rule is set up, based on the significance level and the distribution of the test statistics. Finally, the test statistic from the sample is calculated and the decision on the decision rule is made.

For example, the null hypothesis may postulate the randomness of samples in a group of observations. If the null hypothesis, H_0, is rejected, the alternative hypothesis H_1 is accepted. Since with practical measurements, only a limited number of samples out of the population will be available, the statistical tests usually cannot be directly based on the Gaussian distribution, but have to be performed on distributions derived from the normal distribution.

Comparison of a Mean with a True Value: One-Variable *t*-Test

Testing a sample mean, $\overline{x}$, obtained in an experimental measurement against a population mean, μ, from a normal distribution is carried out on the basis of a Gaussian or t test. In analytics, this is applied to compare an experimental mean with a true value.

The *significance level* represents the probability that the null hypothesis is falsely rejected.

The null hypothesis (H_0) reads: both samples belong to the same population, that is, the difference between the sample and the true value is random, or $\overline{x} = \mu$.

If the null hypothesis is rejected, the alternative hypothesis (H_1) is valid, $\overline{x} \neq \mu$, which suggests that the sample mean is different from the true value.

Before performing the test, the significance level is defined. Typically, the risk α is chosen to be 0.05 or 0.01 (cf. Table 2.5).

If the standard deviation σ of the population is known, the testing procedure can be based on the Gaussian test. The test statistic z is calculated for n parallel measurements by

$$z = \frac{|\overline{x} - \mu|}{\sigma} \sqrt{n} \tag{2.30}$$

The comparison of z with the quantile of the standard normal distribution (Table A.2) gives the result: If z is smaller than the quantile for a given risk level, α, and the number of degrees of freedom is $f = n - 1$, then the null hypothesis is accepted. Otherwise, if z is larger than α, the alternative hypothesis is accepted.

In practice, the population standard deviation, σ, is usually unknown, and the comparison of a sample mean with the true mean must be based on Student's t test. The test statistic is derived from Student's t-distribution as follows:

$$t = \frac{|\overline{x} - \mu|}{s} \sqrt{n} \tag{2.31}$$

where s is the estimate of the standard deviation and n the number of parallel measurements.

A comparison of the calculated test value t with the tabulated value of the t distribution (Table 2.6) at a given risk level α for the degree of freedom $f = n - 1$, $t(1 - \alpha/2; f)$, provides the decision on the test.

In the case that $t < t(1 - \alpha/2; f)$, the test is not significant, that is, the null hypothesis ($\overline{x} = \mu$) is accepted. In other words, the sample mean is only randomly different from the true one. For $t > t(1 - \alpha/2; f)$, the test comes out as statistically significant and the alternative hypothesis is accepted ($\overline{x} \neq \mu$).

Example 2.3 *Two-Sided t-Test*

The phenol concentration of wastewater was determined from $n = 3$ parallel determinations giving a mean of $= 0.513\ \mu g\ l^{-1}$ and a standard deviation of $0.05\ \mu g\ l^{-1}$. It is to be determined at a risk level of 5%, or $\alpha = 0.05$, whether a reference value of $\mu = 0.520\ \mu g\ l^{-1}$ is statistically different.

Null hypothesis H_0: $\overline{x} = \mu$

Alternative hypothesis H_1: $\overline{x} \neq \mu$

Inserting the values into Eq. (2.31) gives

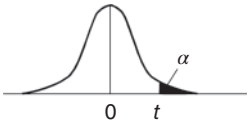
$$t = \frac{|0.513 - 0.520|}{0.05} \sqrt{3} = 0.242$$

The value of the t-distribution in Table 2.6 is $t(1 - \alpha/2 = 0.975; f = 2) = 4.303$. Since $t < t(1 - \alpha/2 = 0.975; f = 2)$, the differences between the sample mean and the true value are not significant, that is, the differences are random.

In the aforementioned example, a *two-sided* t-test was applied. A *one-sided* test is valid if, for example, if the test is whether a

Nonparametric (distribution free) tests do not require assumptions about the distribution of the population of the features to be tested. An example is the Wilcoxon test.

Table 2.6 Quantile of the one-sided Student's t distribution for three significance levels α and different degrees of freedom f. Note how the distribution approaches the Gaussian distribution if the degrees of freedom tend to infinity (cf. Table 2.5).



f	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.01$	f	$\alpha = 0.05$	$\alpha = 0.025$	$\alpha = 0.01$
1	6.314	12.706	31.821	21	1.721	2.080	2.518
2	2.920	4.303	6.965	22	1.717	2.074	2.508
3	2.353	3.182	4.541	23	1.714	2.069	2.500
4	2.132	2.776	3.747	24	1.711	2.064	2.492
5	2.015	2.571	3.365	25	1.708	2.060	2.485
6	1.943	2.447	3.143	26	1.706	2.056	2.479
7	1.895	2.365	2.998	27	1.703	2.052	2.473
8	1.860	2.306	2.896	28	1.701	2.048	2.467
9	1.833	2.262	2.821	29	1.699	2.045	2.462
10	1.812	2.228	2.764	30	1.697	2.042	2.457
11	1.796	2.201	2.718	40	1.684	2.021	2.423
12	1.782	2.179	2.681	50	1.676	2.009	2.403
13	1.771	2.160	2.650	60	1.671	2.000	2.390
14	1.761	2.145	2.624	70	1.667	1.994	2.381
15	1.753	2.131	2.602	80	1.664	1.990	2.374
16	1.746	2.120	2.583	90	1.662	1.987	2.369
17	1.740	2.110	2.567	100	1.660	1.984	2.364
18	1.734	2.101	2.552	300	1.650	1.968	2.339
19	1.729	2.093	2.539	800	1.647	1.963	2.331
20	1.725	2.086	2.528	∞	1.645	1.960	2.326

sample mean does not exceed a regulated value μ at a significance level α. This question is tested for by the following steps:

Null hypothesis $H_0: \overline{x} \leq \mu$
Alternative hypothesis $H_1: \overline{x} > \mu$

The tabulated value of the t-distribution is taken at the upper end of the t-distribution, that is, the value is $t(1-\alpha; f)$. If $t < t(1-\alpha; f)$, then the statistical test is not significant and the regulated value is not exceeded.

Example 2.4 *One-Sided t-Test*

Consider the determination of nitrate in drinking water. From four repetitive measurements, the following concentration values were obtained: 51.0, 51.3, 51.6, and 50.9 mg nitrate l^{-1}.

The maximum concentration allowed for nitrate concentration under European Union regulations is $50\,\mathrm{mg\,l^{-1}}$. To determine whether this regulated value is not exceeded, we perform a one-sided t-test at the upper end. By using the mean of $\overline{x} = 51.2$ and the standard deviation of $s = 0.316$, we calculate the t-value according to Eq. (2.31):

$$t = \frac{51.2 - 50}{0.316} \sqrt{4} = 7.59 \tag{2.32}$$

The tabulated t-value at an α level of 0.05 at the upper end of the t-distribution is $t(1 - \alpha = 0.95;\, f = 3) = 2.353$. Because $t > t(1 - \alpha = 0.95; f = 3)$, the null hypothesis is rejected and the alternative hypothesis is accepted, that is, the sample mean is greater than the limiting value.

The analogous question of whether a regulated value of $50\,\mathrm{mg\,l^{-1}}$ nitrate is exceeded also leads to a one-sided test. The hypotheses are as follows:

- Null hypothesis $H_0: \overline{x} \geq \mu$
- Alternative hypothesis $H_1: \overline{x} < \mu$.

The critical value is now to be taken at the lower end of the distribution, that is, $t(\alpha = 0.05;\, f = 3) = -2.353$ and the null hypothesis is to be accepted, if $t > t(\alpha; f)$. Since by calculation according to Eq. (2.31), $t > t(\alpha = 0.05; f = 3)$, the null hypothesis is accepted, that is, the regulated value is indeed exceeded.

Note: although the used tabulated t-value of -2.353 is not given in Table 2.6, it can be derived from the table since

$$t(1-\alpha; f) = -t(\alpha; f) \tag{2.33}$$

Also, it should be mentioned that in the case of the one-sided test in Eq. (2.31), the actual value, and not the absolute value, is used. To aid the understanding of the different hypothesis tests in more detail, summaries are provided in Table 2.7 and illustrations in Figure 2.8.

The principles of one- and two-sided tests are also applicable to tests based on other distributions.

Table 2.7 Overview on hypothesis testing based on Student's t-test.

t-test	Null hypothesis	Alternative hypothesis	t-Value for the acceptance of H_0	Figure
One-sided at the upper end	$H_0: \overline{x} \leq \mu$	$H_1: \overline{x} > \mu$	$t < t\,(1 - \alpha\,; f = n - 1)$	Figure 2.8a
One-sided at the lower end	$H_0: \overline{x} \geq \mu$	$H_1: \overline{x} < \mu$	$t > t\,(\alpha; f = n - 1)$	Figure 2.8b
Two-sided	$H_0: \overline{x} = \mu$	$H_1: \overline{x} \neq \mu$	$\lvert t \rvert < t\,(1 - \alpha/2; f = n - 1)$	Figure 2.8c

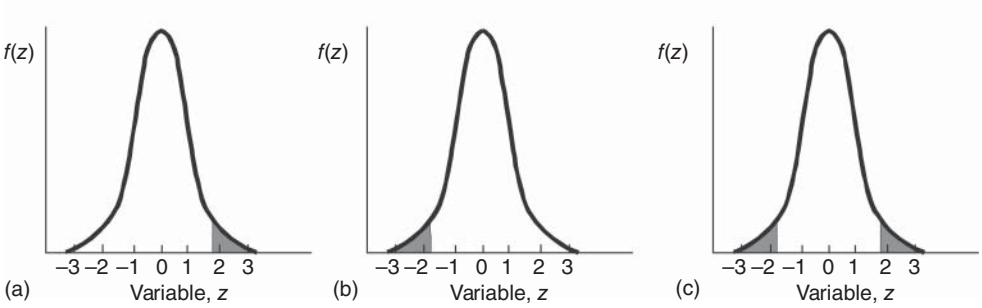


Figure 2.8 Illustration of critical areas for one-sided tests at upper (a) and lower (b) end and for two-sided tests (c).

p-Level Instead of Comparing Test Quantities In statistical software, it is rather unusual to carry out statistical tests the way we have done it earlier. The programs usually compute the specific level of significance at which the test is actually rejected. This value is called *p-level* or *level attained*. If the *p*-level is lower than or equal to the adjusted level, then the null hypothesis is rejected; otherwise, it is accepted.

The *p*-level is the right-tail significance level for the null hypothesis.

Example 2.5 *p-Level*

In Example 2.4, we evaluated the nitrate concentration of drinking water using a one-sided *t*-test at the upper end and testing the hypothesis that the regulated value of 50 mg l^{-1} nitrate is not exceeded ($H_0: \overline{x} \leq \mu\ H_1: \overline{x} > \mu$). The computed *t*-value corresponds to a significance level (*p*-level) of 0.002373. This value is lower than the specified level of $\alpha = 0.05$, so that the null hypothesis is rejected as before.

Comparison of Two Means: Two-Variable *t*-Tests

A statistical test makes only sense if the significance level is fixed in advance, even if the *p*-level will be calculated after the test.

A comparison of two sample means $\overline{x}_1$ and $\overline{x}_2$ is performed as follows:

$$t = \frac{|\overline{x}_1 - \overline{x}_2|}{s_d} \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \tag{2.34}$$

where n_1 and n_2 are the numbers of parallel determinations for the means $\overline{x}_1$ and $\overline{x}_2$, and s_d is the weighted averaged standard deviation:

$$s_d = \sqrt{\frac{(n_1 - 1)\ s_1^2 + (n_2 - 1)\ s_2^2}{n_1 + n_2 - 2}}$$

The null hypothesis is accepted if the two means $\overline{x}_1$ and $\overline{x}_2$ differ only randomly at risk level α, that is, if the calculated t value is lower than the tabulated value for $t(1 - \alpha/2; f = n_1 + n_2 - 2)$.

The assumption for this so-called *extended* t-test is the comparability of the variances of the two random samples, s_1^2 and s_2^2. Comparability here means that the two variances are equal at a given statistical significance level. The significance of the differences between the two variances is tested by means of an F-test (see the following discussion).

Where differences between the variances are not negligible, the general t-test, for example, after Welch, has to be applied:

$$t = \frac{|\overline{x}_1 - \overline{x}_2|}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.35}$$

The degrees of freedom f for the test statistics $t(P; f)$ is calculated according to the equation:

$$f = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}} \tag{2.36}$$

Comparison of Variances: *F*-Test

To compare the variances of two random samples or their standard deviations, Fisher's F-test is applied. The F-value is calculated from the variances s_1^2 and s_2^2 by

$$F = \frac{s_1^2}{s_2^2} \tag{2.37}$$

where $s_1^2 > s_2^2$. The null hypothesis is accepted, if the variances s_1^2 and s_2^2 differ only randomly, that is, if the calculated F-value is lower than the value of the F-distribution at risk level α and the degrees of freedom $f_1 = n_1 - 1$ and $f_2 = n_2 - 1$. The values of the F-distribution at significance levels of 0.95 and 0.99 are tabulated in Table 2.8.

Example 2.6 *Extended t-Test and F-Test*

The titanium content in steel is determined in two laboratories by means of atomic absorption spectrometry. The data are given in the table in the margin. After estimation of the variances for the determinations in the two laboratories, the means should be compared on the basis of the appropriate t-test.

Determination of *titanium contents* (absolute percentage) in two laboratories.

Laboratory 1	Laboratory 2
0.470	0.529
0.448	0.490
0.463	0.489
0.449	0.521
0.482	0.486
0.454	0.502
0.477	—
0.409	—

For comparison of the variances, the F-test is carried out according to Eq. (2.37). Based on the standard deviations:

$$\text{Lab. 1}: \; s_1 = 0.0229$$

$$\text{Lab. 2}: \; s_2 = 0.0182$$

the following F-value from the corresponding variances is obtained:

$$F = \frac{s_1^2}{s_2^2} = \frac{0.0229^2}{0.0182^2} = 1.58$$

Do not confuse "sided" and "tailed," that is, the F-distribution is one-tailed (asymmetric probability density function), but both one- or two-sided tests might be appropriate.

The critical F-value is taken from Table A.4 to be $F(1-\alpha/2; f_1, f_2) = F(0.975; 7, 5) = 6.85$

The calculated F-value is lower than the tabulated one, that is, the test is not significant and the variances differ only randomly.

For comparing the means, the extended t-test can be applied two-sided for *comparable variances* according to Eq. (2.34). Using the mean values

$$\text{Lab 1}: \; \overline{x}_1 = 0.467$$

$$\text{Lab 2}: \; \overline{x}_2 = 0.503$$

and the aforementioned standard deviations, we calculate

$$s_\mathrm{d} = \sqrt{\frac{(8-1)\,0.0299^2 + (6-1)\,0.0182^2}{8+6-2}} = 0.0211.$$

for the test quantity

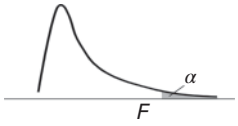
$$t = \frac{|\overline{x}_1 - \overline{x}_2|}{s_\mathrm{d}} \sqrt{\frac{n_1 n_2}{n_1 + n_2}} = \frac{|0.467 - 0.503|}{0.0211} \sqrt{\frac{6 \cdot 8}{6+8}} = 4.07$$

The critical t-value is (according to Table 2.6) $t(1-\alpha/2; n_1 + n_2 - 2) = t(1 - 0.05/2;\; 12) = 2.18$. This means that the calculated t-value is greater than the critical value, that is, the *two-sided test* is significant. The differences in the titanium determination in the two laboratories cannot be explained by random errors.

Testing for Distributions

The previous tests served the purpose of detecting differences between means or variances. The goodness of fit between an observed and a hypothetical distribution is done by two additional tests, the χ^2 and the Kolmogorov–Smirnov tests.

Table 2.8 *F*-quantiles for $\alpha = 0.05$ (*normal*) and $\alpha = 0.01$ (*bold*) and for different degrees of freedom f_1 and f_2.



f_2	$f_1 = 1$	2	3	4	5	6	7	8	9	10
1	161	200	216	225	230	234	237	239	241	242
	4052	**4999**	**5403**	**5625**	**5764**	**5859**	**5928**	**5981**	**6022**	**6056**
2	18.51	19.00	19.16	19.25	19.30	19.33	19.36	19.37	19.38	19.39
	98.49	**9900**	**99.17**	**99.25**	**99.30**	**99.33**	**99.36**	**99.37**	**99.39**	**99.40**
3	10.13	9.55	9.28	9.12	9.01	8.94	8.88	8.84	8.81	8.78
	34.12	**30.82**	**29.46**	**28.71**	**28.24**	**27.91**	**27.67**	**27.49**	**27.34**	**27.23**
4	7.71	6.94	6.59	6.39	6.26	6.16	6.09	6.04	6.00	5.96
	21.20	**18.00**	**16.69**	**15.98**	**15.52**	**15.21**	**14.98**	**14.80**	**14.66**	**14.54**
5	6.61	5.79	5.41	5.19	5.05	4.95	4.88	4.82	4.78	4.74
	16.26	**13.27**	**12.06**	**11.39**	**10.97**	**10.67**	**10.45**	**10.29**	**10.15**	**10.05**
6	5.99	5.14	4.76	4.53	4.39	4.28	4.21	4.15	4.10	4.06
	13.74	**10.92**	**9.78**	**9.15**	**8.75**	**8.47**	**8.26**	**8.10**	**7.98**	**7.87**
7	5.59	4.74	4.35	4.12	3.97	3.87	3.79	3.73	3.68	3.63
	12.25	**9.55**	**8.45**	**7.85**	**7.46**	**7.19**	**7.00**	**6.84**	**6.71**	**6.62**
8	5.32	4.46	4.07	3.84	3.69	3.58	3.50	3.44	3.39	3.34
	11.26	**8.65**	**7.59**	**7.01**	**6.63**	**6.37**	**6.19**	**6.03**	**5.91**	**5.82**
9	5.12	4.26	3.86	3.63	3.48	3.37	3.29	3.23	3.18	3.13
	10.56	**8.02**	**6.99**	**6.42**	**6.06**	**5.80**	**5.62**	**5.47**	**5.35**	**5.26**
10	4.96	4.10	3.71	3.48	3.33	3.22	3.14	3.07	3.02	2.97
	10.04	**7.56**	**6.55**	**5.99**	**5.64**	**5.39**	**5.21**	**5.06**	**4.95**	**4.85**

χ^2-Test The χ^2 goodness-of-fit test is used to test whether the observations of a population are sampled from a hypothetical distribution density function, for example, the normal distribution, at a given significance level α. The null and alternative hypotheses correspond to the following:

- H_0: the population is normally distributed with the mean and the variance s^2.
- H_1: the population is not normally distributed.

For performing the test, the interval for the observations is divided into k classes. The test quantity χ^2 is then obtained by comparing the theoretically expected distribution density with the observed frequency distribution according to (cf. Figure 2.9):

$$\chi^2 = \sum_{i=1}^{k} \frac{(h_i - f_i)^2}{f_i} \tag{2.38}$$

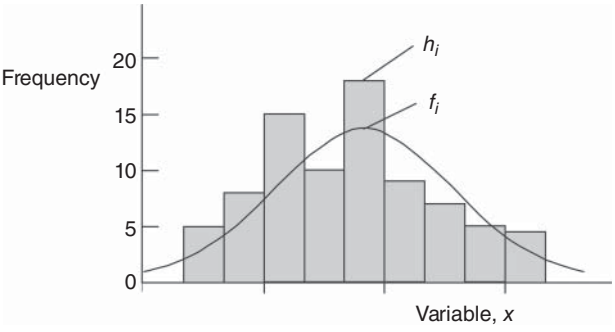


Figure 2.9 Schematic plot of the observed frequency, h_i, and the theoretically expected frequency, f, according to a normal distribution.

where h_i is the observed frequency in class i and f_i the theoretically expected frequency in class i.

By comparison of the calculated χ^2-value with the tabulated value of the χ^2-distribution, the null hypothesis is accepted if it is valid:

$$\chi^2 \leq \chi^2(1-\alpha;\ k-1-2)$$

and is rejected in the case

$$\chi^2 > \chi^2(1-\alpha;\ k-1)$$

Presuppositions for the applicability of the χ^2-test are that

- the frequency in the middle classes is at least 5;
- the frequency in the tail classes is at least 1.

The test is not restricted to testing for normal distribution.

Kolmogorov–Smirnov's Test for Small-Sample Collectives In practice, the χ^2-test often cannot be used because of the aforementioned presuppositions. An alternative here is the test by Kolmogorov–Smirnov, where a hypothetical distribution function $F_0(x)$ is used (cf. Figure 2.7) and not the density function of the distribution as with the χ^2-test.

In the null hypothesis, the distribution function observed $F_0(x)$ is tested by

$$H_0 : F(x) = F_0(x) \tag{2.39}$$

versus the alternative hypothesis

$$H_1 : F(x) \neq F_0(x) \tag{2.40}$$

For this comparison, a test statistics that characterizes the distance between the hypothetical and observed distribution function is used. This distance is computed from the maximum difference between the two distribution curves as d_{max} and is compared to the critical value of the quantity $d(1-\alpha, n)$. If $d_{max} < d(1-\alpha, n)$ the assumed distribution is accepted. Critical d values for testing for the *normal distribution* are given in Table A.6. Again, in practice, it is easier to evaluate the p level (see the previous discussion) provided by the common software packages in order to decide on the significance of the test.

Example 2.7 *Kolmogorov–Smirnov's Test*

The spectrophotometric measurements in Table 2.1 are to be tested versus a normal distribution by means of Kolmogorov–Smirnov's test at a significance level of $\alpha = 0.05$. In the first step, the empirical distribution function, $F(x)$, is evaluated as shown in Figure 2.10. For comparison of the hypothetical distribution function, the cumulative frequency by using the mean and the standard deviation of the data in dependence on the (standard normal) deviate z are plotted (cf. Eq. (2.28)).

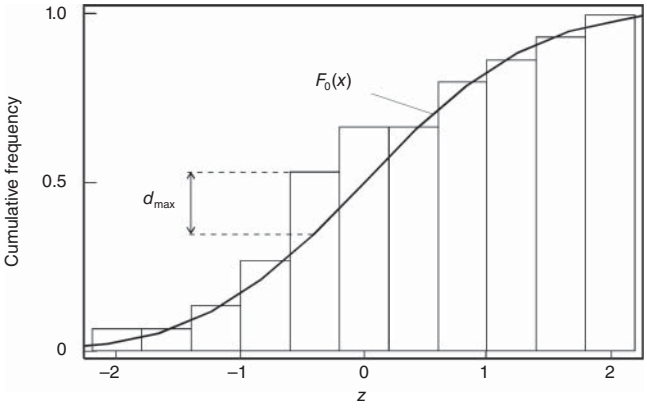


Figure 2.10 Determination of the test statistics of Kolmogorov–Smirnov's test as the maximum difference d_{max} between the empirical cumulative frequency distribution of the data and the hypothetical distribution function $F_0(x)$.

After evaluation of the hypothetical distribution function $F_0(x)$, the difference between both distributions is computed.

The maximum difference amounts to

$$d_{max} = 0.133$$

From Table A.6, a critical value of $d(0.95, 15) = 0.220$ results. Since $d_{max} < d(1 - \alpha, n)$, it can be assumed that the photometric data belong to a normal distribution.

Errors of the First and Second Kind

The risk α corresponds to the error of the first kind, that is, the null hypothesis is rejected, although it is true. The risk, however, cannot be chosen arbitrarily because of the error of the second kind (the null hypothesis being accepted although it is false), which then would considerably increase (see Table 2.9).

The *error of the first kind* is also termed *α error*, type I error, or rejection error. Other names for the *error of the second kind* are β error, type II error, or acceptance error.

Figure 2.11a illustrates the relationship between the error of the first kind, also called *α error*, and the error of the second kind (β error) for the comparison of two means. An error of the first kind is that the means are taken to be different, although they deviate from each other randomly. The error of the second kind is that it is wrongly stated that the two means are comparable.

A shift of the critical values to lower α values is related to an increase in the β error. A simultaneous decrease in the α and β errors is only feasible if the number of measurements, n, is increased. Since the width of the distribution is proportional to $\sigma/\sqrt{n}$, a larger number of measurements leads obviously to a narrower distribution (cf. Figure 2.11b).

In conclusion, the actual situation dictates the costs that the two kinds of error cause and whether an increase in the number of measurements is advantageous.

The failure to recognize a disease, for example, is much more critical than the precautionary therapy of a patient. In the latter case, an error of the first kind is valid, that is, from the clinical data, a healthy person is diagnosed as having a disease. Failure to recognize a disease from clinically abnormal data is an error of the

Table 2.9 Relationship between testing hypotheses and the errors of the first and second kind.

Decision for	Given	
	H_0	H_1
H_0	Correct decision $P = 1 - \alpha$	Error of the second kind $P = \beta$
H_1	Error of the first kind $P = \alpha$	Correct decision $P = 1 - \beta$

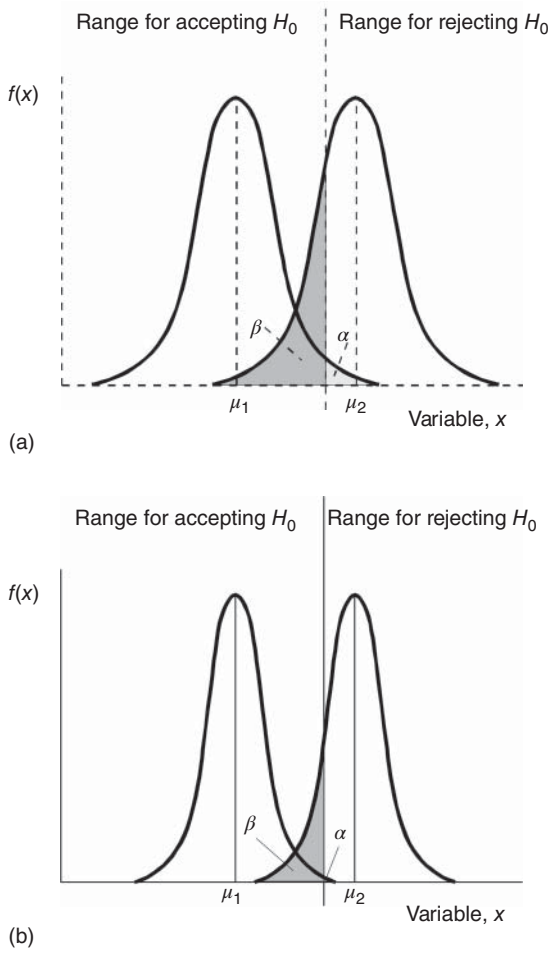


Figure 2.11 Errors of the first and second kind at larger (a) and smaller (b) standard deviation of the distribution.

second kind. In this instance, the data from a diseased patient lead to the diagnosis that the patient is healthy.

Tests for Outliers

An important application of statistical tests is the recognition of outliers. Here, we only consider outliers in a series of measurements. Outlier tests for methods of pattern recognition and modeling are introduced in Chapters 5 and 6, respectively.

Outliers are extraordinarily small or large observations in a series of measurements compared to the bulk of the data.

Before an outlier test is applied, reasons should be sought for these striking measurements, although uncritical elimination of outliers may lead to wrong conclusions. Think about the average age. A very high age of, say, 120 years would definitely be rejected as an outlier by the test to be discussed in the following.

Dixon's *Q*-Test

Testing for an outlier under the assumption of normal distribution can be carried out by the Dixon's test. This test uses the range of measurements and can be applied even in cases where only few data are available. The n measurements are arranged in ascending order. If the very small value to be tested as an outlier is denoted by x_1 and the very large striking value by x_n, then the test statistics is calculated by

$$Q_1 = \frac{|x_2 - x_1|}{|x_n - x_1|} \tag{2.41}$$

$$Q_n = \frac{|x_n - x_{n-1}|}{|x_n - x_1|} \tag{2.42}$$

The null hypothesis that the considered measurement is not an outlier is accepted if the quantity $Q \leq Q(1 - \alpha; n)$. Q values for a selected significance level of 0.99 are given in Table 2.10.

Table 2.10 Critical values for the *Q*-test at the 1% risk level.

n	$Q(0.99; n)$
3	0.99
4	0.89
5	0.76
6	0.70
7	0.64
8	0.59
9	0.56
10	0.53
11	0.50
12	0.48
13	0.47
14	0.45
15	0.44
20	0.39
25	0.36
30	0.34

Example 2.8 *Q Outlier Test*

Trace analysis of polycyclic aromatic hydrocarbons (PAH) in a soil revealed for the trace constituent benzo[a]pyrene the following values (in milligrams per kilogram dry weight):

$$5.30, 5.00,\ 5.10,\ 5.20,\ 5.10,\ 6.20,\ 5.15$$

By using the *Q*-test, whether the smallest and the largest values might correspond to outliers will be determined. The measurements arranged in ascending order are as follows:

x_1	x_2	x_3	x_4	x_5	x_6	$x_{n=7}$
5.00	5.10	5.10	5.15	5.20	5.30	6.20

Inserting the data into Eq. (2.41) results in the following *Q*-values for the corresponding smallest and largest values:

$$Q_1 = \frac{|5.10 - 5.00|}{|6.20 - 5.00|} = 0.083$$

$$Q_n = \frac{|6.20 - 5.30|}{|6.20 - 5.00|} = 0.75$$

From Table 2.10, we obtain $Q(1-\alpha = 0.99;\ n = 7) = 0.64$ as critical values at $\alpha = 0.01$. For the smallest value of 0.50, we obtain $Q_1 < Q(1-\alpha;\ n)$, that is, the value cannot be marked as an outlier. For the largest value, 6.20, the test gives $Q_n > Q(1-\alpha; n)$ significance, that is, it is an outlier.

Grubbs's Test

This test is also based on the assumption of a normally distributed population. It can be applied to series of measurements (3–150 measurements). The null hypothesis that x^* is not an outlier within the measurement series of n values is accepted at level α, if the test quantity T is

$$T = \frac{|\overline{x} - x*|}{s} < T(1-\alpha;\ n) \tag{2.43}$$

where the mean and the standard deviation s are calculated over all values. The distance of the suspicious values to the mean is determined by the test quantity T and related to the standard deviation of the measurements. Critical values for the quantity of Grubbs' test are given in Table 2.11.

Example 2.9 *Grubbs' Outlier Test*

The data for trace analysis of benzo[a]pyrene from Example 2.8 are to be investigated by the Grubbs' test. First, the mean ($\overline{x} = 5.29$) and the standard deviation ($s = 0.411$) of the data are calculated. Next, the smallest and largest values are inserted into Eq. (2.43) giving

$$T_1 = \frac{|5.29 - 5.00|}{0.411} = 0.71$$

$$T_n = \frac{|5.29 - 6.20|}{0.411} = 2.21$$

The critical test value at level $\alpha = 0.01$ is $T(1-\alpha = 0.99;\ n = 7) = 2.10$. As a consequence, the test is not significant $[T_1 < T(1-\alpha;\ n)]$ for the smallest value but significant $[T_n > T(1-\alpha; n)]$ for the largest one. The largest value (6.20) is therefore confirmed as an outlier.

Table 2.11 Critical values for Grubbs' test at two significance levels.

n	T (0.95; n)	T (0.99; n)
3	1.15	1.16
4	1.46	1.49
5	1.67	1.75
6	1.82	1.94
7	1.94	2.10
8	2.03	2.22
9	2.11	2.32
10	2.18	2.41
12	2.29	2.55
15	2.41	2.71
20	2.56	2.88
30	2.75	3.10
40	2.87	3.24
50	2.96	3.34

2.3 Analysis of Variance

One-Way Analysis of Variance

Analysis of variance (ANOVA) is used to analyze observations that depend on the operation of one or more effects. These effects are caused by factors the levels of which are also called *groups*, for example, different laboratories.

Let us start with an ANOVA in case of a single factor, termed a *one-way analysis of variance*. Table 2.12 demonstrates the general scheme of the measurements of this type of ANOVA.

To test for systematic differences between the groups, it is assumed that each measurement, y_{ij}, can be described by the sum of the total mean, $\overline{y}_{\text{total}}$, the group mean, $\overline{y}_j$, and the residual random error, e_{ij}, according to the following:

$$y_{ij} = \overline{y}_{\text{total}} + (\overline{y}_j - \overline{y}_{\text{total}}) + e_{ij} \tag{2.44}$$

where y_{ij} is the measurement of repetitions i in group j, $(\overline{y}_j - \overline{y}_{\text{total}})$ the laboratory bias estimated by the group mean $\overline{y}_j$ of group j, and e_{ij} the random error (residual).

The total variance, expressed as the sum of squares of deviations from the grand mean, is partitioned into the variances *within* the different groups and *between* the groups. This means that the sum of squares corrected for the mean, SS_{corr}, is obtained from the sum of squares *between* the groups or factor levels, SS_{fact}, and the residual sum of squares within the groups, SS_{R}:

$$SS_{\text{corr}} = SS_{\text{fact}} + SS_{\text{R}} \tag{2.45}$$

with

$$SS_{\text{corr}} = \sum_{j=1}^{q} \sum_{i=1}^{n_j} (y_{ij} - \overline{y}_{\text{total}})^2 \tag{2.46}$$

$$SS_{\text{fact}} = \sum_{j=1}^{q} n_j (\overline{y}_j - \overline{y}_{\text{total}})^2 \text{ and } \overline{y}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_{ij} \tag{2.47}$$

Table 2.12 Data scheme for a one-way analysis of variance.

Repetition	Group			
	1	2	...	q
1	y_{11}	y_{12}		y_{1q}
2	y_{21}	y_{22}		y_{2q}
⋮	⋮	⋮		⋮
n	$y_{n_1 1}$	$y_{n_2 2}$		$y_{n_q q}$
Mean	$\overline{y}_1$	$\overline{y}_2$		$\overline{y}_{\text{total}}$

$$SS_R = \sum_{j=1}^{q} \sum_{i=1}^{n_j} (y_{ij} - \bar{y}_j)^2 \tag{2.48}$$

$$\bar{y}_{total} = \frac{1}{n} \sum_{j=1}^{q} \sum_{i=1}^{n_j} y_{ij} \tag{2.49}$$

where q is the number of groups (factor levels), n_j the number of repetitive determinations per group j, and n the number of total measurements, that is, $n = \sum_{j=1}^{q} n_j$.

For deciding on the acceptance of the null hypothesis – that the groups belong to the same population and differ only randomly – an F test is performed (see Eq. (2.37)):

$$F = \frac{\frac{SS_{fact}}{(q-1)}}{\frac{SS_R}{(n-q)}} \tag{2.50}$$

The calculated F-value is compared to the critical value. If the calculated F-value is lower than the critical one, the test is not significant, that is, the groups differ only randomly.

Example 2.10 *One-Way Analysis of Variance*

For the determination of the potassium concentration of water, parallel determinations are carried out in four laboratories. Before aggregation of the determined values, the data is tested using a one-way analysis to see whether there are systematic differences between the laboratories' results.

Each laboratory performs triple determinations giving concentrations for potassium as summarized in Table 2.13, together with their means. The mean values of the different laboratories range between 10.20 and 10.77 mg potassium l^{-1}.

Table 2.13 Potassium concentration in mg·l^{-1} from triple determinations in four different laboratories.

Repetition	Laboratory (group)			
	1	2	3	4
1	10.2	10.6	10.3	10.5
2	10.4	10.8	10.4	10.7
3	10.0	10.9	10.7	10.4
Mean	10.20	10.77	10.47	10.53

The results for the sums of squares within and between the laboratories based on the potassium determinations in Table 2.13 are given in Table 2.14. This representation of data is called an *ANOVA table*.

Table 2.14 Results of a one-way analysis of variance for the potassium determinations as in Table 2.13.

Source of variation	Degrees of freedom	Sum of squares
Between laboratories, SS_{fact}	3	0.489
Within laboratories, SS_{R}	8	0.260
Corrected for the mean, SS_{corr}	11	0.749

The F-value is calculated according to Eq. (2.50):

$$F = \frac{\frac{0.489}{(4-1)}}{\frac{0.260}{(12-4)}} = 5.02$$

The value for $F = 5.02$ is greater than the tabulated one $F(1-\alpha = 0.95; f_1 = 3; f_2 = 8) = 4.07$ (see Table A.4). The test is therefore significant, that is, the differences between the laboratories cannot be considered to be different at random. The potassium determinations of at least one laboratory deviate systematically from each other.

By means of a one-way ANOVA, the effect of one factor can be investigated at different levels. In our example, the effect of a laboratory on the results of determinations was tested. In many applications, several factors have to be evaluated simultaneously. For example, apart from the laboratory effects, influences by the operator and the quality of the instrumentation are to be expected. These effects can be studied by using the two- or multiway analysis of variance.

Two-Way and Multiway Analysis of Variance

Consider the situation where two effects on analytical data are to be investigated simultaneously. This is done by a two-way ANOVA. Given the two factors, A and B, the following model can be assumed:

$$y_{ij} = \overline{y}_{\text{total}} + (\overline{y}_i^{\text{A}} - \overline{y}_{\text{total}}) + (\overline{y}_j^{\text{B}} - \overline{y}_{t\text{otal}}) + e_{ij} \tag{2.51}$$

Table 2.15 Data design for a two-way analysis of variance.

Factor A	Factor B				Mean
	1	2	...	q	$\overline{y}_i$
1	y_{11}	y_{12}		y_{1q}	$\overline{y}_1^{\mathrm{A}}$
2	y_{21}	y_{22}		y_{2q}	$\overline{y}_2^{\mathrm{A}}$
⋮					
p	y_{p1}	y_{p2}		y_{pq}	$\overline{y}_p^{\mathrm{A}}$
Mean $\overline{y}_i^{\mathrm{B}}$	$\overline{y}_1^{\mathrm{B}}$	$\overline{y}_2^{\mathrm{B}}$		$\overline{y}_q^{\mathrm{B}}$	$\overline{y}_{\mathrm{total}}$

where y_{ij} is the measurement of row i and column j, $\overline{y}_i^{\mathrm{A}}$ the mean of factor A in row I, $\overline{y}_j^{\mathrm{B}}$ the mean of factor B in column j, and e_{ij} the random error (residual).

The data are ordered such that the effects of the one factor form the rows, and those of the other factor the columns, of a matrix (Table 2.15).

Calculation of the means is based here on the following formulae:

$$\overline{y}_i^{\mathrm{A}} = \frac{1}{q} \sum_{j=1}^{q} y_{ij} \tag{2.52}$$

$$\overline{y}_j^{\mathrm{B}} = \frac{1}{p} \sum_{i=1}^{p} y_{ij} \tag{2.53}$$

$$\overline{y}_{\mathrm{total}} = \frac{1}{pq} \sum_{j=1}^{q} \sum_{i=1}^{p} y_{ij} \tag{2.54}$$

For the total sum of squares, Eq. (2.45) is again valid, that is, the total sum of squares corrected for the mean is obtained as sum of squares of factor A, SS_{A}, factor B, SS_{B}, and a residual sum of squares, SS_{R}:

$$SS_{\mathrm{corr}} = SS_{\mathrm{A}} + SS_{\mathrm{B}} + SS_{\mathrm{R}} \tag{2.55}$$

with

$$SS_{\mathrm{corr}} = \sum_{j=1}^{q} \sum_{i=1}^{p} (\mathrm{y}_{ij} - \overline{y}_{\mathrm{total}})^2 \tag{2.56}$$

$$SS_{\mathrm{A}} = q \sum_{i=1}^{p} (\overline{y}_i^{\mathrm{A}} - \overline{y}_{\mathrm{total}})^2 \tag{2.57}$$

$$SS_{\mathrm{B}} = p \sum_{j=1}^{q} (\overline{y}_j^{\mathrm{B}} - \overline{y}_{\mathrm{total}})^2 \tag{2.58}$$

$$SS_{\mathrm{R}} = SS_{\mathrm{corr}} - SS_A - SS_{\mathrm{B}} \tag{2.59}$$

In addition, measurements could be repeated for all factor combinations, as was shown for one-way ANOVA (see. Table 2.12). All sums must then be indexed over the repetitions for computation of the means (Eqs. 2.52 – 2.54).

Example 2.11 *Two-Way Analysis of Variance*

For preparation of a standard reference sample for determination of manganese in alloyed steel, a round-robin test is carried out. Four laboratories participate, each using three different analytical principles. Testing whether there are systematic differences between the laboratories and the principles of analysis is done by means of a two-way analysis. The results of the chemical analyses are given in Table 2.16.

Table 2.16 Analytical determinations of manganese (percentage mass) in steel carried out in four different laboratories based on three analytical principles.

Analytical principle	Laboratory				Mean
	1	2	3	4	$\overline{y}_i^{\mathrm{A}}$
1	2.01	1.96	1.99	2.03	2.00
2	1.97	2.05	2.04	1.99	2.01
3	2.05	2.06	2.11	2.12	2.09
Mean $\overline{y}_i^{\mathrm{B}}$	2.01	2.02	2.05	2.05	$\overline{y}_{\mathrm{total}} = 2.03$

The result of a two-way ANOVA according to Eqs. (2.55) – (2.59) is summarized in Table 2.17. Calculation of the F-values is done separately for the two factors.

- Effect of the factor *analytical principle:*

$$F_{\mathrm{A}} = \frac{\frac{SS_{\mathrm{A}}}{(p-1)}}{\frac{SS_{\mathrm{R}}}{(p-1)\ (q-1)}} = \frac{\frac{0.0175}{3-1}}{\frac{0.00788}{(3-1)\ (4-1)}} = 6.67$$

  For the critical F-value at a significance level of $\alpha = 0.05$ from Table 2.8, the value is $F(1-\alpha = 0.95; f_1 = 2; f_2 = 6) = 5.14$. Since $F_{\mathrm{A}} > F(1-\alpha; f_1; f_2)$, the test is significant, that is, the differences in the analytical principle are of a systematic nature.
- Effect of the factor *laboratory:*

$$F_{\mathrm{B}} = \frac{\frac{SS_{\mathrm{B}}}{(q-1)}}{\frac{SS_{\mathrm{R}}}{(p-1)\ (q-1)}} = \frac{\frac{0.00297}{4-1}}{\frac{0.00788}{(3-1)\ (4-1)}} = 0.753$$

Here, the critical F-value at a significance level of $\alpha = 0.05$ is $F(1-\alpha = 0.95;\ f_1 = 3;\ f_2 = 6) = 4.76$. The comparison of the calculated F-value with the tabulated one provides $F_B < F(1-\alpha;\ f_1;\ f_2)$, that is, the test is statistically not significant. The differences among the laboratories are random.

Table 2.17 ANOVA for two-way analysis of variance of the data in Table 2.16.

Source of variation	Degrees of freedom	Sum of squares	p level	F value	F_{tab} value
Principle, SS_A	2	0.01752	0.0299	6.67	5.14
Laboratory, SS_B	3	0.00297	0.559	0.753	4.76
Residual, SS_R	6	0.00788	—	—	—
Corrected for the mean, SS_{corr}	11	0.02837	—	—	—

In Table 2.17, the p-level, as well as the F-value, is given. For the significant effect of the analytical principle, this value is smaller than the previously assumed level of $\alpha = 0.05$. In the case of the nonsignificant effect of the laboratory, the p-level is larger than 0.05.

In principle, the ANOVA is not restricted to two factors. Common software usually offers multiway analysis where the tests can be performed for as many factors as the problem to be solved requires.

The effects of the different factors can be investigated independently of each other, or the interactions of factors can be considered in the ANOVA. The data for multiway ANOVA are ordered in a matrix, such that the rows represent the individual runs and the columns contain the factor levels as well as the responses (Table 2.18). This treatment of data corresponds to those used in experimental designs (see Section 4.2).

MANOVA: Multidimensional Analysis of Variance

The aforementioned treatment was limited to the study of factors on a single response or measurement, that is, only *one* dependent variable was investigated. In our examples, this was the potassium concentration in water or the manganese content in steel. More

Table 2.18 Schematic representation of factors and the response for a multiway (multifactor) analysis of variance at factor levels 1–4.

Experiment	Factor A	Factor B	Factor C	…	Response
1	1	1	4		y_1
2	2	1	3		y_2
:					
n	4	2	3		y_n

generally, it can be of interest to estimate the effects of factors on a complete spectrum or on an elemental pattern. For this purpose, an ANOVA is needed that can handle several responses, that is, a multidimensional ANOVA or MANOVA.

The data to be studied are arranged analogously to the design in Table 2.18, with the difference that several columns for the different dependent variables, for example, the wavelength of an optical spectrum, become necessary.

As the test statistics of a MANOVA, the Wilks λ value can be applied. This value is defined by the i eigenvalues of the data matrix λ_i as follows:

$$\lambda = \prod_i \left(\frac{\lambda_i}{1 + \lambda_i} \right) \tag{2.60}$$

For Wilks λ-values close to 0, the null hypothesis is rejected according to which the effect of the factors can be attributed to random effects. For λ-values close to 1, no factor effect can be deduced. The significance of the tests is again derived from testing the Wilks λ-value against a theoretical distribution. At least for large n, this distribution is the χ^2-distribution. In practice, the p-level is given by common software packages and compared with the risk level α.

General Reading

1. Dunn, O.J. and Clark, V.A. (2004) *Applied Statistics: Analysis of Variance and Regression*, 3rd edn, John Wiley & Sons, Inc., New York.
2. Graham, R.C. (1993) *Data Analysis for the Chemical Sciences – A Guide to Statistical Techniques*, VCH Publishers, New York.
3. Massart, D.L., Vandeginste, B.G.M., Deming, S.N., Buydens, L.C.M., De Jong, S., Lewi, P.J., and Smeyers-Verbeke, J. (1997) *Handbook of Chemometrics and Qualimetrics, Part A*, Elsevier, Amsterdam.
4. Miller, J.C. and Miller, J.N. (2001) *Statistics for Analytical Chemistry*,

4th edn, Ellis Horwood, Chichester.
5. Ellison, S.L.R., Rosslein, M., and Williams, A. (eds) (2000) *Quantifying Uncertainty in Analytical Measurements,* EURACHEM/CITAC Guide CG4, EURACHEM, London.
6. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. 1, Elsevier, Amsterdam.

Questions and Problems

1. Calculate the following descriptive statistics for the data on water hardness (mmol l^{-1}) given as follows: arithmetic mean, median, standard deviation, variance, standard error, confidence interval at a significance level of 0.01, range, and the interquartile distance – 8.02; 7.84; 7.98; 7.95; 8.01; 8.07; 7.89.

2. Characterize the data in Problem 1 graphically by a box-and-whisker plot.

3. Determination of vitamin E in salad oil was carried out by a routine voltammetric method and by flow injection analysis (FIA). The following results (% mass) were obtained:

Sample	Photometric	FIA
1	32.1	31.9
2	32.3	31.8
3	31.9	31.7
4	32.1	31.8
5	32.0	31.6
6	32.1	31.9
7	31.8	31.8

Compare the precision of the two methods by means of an F-test at a significant level of $\alpha = 0.05$.

4. Decide on the basis of the result in Problem 3 by use of an appropriate t-test whether the different results of the two methods are significant at a level of $\alpha = 0.05$.

5. Use Dixon's and Grubbs' methods to test for outliers in the water hardness data in Problem 1.

6. Five charges of gasoline (A, B, C, D, E) must be compared with respect to their octane rating. To account for confounded effects from the analyst and the day of analysis, a 5×5 Latin square is used in the experimental design.

Day	Analyst				
	1	2	3	4	5
1	A	B	C	D	E
2	B	C	D	E	A
3	C	D	E	A	B
4	D	E	A	B	C
5	E	A	B	C	D

The octane ratings obtained were as follows:

Day	Analyst				
	1	2	3	4	5
1	96.6	96.6	95.7	96.1	97.0
2	96.2	96.2	95.3	96.5	95.9
3	96.2	96.0	95.8	95.8	95.8
4	94.9	96.6	95.9	96.3	95.7
5	96.0	96.2	96.1	95.2	95.9

Use a commercial software package enabling multifactor ANOVA to decide whether there are systematic differences between the five charges at a significance level of $\alpha = 0.05$.

3
Signal Processing and Time Series Analysis

Learning Objectives

- To learn about digital filters, such as the nonrecursive moving-average and polynomial filters, as well as the recursive Kalman filter
- To apply filters for smoothing, derivation, and integration of signals
- To introduce signal manipulation based on Fourier and Hadamard transformations and their appropriate usage for filtering, convolution, deconvolution, integration, data reduction, and background correction
- To understand interpolating and smoothing splines
- To characterize correlated data by autocorrelation and cross-correlation functions.

In this chapter, we first learn about methods that can be used for *signal processing*. Analytical signals are recorded as spectra, chromatograms, voltammograms, or titration curves. These signals are monitored in the frequency, wavelength, or time domains. Typical aims are smoothing and filtering, derivation, area determination, transformation, or deconvolution of signals.

Methods for evaluation of analytical signals are as follows: transformation, smoothing, correlation, convolution, deconvolution, derivation, and integration.

Second, we investigate *time series*, for example, pH values measured in a lake over 1 year. In contrast to the independent random data considered in Chapter 2, here we deal with correlated data. The observations are assumed to be realizations of a stochastic process, where the observations made at different time points are statistically dependent. In order to recognize drifts, periodicities, or noise components in a time series, the correlation within the time series must be investigated.

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

3.1 Signal Processing

Digital Smoothing and Filtering

The extensive use of computers in the analytical laboratory is responsible for the fact that data are usually processed *digitally* (cf. Section 1.1). As a consequence, further processing of the digital signals is carried out by software rather than by hardware as previously.

Digital filters therefore dominate analog filters, possessing many advantages compared to analog ones:

- The digitized data can be immediately further processed.
- Temperature- or time-dependent noise does not exist. The only noise components to be accounted for are round-off errors.
- Digital filters are easier to modify and usually easier to understand.
- The filters implemented as software can be transferred to other computers.
- The filters can be easily optimized and tailor-made to specific processes.

Preprocessing of analytical signals by filtering serves the following purposes:

- Enhancement of the signal versus noise.
- Derivation of signals for subtraction of background and for improvement of visual resolution.
- Integration for quantitative signal evaluation.

Moving-Average Filter

Averaging is one means for *smoothing* of data.

The simplest filter operates on the basis of moving averages (Figure 3.1). Within a predefined window, a weighted linear combination of all the signal values is formed. The window determines the *filter width* and is moved successively along the equally spaced data. For calculation of the filtered value, y_k^*, from the raw signal values, y_k, the following relationship holds:

$$y_k^* = \frac{1}{2m+1} \sum_{j=-m}^{j=m} y_{k+j} \tag{3.1}$$

where k is the index for the actual data point, $2m+1$ represents the size of the window (filter width), and m is the variable to adjust the filter width.

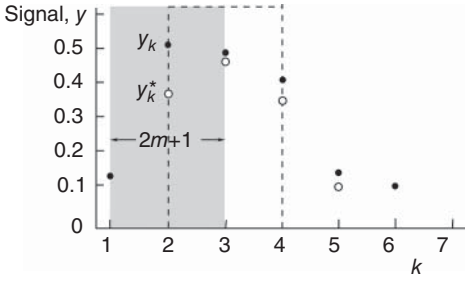


Figure 3.1 Moving-average filter for a filter width of $2m+1=3$, that is, $m=1$. Note that for the extreme points, no filtered data can be calculated, since they are needed for computing the first and last average. • Original signal value, ○ filtered signal value.

After applying the moving-average filter, the data contain *less noise.* In the case of structured data, the filter width has to be chosen such that the structure of the data, for example, of a peak, is not distorted.

Figure 3.2 demonstrates the filtering of raw data by using a 5-point moving-average filter (curve 1). In this example, the filter width of 5 points leads already to the distortion of peaks. This effect is enhanced if the filter width is further increased as demonstrated here for an 11-point filter (Figure 3.2, curve 2). The appropriate choice of the filter width is discussed as follows.

Too large a filter width decreases the original height of a peak and leads simultaneously to its broadening.

Polynomial Smoothing: Savitzky–Golay Filter

Very efficient smoothing of data is obtained with filters that weight the raw data differently. In the case of the moving-average filter, all the data were weighted by the same factor, that is, $1/(2m+1)$ (cf. Eq. (3.1)). A better fit results if weights are used that

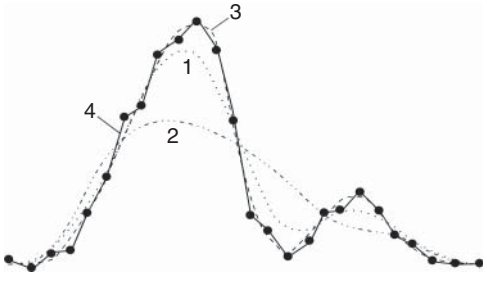


Figure 3.2 Filtering of a discrete analytical signal consisting of k data points with different filters: (1) A 5-point moving-average filter, (2) an 11-point moving-average filter, (3) a 5-point Savitzky–Golay filter, and (4) interpolated signal.

approximate the data by a polynomial of higher order (Savitzky–Golay filter). The filter coefficients are identical for polynomials of second and third order.

After deciding on the filter width, the filtered value for the kth data point is calculated by

$$y_k^* = \frac{1}{\text{NORM}} \sum_{j=-m}^{j=m} c_j y_{k+j} \tag{3.2}$$

where NORM is a normalization factor obtained from the sum of the coefficients c_j.

The filter coefficients c_j are tabulated in Table 3.1 for different filter widths. Figure 3.2, curve 3, demonstrates the effect of a Savitzky–Golay filter with a filter width of 5 points applied to the raw data. Compared to the 5-point moving-average filter, the obviously better fit can be seen.

Filter Selection

For selecting the most appropriate filter, some rules are followed:

- If the filter is applied to the data *repetitively*, the largest smoothing effect (>95%) is observed in the first application. Therefore, single smoothing is usually sufficient.
- The *filter width* should correspond to the full width at half maximum of a band or a peak. Too small a filter width results in unsatisfactory smoothing. Too large a filter width leads to distortion of the original data structure (cf. Figure 3.2).
- *Distortion* of data structure is more severe with respect to the area than to the height of peaks. Therefore, the filter width chosen is smaller if the height rather than the area is evaluated.

The influence of the filter width on the distortion of peaks can be quantified by means of the *relative filter width*, b_{relative}, according to

$$b_{\text{relative}} = \frac{b_{\text{filter}}}{b_{0.5}} \tag{3.3}$$

where b_{filter} is the filter width and $b_{0.5}$ is the full width at half maximum.

To illustrate the influence of the filter width on the distortion of the peak shape, the error of evaluating the height and area by polynomial smoothing of Gaussian and Lorentz peaks is shown in Figure 3.3 (cf. Figure 3.4).

If the peak area is measured, distortions become important only at relative filter widths greater than 1. In contrast, in the case of measuring the peak height, the relative filter width chosen should not be much larger than 0.5.

Table 3.1 Coefficients of the Savitzky–Golay filter for smoothing based on a quadratic/cubic polynomial according to Eq. (3.2).

Points	25	23	21	19	17	15	13	11	9	7	5
−12	−253	—	—	—	—	—	—	—	—	—	—
−11	−138	−42	—	—	—	—	—	—	—	—	—
−10	−33	−21	−171	—	—	—	—	—	—	—	—
−9	62	−2	−76	−136	—	—	—	—	—	—	—
−8	147	15	9	−51	−21	—	—	—	—	—	—
−7	222	30	84	24	−6	−78	—	—	—	—	—
−6	287	43	149	89	7	−13	−11	—	—	—	—
−5	322	54	204	144	18	42	0	−36	—	—	—
−4	387	63	249	189	27	87	9	9	−21	—	—
−3	422	70	284	224	34	122	16	44	14	−2	—
−2	447	75	309	249	39	147	21	69	39	3	−3
−1	462	78	324	264	42	162	24	84	54	6	12
0	467	79	329	269	43	167	25	89	59	7	17
+1	462	78	324	264	42	162	24	84	54	6	12
+2	447	75	309	249	39	147	21	69	39	3	−3
+3	422	70	284	224	34	122	16	44	14	−2	—
+4	387	63	249	189	27	87	9	9	−21	—	—

(*continued overleaf*)

Table 3.1 (*Continued*)

Points	25	23	21	19	17	15	13	11	9	7	5
+5	322	54	204	144	18	42	0	−36	—	—	—
+6	287	43	149	89	7	−13	−11	—	—	—	—
+7	222	30	84	24	−6	−78	—	—	—	—	—
+8	147	15	9	−51	−21	—	—	—	—	—	—
+9	62	−2	−76	−136	—	—	—	—	—	—	—
+10	−33	−21	−171	—	—	—	—	—	—	—	—
+11	−138	−42	—	—	—	—	—	—	—	—	—
+12	−253	—	—	—	—	—	—	—	—	—	—
NORM	5175	8059	3059	2261	323	1105	143	429	231	21	35

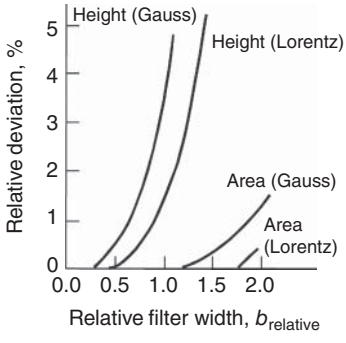


Figure 3.3 Relative error for smoothing Gaussian and Lorentz peaks in dependence on the relative filter width (Eq. (3.3)).

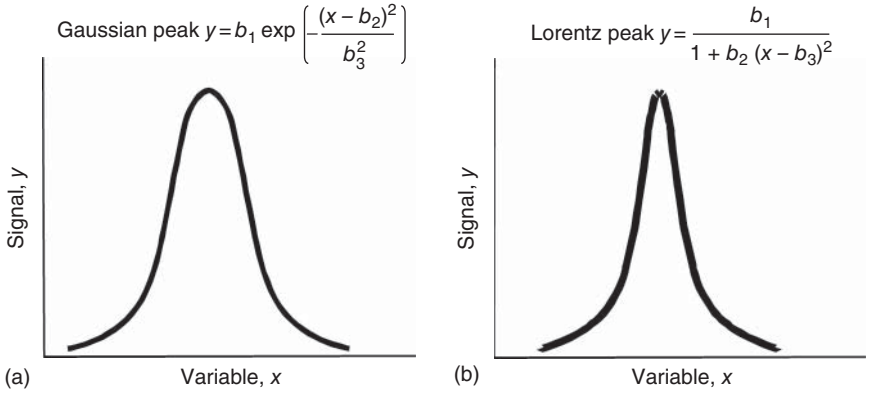


Figure 3.4 Shape of a Gaussian peak (see Eq. (2.2)) and a Lorentz peak. b_1, b_2, and b_3 are constants.

From Figure 3.3, it can also be understood that the distortion of peaks is dependent on their *shape.* Lorentz peaks are less distorted than Gaussian peaks.

Example 3.1 *Savitzky–Golay Filter*

The following data are to be smoothed by a quadratic 5-point polynomial filter:

Data point k	1	2	3	4	5	6	7	8	9	10	11
Signal value y_k	0.11	0.52	0.49	0.41	0.30	0.27	0.16	0.15	0.12	0.08	0.02

By using the coefficients from Table 3.1, we obtain the following for the smoothing polynomial of the kth data point:

$$y_k^* = -\frac{3}{35}y_{k-2} + \frac{12}{35}y_{k-1} + \frac{17}{35}y_k + \frac{12}{35}y_{k+1} - \frac{3}{35}y_{k+2}$$

As the first smoothed value, the third data point can be calculated, that is,

$$\begin{aligned} y_3^* &= -\frac{3}{35}y_{3-2} + \frac{12}{35}y_{3-1} + \frac{17}{35}y_3 + \frac{12}{35}y_{3+1} - \frac{3}{35}y_{3+2} \\ &= -\frac{3}{35}0.11 + \frac{12}{35}0.52 + \frac{17}{35}0.49 + \frac{12}{35}0.41 - \frac{3}{35}0.30 \\ &= 0.522 \end{aligned}$$

After moving the filter by one data point to the right, a value of 0.398 is obtained for the fourth data point and so on.

Recursive Filter: Kalman Filter

The aforementioned filters exploit only the raw data. They operate in a nonrecursive way. *Recursive filters* also use smoothed data. They were developed for smoothing fast processes in real time. The most popular of the recursive filters is the *Kalman filter*, which was developed to control the height of a missile skimming across the waves over sea and land.

A Kalman filter is based on a *dynamical system model*

$$\boldsymbol{x}(k) = \boldsymbol{F}\boldsymbol{x}(k-1) + \boldsymbol{w}(k-1) \tag{3.4}$$

and the *measurement model*

$$y(k) = \boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{3.5}$$

where $\boldsymbol{x}$ represents the state vector, y is the measurement, $\boldsymbol{F}$ is the system transition matrix, and $\boldsymbol{H}$ represents the measurement vector (or matrix). System noise is characterized by the vector $\boldsymbol{w}$ and measurement noise by the vector $\boldsymbol{v}$. The index k again denotes the actual measurement or time.

The recursive algorithm for Kalman filtering operates according to the scheme in Table 3.2.

For the system noise $\boldsymbol{Q}$ and the measurement noise R, the following assumptions are valid:

$$\boldsymbol{Q}(k) = E[\boldsymbol{w}(k)\boldsymbol{w}^{\mathrm{T}}(k)] \tag{3.11}$$

$$R(k) = E[\boldsymbol{v}(k)\boldsymbol{v}^{T}(k)] \tag{3.12}$$

Sequential estimation of the filter parameters requires initial values to be set for $\boldsymbol{x}$ and the covariance matrix $\boldsymbol{P}$. The latter

Table 3.2 Kalman filter algorithm.

Propagation of the filter states with time

$$\boldsymbol{x}(k) = \boldsymbol{F}(k)\boldsymbol{x}(k-1) \tag{3.6}$$

Propagation of the state covariance with time

$$\boldsymbol{P}(k|k-1) = \boldsymbol{F}(k)\boldsymbol{P}(k-1|k-1)\boldsymbol{F}^{\mathrm{T}}(k) + \boldsymbol{Q}(k) \tag{3.7}$$

Kalman gain

$$\begin{aligned}\boldsymbol{K}(k) = {} & \boldsymbol{P}(k|k-1)\boldsymbol{H}(k) \\ & \times [\boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{P}(k|k-1)\boldsymbol{P}(k|k-1)\boldsymbol{H}(k) + \boldsymbol{R}(k)]^{-1}\end{aligned} \tag{3.8}$$

State estimate update

$$\boldsymbol{x}(k|k) = \boldsymbol{x}(k|k-1) + \boldsymbol{K}(k)[y(k) - \boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{x}(k|k-1)] \tag{3.9}$$

Error covariance update

$$\boldsymbol{P}(k|k) = \boldsymbol{P}(k|k-1) - \boldsymbol{K}(k)\boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{P}(k|k-1) \tag{3.10}$$

quantity is chosen in the sense of a measure of the uncertainty of the initial estimate for $\boldsymbol{x}$. With correct models, the sequential estimates $\boldsymbol{x}(k)$ quickly become independent of the initial guess $\boldsymbol{x}(0)$, provided that sufficiently large values for $\boldsymbol{P}(0)$ are chosen, for example, 1–100 times $\boldsymbol{x}(0)$ works well for the diagonal values of $\boldsymbol{P}(0)$; the off-diagonal elements may be set to zero. A system error is specified only if the matrix $\boldsymbol{F}$ is not the identity matrix.

The Kalman filter can be applied for filtering, smoothing, and prediction. Additional applications for corrections of drift and in multicomponent analysis are known.

Signal Derivatives

The Savitzky–Golay filter can also be used for derivation of signal curves. For this, appropriate filter coefficients are inserted as given in Table A.7 for the first derivative and in Table A.8 for the second derivative.

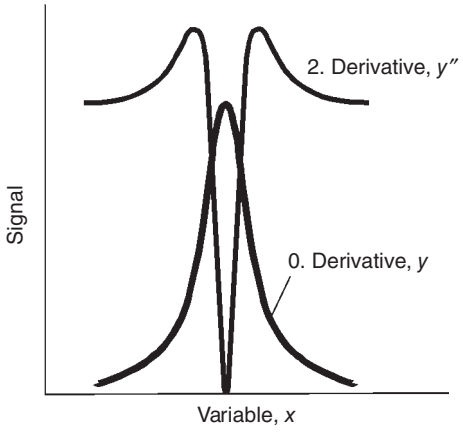
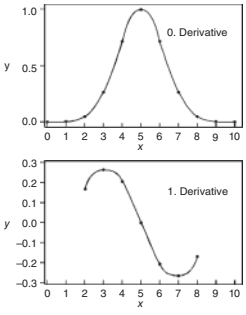


Figure 3.5 Second derivative of a peak based on a Lorentz function.



Derivative of a peak calculated by means of the Savitzky–Golay coefficients for the first derivative.

For example, the first derivative, y'_k, is obtained on the basis of a 5-point quadratic polynomial filter by

$$y'_k = -\frac{2}{10}y_{k-2} - \frac{1}{10}y_{k-1} + \frac{0}{10}y_k + \frac{1}{10}y_{k+1} + \frac{2}{10}y_{k+2}$$

Derivatives are useful for eliminating the *background* of a signal, determining the *peak position,* and improving the *visual resolution* of peaks.

In Figure 3.5, the second derivative of a peak is shown. At the peak maximum, the derivative has a pronounced minimum that is suitable for evaluation of the peak position. Compared to the original signal, the full width at half maximum is smaller for the peak in the second derivative. As a consequence, two peaks can be distinguished in the second derivative although they are not recognized in the original signal. In Figure 3.6, this is demonstrated for two peaks with a distance of 0.4 units of full width at half maximum.

Note that the noise increases if a signal derivative is formed. This is especially important in the case of quantification on the basis of derivatized signals.

Example 3.2 *Noise Characteristics for Derivatives of Signals*

The following *y*-signal values are given around an observation point *k:*

y_{k-2}	y_{k-1}	y_k	y_{k+1}	y_{k+2}
0.2	0.5	0.7	0.4	0.1

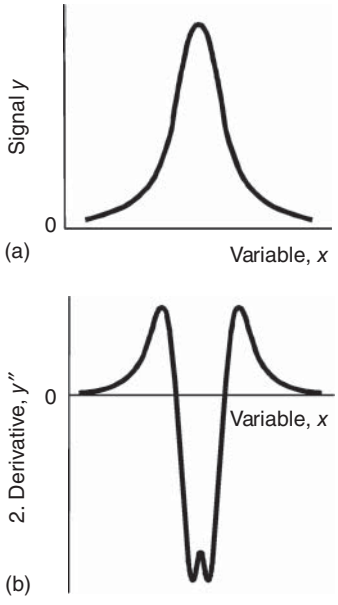


Figure 3.6 Visual resolution of two Lorentz peaks with a resolution of 0.4 full width at half maximum (a) and after formation of the second derivative (b).

The second derivative of a peak is easy to interpret, since it has a "*negative peak*" shape.

At this point *k*, the filtered value is calculated up to the second derivative on the basis of a 5-point Savitzky–Golay filter.

To characterize the corresponding noise, we consider the error propagation for the polynomial filter. For the Savitzky–Golay filter (see Eq. (3.2)), the result of error propagation (see Table 2.4) is expressed here by the standard deviation of the smoothed signal at point *k*, s_{y_k*}:

$$s_{y_k*} = \sqrt{\frac{1}{\mathrm{NORM}^2} \sum_{j=-m}^{j=m} c_j^2 y_{k+j}} \tag{3.13}$$

Calculation of the filtered values and their standard deviations by means of the tabulated filter coefficients leads to:

$$y_k^* = \frac{1}{35}((-3)\cdot 0.2 + 12\cdot 0.5 + 17\cdot 0.7 + 12\cdot 0.4 + (-3)\cdot 0.1)$$
$$= 0.623$$

Zeroth derivative:

$$sy_k^* = \frac{1}{35^2}((-3)^2\cdot 0.2 + 12^2\cdot 0.5 + 17^2\cdot 0.7$$
$$+12^2\cdot 0.4 + (-3)^2\cdot 0.1)^{1/2}$$
$$= 0.623$$

$$\text{relative error}: \frac{s_{y_k*}}{|y_k^*|} = \frac{0.522}{0.623} = 0.838$$

First derivative:

$$y'_k = \frac{1}{10}((-2)\cdot 0.2 + (-1)\cdot 0.5 + 0\cdot 0.7 + 1\cdot 0.4 + 2\cdot 0.1)$$
$$= -0.03$$

$$s_{y'_k} = \left(\frac{1}{10^2}\left((-2)^2 0.2 + (-1)^2 0.5 + 0^2 0.7 + 1^2 0.4 + 2^2 0.1\right)\right)^{1/2}$$
$$= 0.145$$

$$\text{relative error}: \frac{s_{y'_k}}{|y'_k|} = \frac{0.145}{0.03} = (4.83)$$

Second derivative:

$$s_{y''_k} = \left(\frac{1}{7^2}\left(2^2 0.2 + (-1)^2 0.5 + (-2)^2 0.7 + (-1)^2 0.4 + 2^2 0.1\right)\right)^{1/2}$$
$$= 0.316$$

$$y''_k = \frac{1}{7}(2\cdot 0.2 - 1\cdot 0.5 - 2\cdot 0.7 - 1\cdot 0.4 + 2\cdot 0.1)$$
$$= -0.243$$

$$\text{relative error}: \frac{s_{y''_k}}{|y''|_k} = \frac{0.316}{0.243} = 1.301$$

A comparison of the relative error between the zeroth and the second derivatives reveals an increase by a factor of 1.5. The error propagation for the first derivative demonstrates the limits of the procedure. Since the first derivative is close to zero at the peak maximum, an unrealistically large value is obtained for the relative error.

Integration for Area Determination

The area under a chromatographic peak, a spectroscopic band, or in thermal analysis is often directly proportional to the analyte concentration. Integration of signals is, therefore, important for quantitative data evaluation.

In the simplest case, the area is obtained from the sum of signals formed over the corresponding variable, such as time, energy, or wavelength. It is known, however, that random variations at the edge of a peak are much more severe than at the middle. Therefore,

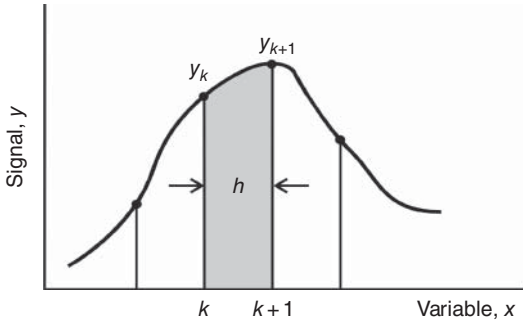


Figure 3.7 Trapezoidal rule for integration of a signal.

the integration formula should weight the middle signals higher than those at the edges. This leads to integration formulae, which are known as the *trapezoidal* and *Simpson rules*.

Figure 3.7 demonstrates the trapezoidal rule. The area, A_T, is calculated from the sum of the signal intensities, y_k, times the distance between two successive data points at the abscissa, h, by

$$A_T = \int_X y\,dx = h\left[\frac{1}{2}y_1 + \frac{1}{2}y_n + \sum_{k=2}^{n-1} y_k\right] \tag{3.14}$$

Note that the edge points ($k=1$ and $k=n$) are counted only once, so that they are multiplied in Eq. (3.14) by a factor of 0.5.

Still better results are obtained by the Simpson rule, which is based on approximation by a quadratic polynomial. Here, the area, A_S, is obtained by

$$A_S = \int_X y\,dx = \frac{h}{3}[y_1 + 4y_2 + 2y_3 + \cdots + 4y_{n-1} + y_n] \tag{3.15}$$

In general, the Simpson rule reads

$$A_S = \frac{h}{3}\left[y_1 + y_n + 4\sum_{k=2}^{n-1} \mathrm{I}y_k + 2\sum_{k=3}^{n-2} \mathrm{II}y_k\right] \tag{3.16}$$

where the sum $\sum_I$ is to be taken over all even and the sum $\sum_{II}$ over all odd numbers.

A prerequisite for application of the Simpson rule according to Eq. (3.6) is an odd number n of data points, that is, an even number of equidistant intervals. In the case of an odd number of data intervals, integration is started on the basis of the Simpson rule over an even number of intervals and the remaining area is calculated by another method. Typically, the 3/8 rule is used as the second method, which is based on a cubic polynomial.

Suppose for an odd number of intervals, the signal values are to be integrated from $k = 1$ to $k = n - 3$ by the Simpson rule. Then the remaining area is obtained by the 3/8 rule according to

$$A_{3/8} = \frac{3}{8}[y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n] \tag{3.17}$$

In this way, area determinations can be performed for an arbitrary number of data points.

Example 3.3 *Integration by Means of the Simpson Rule*

For the signal values in Example 3.1, the area is determined on the basis of a quadratic polynomial. According to Eq. (3.16), the resulting area is

$$A_S = \frac{1}{3}[0.11 + 0.02 + 4(0.52 + 0.41 + 0.27 + 0.15 + 0.08) + 2(0.49 + 0.30 + 0.16 + 0.12)] = 2.66$$

Transformations: Fourier and Hadamard

Mathematical transformations of the original data can be used for filtering if the transformed data are multiplied by appropriate filter functions and are subsequently back-transformed into the original data domain. Most frequently, the Fourier transformation (FT) is applied. In addition, we will also learn about Hadamard transformation (HT).

Apart from the filtering of data, transformations are also useful for convolution and deconvolution of analytical signals, for integration, for background correction, and for reduction of the number of data points, for example, in a spectrum.

Fourier Transformation

A signal in the time domain can be represented by a combination of periodic functions. This is shown in Figure 3.8 for the combination of two sine functions.

The maximum frequency that can be observed depends on the spacing of Δt according to

$$\nu_{max} = \frac{1}{2\Delta t}$$

This is called the *Nyquist frequency*.

Transformation of the signal from the time domain, $f(t)$, into the frequency domain, $F(\nu)$, provides two frequencies at 1 and 3 s^{-1}. Because any time-dependent or, in general, continuous signal graph can be considered a combination of sine and cosine functions, the FT is widely applicable.

In the time domain, information is obtained on the total amplitude, but not on the frequencies of the signal. The frequency domain contains information on the frequencies and

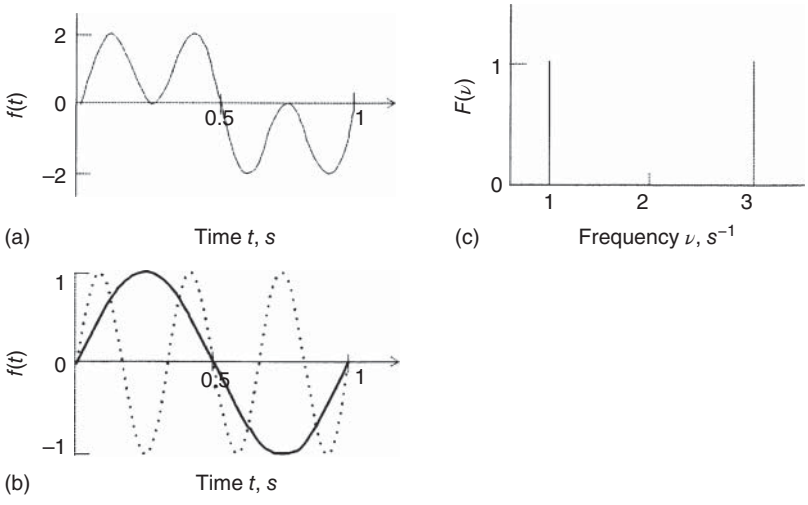


Figure 3.8 Fourier transformation: the sum signal (a) contains the sine functions with the periods $t_1 = 1$ s and $t_2 = 1/3$ s (b). The intensity is given in dependence on the frequency after Fourier transformation in (c).

their corresponding amplitudes of the signal, but information on the total amplitude is lost.

In conventional spectroscopy, measurements are carried out in the frequency domain. The intensity of radiation is recorded in dependence on the frequency or reciprocal wavelength. Some analytical methods, such as Fourier transform infrared (FT-IR) or pulsed nuclear magnetic resonance (NMR) spectroscopies, provide the information in the time domain. There, the opposite transformation into the frequency domain is of interest.

Discrete FT Digitized signal values in the time domain can be directly treated by discrete FT. For n discrete, equally spaced signal values, we obtain

Aliased frequencies are those that exist as different frequencies in a signal at the same time.

$$F(\nu) = \frac{1}{n}\sum_{t=1}^{n} f(t)\mathrm{e}^{-(j2\pi\nu\, t/n)} \tag{3.18}$$

for the transformation into the frequency domain where $j = \sqrt{-1}$.

The results are complex numbers consisting of a real and an imaginary part. The first term, $F(1)$, is always real and corresponds to the average of the data.

The following is used as the expression for the exponential function:

$$e^{-[j2\pi\nu/n]} = \cos(2\pi\nu t/n) - j\sin(2\pi\nu t/n) \tag{3.19}$$

Inverse FT Back transformation from the frequency into the time domain is carried out by inverse FT, that is,

$$f(t) = \sum_{\backsim=1}^{n} F(\nu) e^{(j2\pi\nu t/n)} \tag{3.20}$$

In practice, an algorithm for fast FT of data is applied. There, the number n of the k data points must be a power of 2, that is, $n = 2^k$. If complex conjugated numbers are used for Eq. (3.20), forward and back transformations can be performed with the same algorithm.

Wavelet transformations allow the use of arbitrary basis functions.

Hadamard Transformation

As an alternative to FT, the HT can be applied. The latter differs from FT in the basis function. HT is based on the Walsh function in contrast to the sine and cosine functions in FT (cf. Eq. (3.19)). The Walsh function, in the boundaries of ±1, is demonstrated in Figure 3.9.

Transformation of the original data, *y,* into the transformed values, y^*, can be easily represented by the following equation:

$$\boldsymbol{y}^* = \boldsymbol{H} \cdot \boldsymbol{y} \tag{3.21}$$

where $\boldsymbol{H}$ is the $(n \times n)$ HT matrix, $\boldsymbol{y}$ is the vector of the original n signal values, and $\boldsymbol{y}^*$ is the vector of the transformed n signal values.

As for the fast Fourier transformation (FFT) (c.f. Eq. (3.8)), in total, n of the k data points are used to represent a power of 2, that is, $n = 2^k$. The kth HT matrix is obtained by a simple

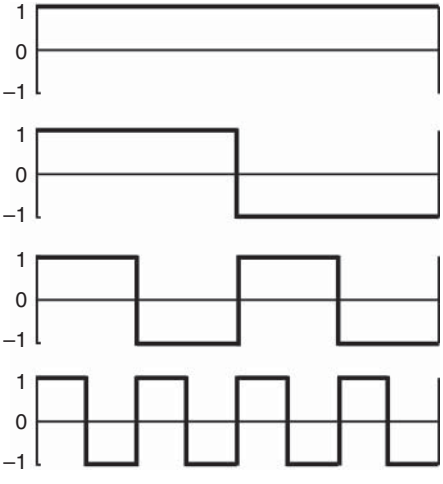


Figure 3.9 Walsh function as the basis function for Hadamard transformation (HT).

iteration rule:

$$\boldsymbol{H}_k = \begin{pmatrix} \boldsymbol{H}_{k-1} & \boldsymbol{H}_{k-1} \\ \boldsymbol{H}_{k-1} & -\boldsymbol{H}_{k-1} \end{pmatrix} \tag{3.22}$$

Example 3.4 *Hadamard Transformation*

Four data points are to be treated by HT. With $n = 2^k = 4$, we have $k = 2$. If we set $\boldsymbol{H}_0 = 1$, then we obtain for the matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ iteratively:

$$\boldsymbol{H}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\boldsymbol{H}_2 = \begin{pmatrix} \boldsymbol{H}_1 & \boldsymbol{H} \\ \boldsymbol{H}_1 & -\boldsymbol{H}_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

The transformation equation is

$$\begin{pmatrix} y_1^* \\ y_2^* \\ y_3^* \\ y_4^* \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

according to Eq. (3.21).

Multiplication of the equations for the transformed signal values provides

$$\begin{aligned} y_1^* &= y_1 + y_2 + y_3 + y_4 \\ y_2^* &= y_1 - y_2 + y_3 - y_4 \end{aligned} \tag{3.23}$$

and so on.

Insert your own numbers to transform your signals.

HT requires only simple arithmetical operations: addition and subtraction. This is in contrast to FT calculations, where complex numbers and trigonometric functions have to be processed. As a consequence, the algorithm for fast Hadamard transformation (FHT) is faster by a factor of about 3 than the FFT algorithm.

An additional advantage is the fact that the result of a HT is real, that is, there is no imaginary part.

Application of FT and HT

In many applications for filtering, integration, or data reduction, both FT and HT provide comparable results. If possible, both transformations should be tested in the actual application.

A particular transformation might be advantageous if one of its specific properties is required. Thus, HT is favored over FT if the speed of computation is important for the transformation. Furthermore, flat graphs, for example, an empty spectral range, are better approximated by HT than by FT, since FT models flat segments with waves of the highest possible frequency, which are poor substitutes for a straight line. On the other hand, FT is better suited to describe rounded shapes, such as structured bands or peaks. In these cases, HT reveals an undesired signal characterization because of its "box"-wave-like basis function.

Selected applications for both transformations are outlined as follows.

Signal Filtering For FT filtering, the signals are transformed from the time domain according to Eq. (3.18) into the frequency domain, $F(\nu)$. After that, multiplication by a filter function, $H(\nu)$, and back transformation with the inverse FT function (Eq. (3.20)) are performed. The filtered data, $G(\nu)$, are obtained by

$$G(\nu) = F(\nu)H(\nu) \tag{3.24}$$

Since the signal of interest usually lies in the intermediate frequency range, the filter function serves the purpose of filtering very high or low frequencies. To suppress high frequencies, which are typical for noise, a *low-pass filter* is required. For filtering low frequencies, corresponding, for example, to a drift, a *high-pass filter* is used (Figure 3.10).

FTs are especially suitable if different frequency parts are present in the original signal, or if, for example, the ac mains frequency has to be removed. From spectral analysis in the frequency domain, important information about the type of noise can be derived, and as a consequence, the signal-to-noise ratio can be systematically improved. *Shot noise* is recognized by a uniform frequency spectrum. This is typical for thermal

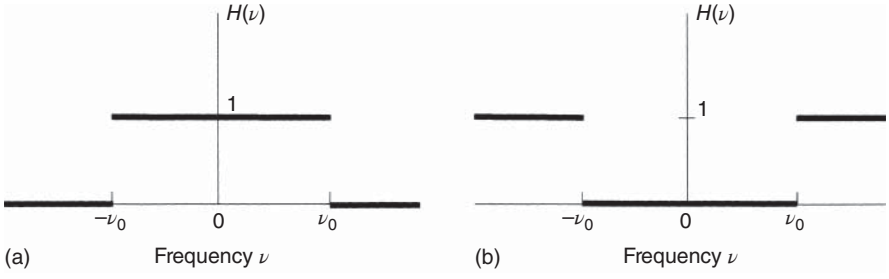


Figure 3.10 Low-pass filter (a) and high-pass filter (b) in the frequency domain.

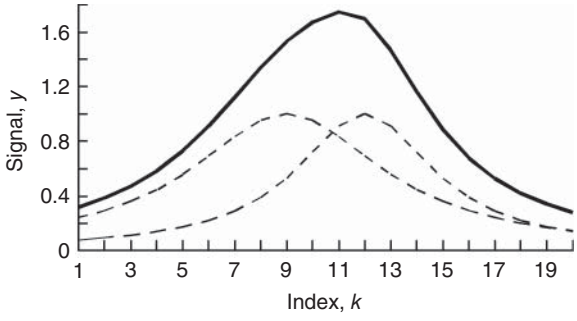


Figure 3.11 Decomposition (deconvolution) of a peak (*solid line*) into two underlying individual peaks (*dashed lines*) by means of Fourier transformation (FT).

noise or quantization noise of photomultiplier tubes. A continuously changing frequency spectrum is observed in the case of a drift (*1/f noise*). *Interference noise* is characterized by a specific frequency (range), for example, by a superposed 50 Hz power frequency.

Convolution and Deconvolution For restoration of an analytical signal that is distorted by an instrument function, or for decomposition of overlapping signals (cf. Figure 3.11), the use of FT is advantageous.

Let us denote the undistorted signal in dependence on time $f(t)$, the overlapped function, for example, instrument or Gaussian function, $h(t)$, and the observed function $g(t)$. Convolution (denoted by *) of the original signal with the interfering function is then expressed by

$$g(t) = f(t) * h(t) \tag{3.25}$$

In the frequency domain, this corresponds to a simple scalar product of the Fourier-transformed functions:

$$G(\nu) = F(\nu)H(\nu) \tag{3.26}$$

In order to evaluate the undistorted signal, f(t), Eq. (3.26) is solved for $F(\nu)$, that is,

$$F(\nu) = \frac{G(\nu)}{H(\nu)} \tag{3.27}$$

The result is back-transformed to the time domain according to Eq. (3.20).

Note the similarity to signal filtering in Eq. (3.24). One difference compared to deconvolution is the type of the

Components of an analytical signal in different frequency ranges:

Frequency	Component
Intermediate	Peak, band
Low	Drift
High	Noise

interfering function $H(\nu)$. In the case of filtering, this function is a simple step function: for deconvolution of signals, it must be determined whether a trapezoidal, triangular, Bessel, cosine, or Gaussian function would be the best.

Problems arise in the case of noise. If the signal noise is denoted by $N(\nu)$, Eq. (3.16) is modified to

$$G(\nu) = F(\nu)H(\nu) + N(\nu) \tag{3.28}$$

This leads to a different deconvolution in the frequency domain, that is,

$$\frac{G(\nu)}{H(\nu)} = F(\nu) + \frac{N(\nu)}{H(\nu)} \tag{3.29}$$

and

$$\widehat{F}(\nu) = F(\nu) + \frac{N(\nu)}{H(\nu)} \tag{3.30}$$

The function estimate obtained after deconvolution, $\widehat{F}(\nu)$, does not correspond to the FT of the undistorted signal, but also contains an unwanted noise component in the frequency domain.

An additional problem in FT deconvolution results from the finite number of data points. Back transformation frequently leads to wave-like curves, which cannot be attributed to real periodicities of the Fourier transform. To suppress these undesired side effects, an *apodization function* in the form of a triangular or parabolic function is applied. In analogy to Eq. (3.27), the deconvoluted signal is calculated by using an apodization function, $D(\nu)$, by

$$F(\nu) = \frac{D(\nu)\,G(\nu)}{H(\nu)} \tag{3.31}$$

For decomposition of signals, the concrete form of the interfering signal is determined separately. So, for example, for the deconvolution of an overlapping chromatographic signal into Gaussian peaks (cf. Eq. (2.2)), the full width at half maximum or the standard deviation must be known.

Integration From the equations in Example 3.3, we recognize that the first row in the HT matrix, which leads to the first Hadamard coefficient, y_1^*, by multiplication with the original data, is equal to the sum of all signal values (Eq. (3.13)). Based on that sum, the integral over all signal values can be deduced, if, for example, the trapezoidal formula according to Eq. (3.4) is applied. The area, A, is calculated by subtraction of half of the sums of the first and last

signal values:

$$A = y_1^* - \frac{1}{2}(y_1 + y_n) \tag{3.32}$$

A similar result is obtained for FT. According to Eq. (3.8), the first Fourier coefficient, $F(1)$, corresponds to the average of signal values. Multiplication with the data number n provides the sum over all values for integration by Eq. (3.32). In the latter case, we assumed that only the real part of FT is considered. Fortunately, for any real function, the first coefficient, $F(1)$, should be always real.

Data Reduction and Background Correction Reduction of data points is important if, for example, further processing of a spectrum is only feasible if the number of data points is decreased. For reduction of measurements in the original data vector, the data are transformed by means of FT or HT. After that, back transformation is performed on the basis of a limited number of Fourier or Hadamard coefficients. For back transformation, the coefficients are sorted according to importance, and the effect of less important coefficients is thus eliminated (cf. Zupan, Section 3.3). Practically, the number of coefficients is not changed, but unimportant coefficients are set to zero.

If the most significant coefficient, $F(2)$ for FT or y_2^* for HT, is set to zero, then a background correction results, explained by the fact that the coefficients correspond to the basis function of those transformations. Because of the box-wave-like basis function of HT (cf. Figure 3.9), practical usage of this kind of background correction can only be recommended for FT.

In Figure 3.12, the results of forward and backward transformations of a spectrum are demonstrated. Even if only half of the original 32 coefficients are used, that is, the 16 most important ones, the original data are quite well reproduced. In the case of back transformation with only few coefficients, the different basis functions of the two transformations can be easily recognized, that is, for FT, the trigonometric and for HT, the Walsh function.

In Figure 3.12, a filtering effect can also be seen in connection with the reduction of coefficients. The difference from the aforementioned methods of signal filtering lies only in the concrete choice of the filter coefficients. The principle of back transformation is the same here as in the case of data reduction.

Spline Functions

In addition to the smoothing methods based on digital filters and on transformations, some further possibilities exist for signal smoothing. Among these are the following:

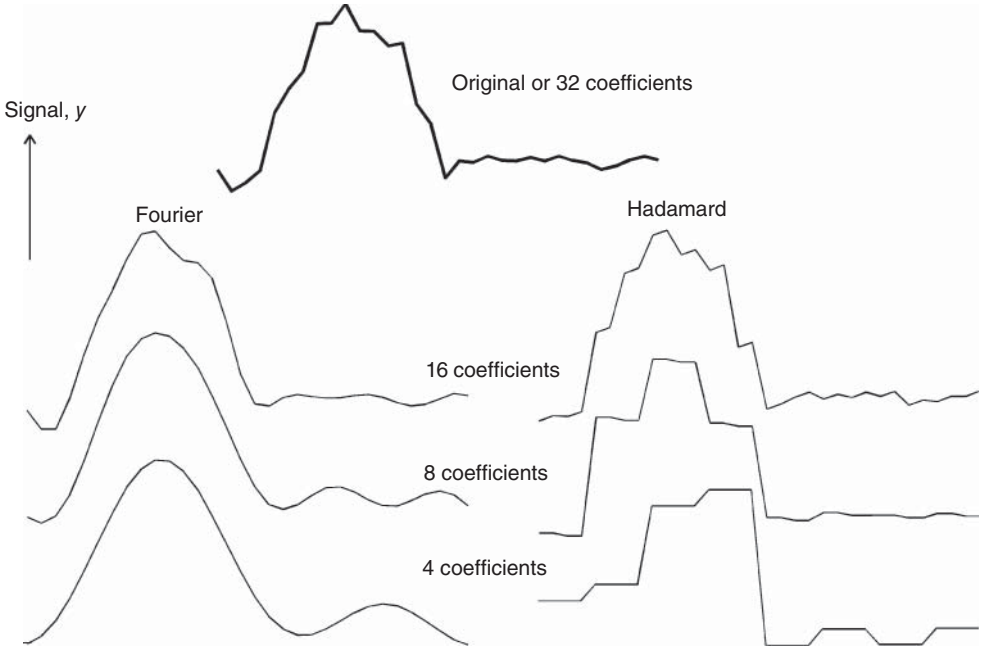


Figure 3.12 Transformation of signals from 32 data points and back transformation by FT and HT using different numbers of coefficients. The numbers correspond to the remaining coefficients except the first coefficient, which represents the average (FT) or the sum (HT) of the spectrum.

- *Local approximations:* here, the functional dependence or signal curve is split into intervals, and these intervals are fitted piecewise, for example, by straight-line models. Unfortunately, no smooth curves result, but discontinuities emerge in the derivatives of the curve.
- *Modeling with known basis functions:* if it is known, that, for example, a spectroscopic band obeys the Lorentz function, then signal processing (smoothing, analytical derivatives) is feasible by estimation of the parameters of that basis function. The parameters are commonly estimated by nonlinear regression analysis (cf. "Nonlinear Regression Analysis" Section). A typical problem with this method is that no appropriate basis function can be found for the entire measurement range.
- *Spline functions:* these represent a compromise between a polygon trace and an interpolation polynomial of higher order. The main advantage of spline functions is their differentiability in the entire measurement domain.

The spline functions will be considered in more detail as follows.

Interpolating Splines

To construct an interpolating spline, the x range is split into several intervals, separated by the so-called *knots.* The knots may be identical with the index points at the x variable axis.

An *interpolating* cubic spline $S(x)$ for the observations at the abscissa grid $x_1 < x \ldots < x_n$ fulfills the following conditions:

- The spline $S(x)$ is interpolating, that is, at the knots $k = 1, \ldots, n$, the measured value, y_k, is equal to the spline value $S(x_k)$.
- Within the knots k, $S(x)$ obeys the continuity constraint on the function and on its twofold derivatives.
- $S(x)$ is a cubic function in each considered subrange $[x_k, x_{k+1}]$ for $k = 1, \ldots, n-1$.
- Outside the range from x_1 to x_n, $S(x)$ is a straight line.

Splines with variable knot locations are termed *adaptive splines.*

A spline function that fulfills the aforementioned conditions is shown in Figure 3.13.

The cubic function is defined by

$$y = A_k(x - x_k)^3 + B_k(x - x_k)^2 + C_k(x - x_k) + D_k \tag{3.33}$$

where A_k, B_k, C_k, and D_k are the spline coefficients at data point k.

For a fixed interval between the data points x_k and x_{k+1}, the following is valid for the signal values and their derivatives:

$$\begin{aligned} y_{k+1} &= A_k(x - x_k)^3 + B_k(x - x_k)^2 + C_k(x - x_k) \\ y_k &= D_k \\ y'_k &= C_k \\ y'_{k+1} &= 3A_k(x - x_k)^2 + 2B_k(x - x_k) + C_k \\ y''_k &= 2B_k \\ y''_{k+1} &= 6A_k(x - x_k) + 2B_k \end{aligned} \tag{3.34}$$

By additional reshaping, the spline coefficients can be determined from Eq. (3.34).

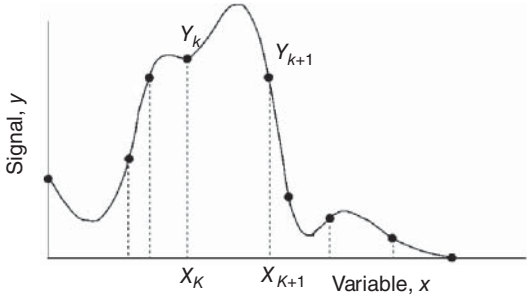


Figure 3.13 Interpolating spline function.

Smoothing Splines

The spline coefficients can also be determined in such a way that the data are smoothed simultaneously. For this, the ordinate values $\hat{y}_k$ are calculated such that the differences from the observed values y_k are positive proportional jumps r_k in their third derivative at point x_k:

$$r_k = f_k'''(x_k) - f_{k+1}'''(x_{k+1}) \tag{3.35}$$

$$r_k = p_k(y_k - \hat{y}_k) \tag{3.36}$$

The proportionality factors p_k are, for example, determined by cross-validation (cf. "Factorial Methods" Section).

In contrast to polynomials, spline functions may approximate and smooth any kind of curve shape.

Problems, however, arise if the intervals between the knots are not narrow enough and the spline begins to oscillate (cf. Figure 3.13). Also, in comparison to polynomial filters, many more coefficients are to be estimated and stored, since in each interval, different coefficients apply. An additional disadvantage is valid for smoothing splines, where the parameter estimates are not expectation-true. The statistical properties of spline functions are, therefore, more difficult to describe than in the case of linear regression (cf. Section 6.1).:

Wavelet Analysis

Fourier analysis is very useful to describe the signal's frequency content. However, it has a serious drawback. In a typical FT of a signal, it is impossible to tell *when* an event has happened because time information is lost. For all nonstationary signals that contain drift, trends, or abrupt changes, it is important to keep the time information in the transformation. In principle, time information can be found in the imaginary part of a complex FT. However, since only the real part is usually considered, the time information will not be available.

In wavelet analysis, the concept of frequency can be replaced by the idea of scale. Unlike sine waves known from FT analysis (cf. Figure 3.8), a *wavelet* is a waveform with an average value of zero and limited duration. Wavelets tend to be irregular and asymmetric. In Figure 3.14, the wavelets of Classes Haar, Daubechies, Coiflet, Symmlet, Morlet, and Mexican Hat are given as examples.

Wavelet analysis breaks up signals into shifted and stretched versions of the analyzing wavelet. Shifting is called *translation* and means delaying or hastening the onset of a wavelet. Shifting the wavelet function $\psi(x)$ by b reveals the shifted wavelet function $\psi(x-b)$, as depicted in Figure 3.15. Stretching is termed a *dilation* and is related to widening the basis wavelet with a factor *a.*

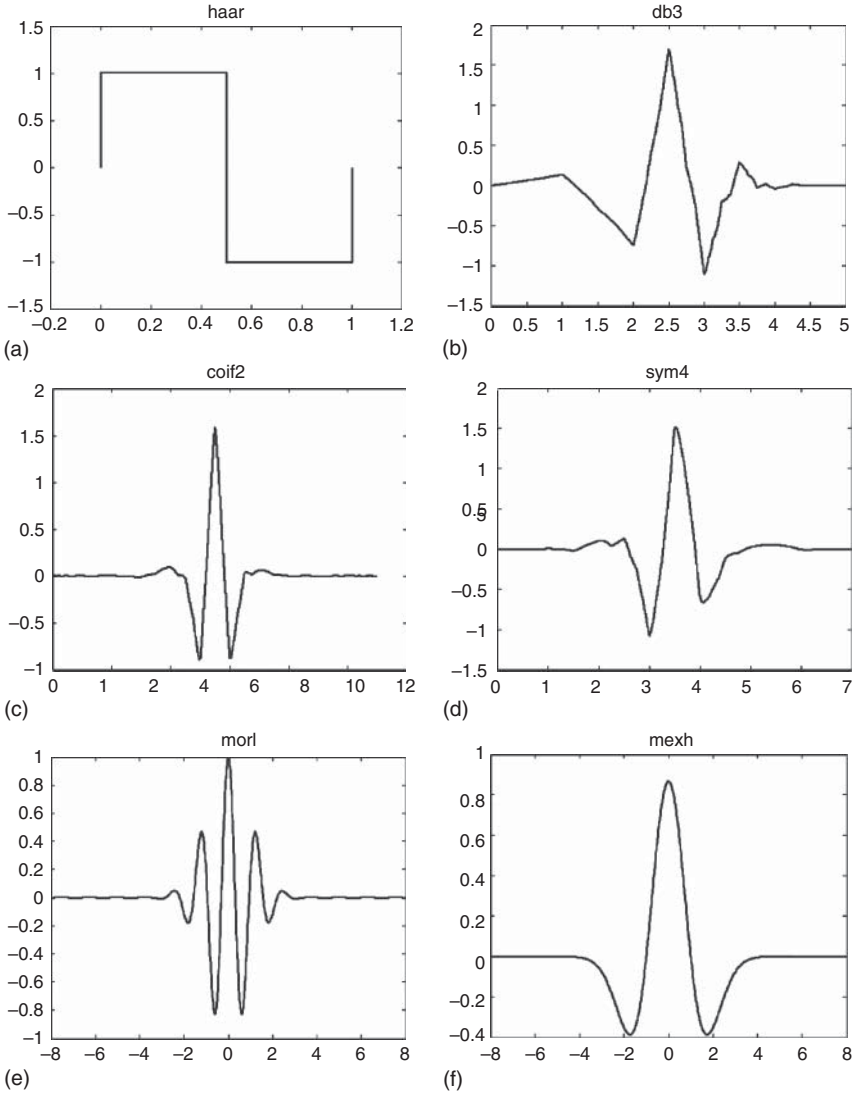


Figure 3.14 Wavelets of classes Haar (a), Daubechies (b), Coiflet (c), Symmlet (d), Morlet (e), and Mexican hat (f).

A series of wavelets $\psi_{i,j}$ is obtained according to

$$\psi_{i,j} = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{3.37}$$

where $a = 2^i$ and $b = \frac{j}{2^i}t$ for $1 < j < 2^i$.

The wavelet transform of $\boldsymbol{y}^*$ could be computed by an orthonormal wavelet basis matrix $\boldsymbol{W}$, the same approach as used in FT

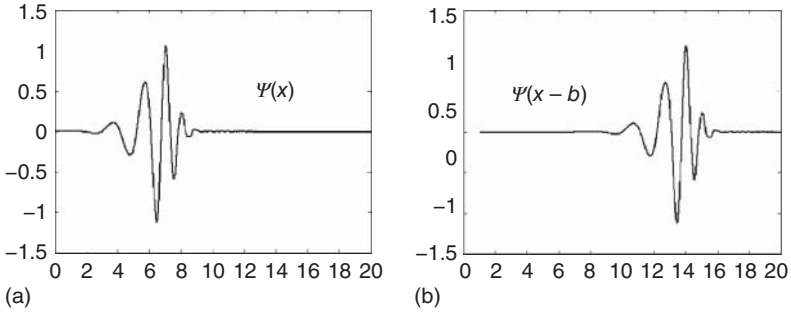


Figure 3.15 Shifting the Daubechies-7 wavelet function $\psi(x)$ (a) and by the value b (b).

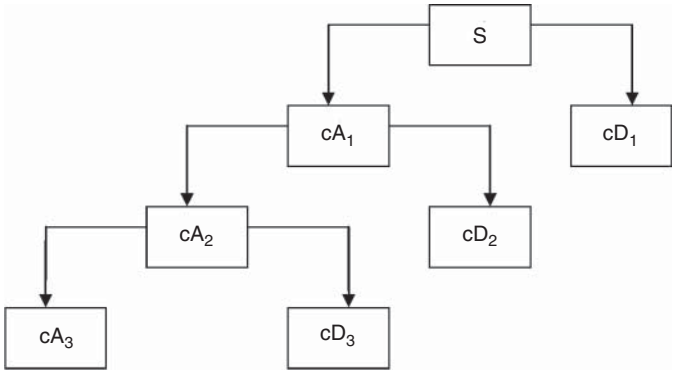


Figure 3.16 Multilevel decomposition of a signal by discrete wavelet transformation (DWT).

analysis, that is, $\boldsymbol{y}^* = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{y}$. Practically, a pyramidal scheme is used to compute $\boldsymbol{y}^*$ even faster than with FFT.

The continuous wavelet transformation approach is replaced by discrete wavelet transformation (DWT), where scales and positions are computed based on the powers of 2. Filtering is performed separately for the low and high frequency parts of a signal. The high-scale, low-frequency components are called *approximations*, A, and the low-scale, high-frequency components are details, D. The scheme is sketched in Figure 3.16 for decomposition of the original signal, S, into approximation coefficients, cA, and detail coefficients cD up to a level of 3.

Application of this scheme is exemplified in Figure 3.17 for a signal from mass spectrometry of a protein. Note how the approximation cA_3 provides a filtered signal without the distortion of the original peaked signal that occurs with conventional digital filters. Reconstruction of the signal is then feasible by summation over the coefficients in the four parts cA3, cD3, cD2, and cD1.

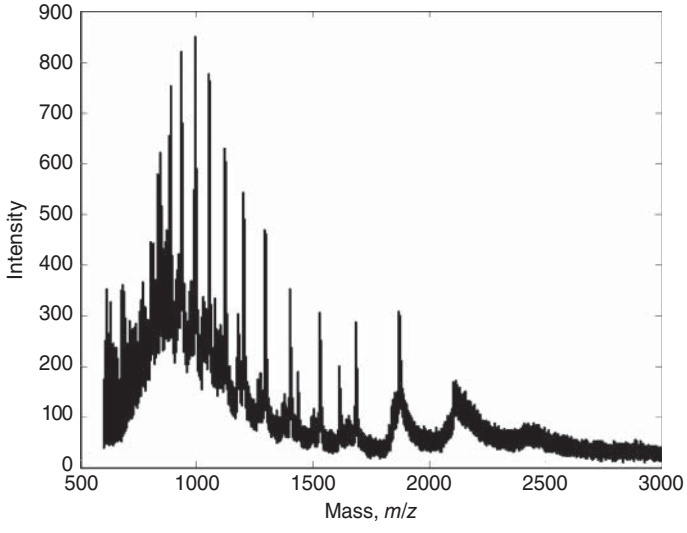


Figure 3.17 Signal from mass spectrometry of staphylococcus nuclease (SNase).

Apart from filtering and smoothing, DWT can also be applied for data compression, baseline removal, resolution enhancement, or in combination with regression, classification, and projection methods such as factor analysis. Extension to 2D provides important applications in image analysis.

Example 3.5 *Wavelet Transformation*

A mass spectrum of the protein staphylococcus nuclease (SNase) (Figure 3.17) is to be decomposed by the Daubechies-3 wavelet transform at three levels and approximation A_3 as well as details $D_1 - D_3$ are to be evaluated.

As one can see in Figure 3.18, the approximation in the third level resembles the original signal with respect to the high frequencies and filtered to some extent. The details reflect the low frequency part of the signal ranging over the whole mass-to-charge ratio, m/z, from 500 to 3000 mass units.

Alignment Methods

The time axes in chromatograms and the energy axes in spectra often change from one run to another. Statistical analysis of a set of chromatograms and spectra, however, is only possible if the features in those curves coincide at the same channel, that is, at the same column in the signal data matrix.

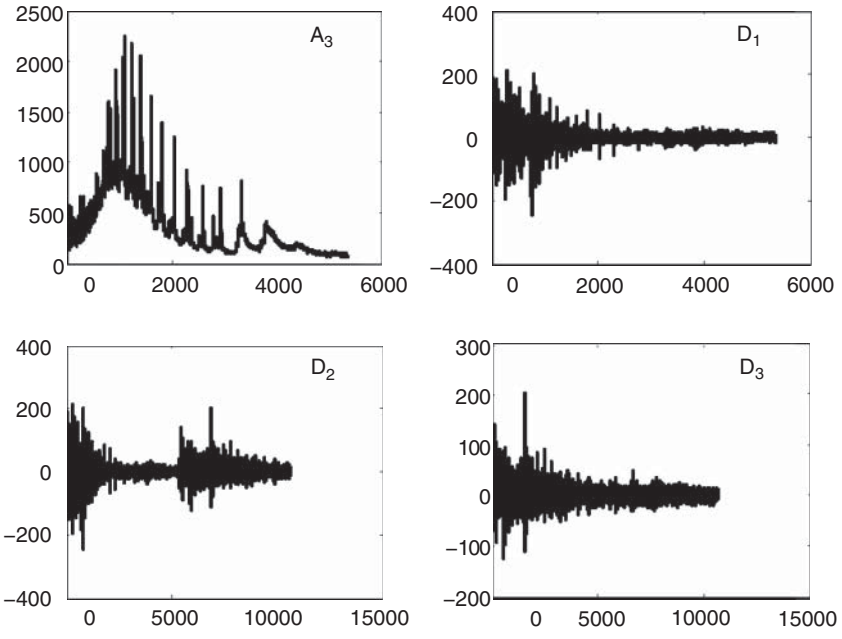


Figure 3.18 Decomposition of the signal in Figure 3.17 into approximation A_3 and details D_1 to D_3.

To match the shifted points of signal curves, dynamic time warping (DTW) is frequently applied. Warping stretches and compresses the axis in order to match the signal traces. In this kind of alignment, a reference signal curve is chosen from the set of curves. The individual points of the reference signal, $n = 1, \ldots, N$, and those of the sample signal, $m = 1, \ldots, M$, are combined in a common set of data points, $k = 1, \ldots, K$. The warping path, W, then contains as elements the indexes of the sample and reference signals according to

$$W = [n(k), m(k)] \qquad k = 1, \ldots, K \tag{3.38}$$

If each point, $c(k)$, is a pair of indexes, $n(k)$ and $m(k)$, related to the reference and sample signals, the warping path is given by

$$W = \{c(1), c(2), \ldots, c(K)\} \tag{3.39}$$

The same signal intensities might be valid for consecutive points of either the sample or the reference signal. As measure of matching, the minimum distance or the correlation coefficient can be chosen. Further constraints for the path are fixed begin and end points, monotonicity, and a limited area for shifting points.

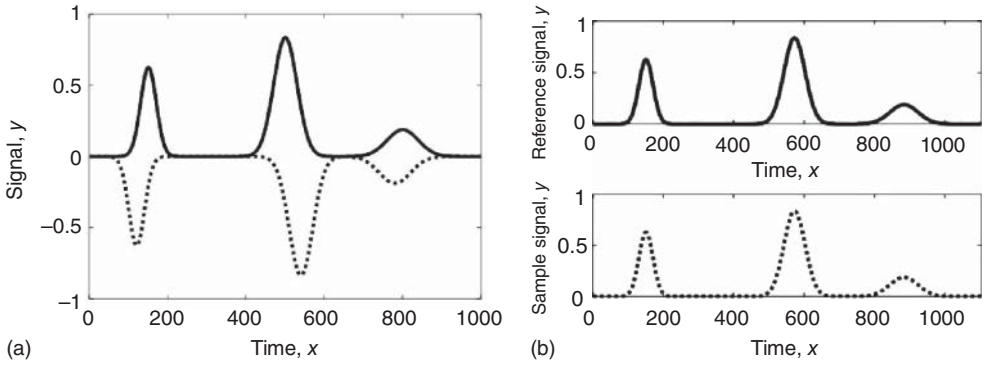


Figure 3.19 (a) Simulated chromatogram for a reference signal (solid line) and a sample signal (dotted line). (b) Alignment of reference and sample signal curves by dynamic time warping.

A simple example is depicted in Figure 3.19a, for a simulated chromatogram. In this figure, the solid line characterizes the reference signal and the dotted line the sample signal in a mirrored representation. If DTW is applied, the two curves are perfectly aligned as shown in Figure 3.19b. Notice that the original number of 1000 points has increased to more common points ($K = 1104$).

3.2 Time Series Analysis

Characterization of a set of measurements as a time series in the sense of a stochastic process is of interest in a different way, that is,

- time-dependent monitoring of pH values, metal, or ion concentrations in waters and soils;
- determination of constituents in biological fluids over time, for example, for controlling the blood glucose level;
- description of the time-dependent stability of a spectroscopic source, for example, the inductively coupled plasma in atomic emission spectroscopy;
- the judgment of a continuously or discontinuously operating analyzer.

Relationships between variables can be described by means of *correlation* and *covariance.*

The methods for smoothing, derivation, integration, or transformation as discussed in Section 3.1 can also be applied to time series. In this section, we learn about correlation methods. Correlations within a time series are described in the form of *autocorrelation* or *autocovariance.* Two different time series are characterized by means of *cross-correlation*.

Typical information to be derived from such models is about

- drift;
- noise;
- periodicity of processes, for example, seasonal components;
- forecasting (prediction) of future values on the basis of the series history.

We begin with correlations *within* a measurement series.

Autocorrelation and Autocovariance

Correlations of data within a time series can be found, if the data are plotted against their successors. Consider the time series given in Figure 3.20, where the monthly recorded sulfur concentration per liter snow, $y(t)$, is shown in dependence on time, t.

The correlations are obtained by plotting the measurement at time t, $y(t)$, against the value at time $t+1$, that is, $y(t+1)$, at time $y(t+2)$, or in general at time $y(t+\tau)$, where τ represents the so-called *time lag*. In Figure 3.21, the dependencies for the lags 0, 1, 7, and 12 are plotted.

A time lag of $\tau = 0$ represents the plot of the time series against itself. There, complete correlation is valid. With increasing time lag, that is, with increasing distance of the data points, the correlation is expected to decrease if no periodicities or drifts within the time series are present.

As a measure of the degree of correlation, the empirical autocorrelation is applied (cf. correlation coefficient according to Eq. (5.12)). For autocorrelation of a function of n data points, the *empirical autocorrelation*, $r(\tau)$, for time lag τ is defined by

$$r(\tau) = \frac{\sum_{t=1}^{n-\tau} (y_t - \bar{y})(y_{t+\tau} - \bar{y})}{\sum_{t=1}^{n} (y_t - \bar{y})^2} \tag{3.40}$$

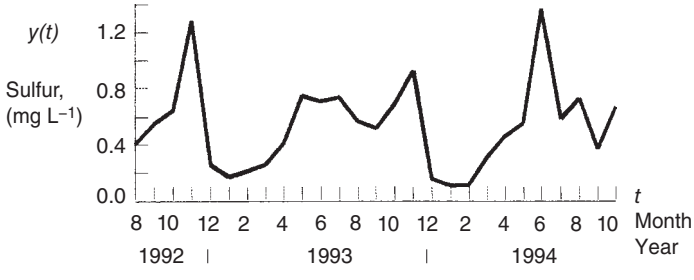


Figure 3.20 Time series for monthly recorded concentration of sulfur as sulfate.

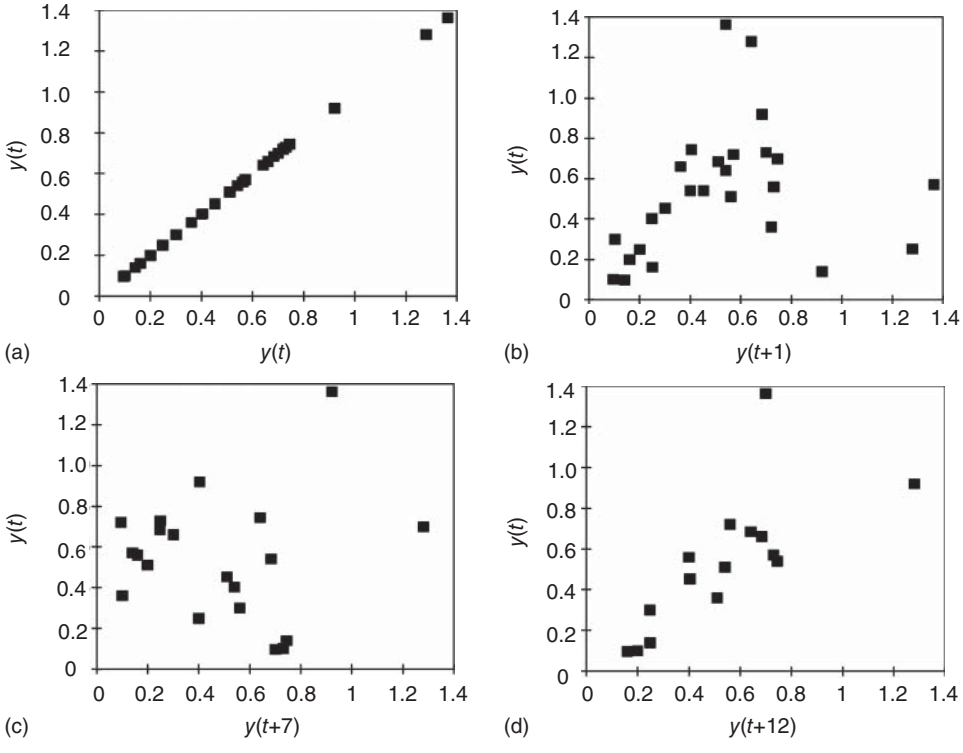


Figure 3.21 Pointwise correlations for the time series in Figure 3.20 for different time lags of $\tau = 0$ (a), 1 (b), 7 (c), and 12 (d) data points with the correlation coefficients $r(\tau) = 1.000$, 0.243, 0.0209, and 0.365, respectively.

here, $\overline{y}$ is the arithmetic mean. The expression in the denominator of Eq. (3.40) is a measure for the variance, s^2, since

$$s^2 = \frac{\sum_{t=1}^{n} (y_t - \overline{y})^2}{n - 1 - \tau} \quad \text{or} \quad \sum_{t=1}^{n} (y_t - \overline{y})^2 = (n - 1 - \tau)s^2$$

Mean and variance are considered to be constant. This assumption only holds in the case of a *stationary process.*

A stochastic process is termed *stationary*, if the signal generating process is *time invariant.* All distributions and statistical parameters of a stationary process are independent of time. A process varying with time is called *nonstationary*.

It should be mentioned that the numerator in Eq. (3.40) contains the autocovariance (cf. Eq. (5.10)). The *empirical autocovariance* with time lag τ is defined by

$$c(\tau) = \frac{1}{n} \sum_{t=1}^{n-\tau} (y_t - \overline{y})\,(y_{t+\tau} - \overline{y}) \tag{3.41}$$

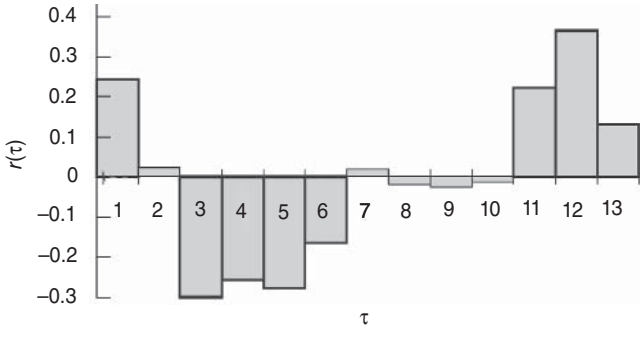


Figure 3.22 Autocorrelation function for the time series in Figure 3.20. The correlation at $\tau = 0$ (cf. Figure 3.21) is not drawn in the figure.

If one calculates the autocorrelation coefficients for the correlations in Figure 3.21, then in the case of $\tau = 0$, we obtain a value of 1 for the correlation, since

$$r(0) = \frac{\sum_{t=1}^{n} (y_t - \overline{y})(y_t - \overline{y})}{\sum_{t=1}^{n} (y_t - \overline{y})^2} = 1 \tag{3.42}$$

For increasing time lags from 1 to 7, the autocorrelations decrease to $r(1) = 0.243$ and $r(7) = 0.0209$, respectively. For $\tau = 12$, a higher value again for the empirical autocorrelation results, that is, at $r(12) = 0.365$. This points to a periodicity in the time series.

In order to evaluate all possible autocorrelations, $r(\tau)$ is plotted against the time lag τ in a correlogram, also called an *autocorrelation function*. The autocorrelation function for the time series of sulfur concentrations in snow is given in Figure 3.22. The time lags τ correspond here to monthly distances.

At the beginning of the correlogram, the correlations between the measurements decrease rapidly. At distances of 3 months or more, the sulfur values produce a negative correlation that might correspond to the four seasons (cf. Figure 3.22).

Example 3.6 *Empirical Autocorrelation*

For the time series of sulfur concentrations in snow (Figure 3.20), the empirical autocorrelation is calculated

for the time lag at $\tau = 12$. The 27 individual data are given in Table 3.3.

A value of $\overline{y} = 0.53$ is obtained for the data mean. By means of Eq. (3.27), the result for the autocorrelation is

$$r(12) = \frac{\sum_{t=1}^{27-12} (y_t - \overline{y})(y_{t+12} - \overline{y})}{\sum_{t=1}^{27} (y_t - \overline{y})^2}$$

$$= \frac{(0.40 - 0.53)(0.56 - 0.53) + (0.54 - 0.53)(0.51 - 0.53) + \cdots + (0.684 - 0.53)(0.66 - 0.53)}{(0.40 - 0.53)^2 + (0.54 - 0.53)^2 + \cdots + (0.66 - 0.53)^2}$$

$$= 0.365$$

The corresponding plot for the calculated value of 0.365 of the autocorrelation at $\tau = 12$ is illustrated in Figure 3.21.

Table 3.3 Individual values for the time series in Figure 3.19.

t	Month/Year	*y*(*t*)
1	8/92	0.400
2	9/92	0.540
3	10/92	0.640
4	11/92	1.280
5	12/92	0.250
6	1/93	0.160
7	2/93	0.200
8	3/93	0.248
9	4/93	0.404
10	5/93	0.744
11	6/93	0.700
12	7/93	0.730
13	8/93	0.560
14	9/93	0.510
15	10/93	0.684
16	11/93	0.920
17	12/93	0.140
18	1/94	0.096
19	2/94	0.100
20	3/94	0.300
21	4/94	0.452
22	5/94	0.540
23	6/94	1.364
24	7/94	0.570
25	8/94	0.720
26	9/94	0.360
27	10/94	0.660

Autocorrelation Functions for Typical Processes

For interpretation of the autocorrelation function, it is useful to know the graphs of characteristic time series.

Uncorrelated Data In the first step of data analysis, it should be checked whether the data are uncorrelated or correlated. Uncorrelated data do not show any trends in their autocorrelation function (Figure 3.23). Note how small the $r(\tau)$ values are for the empirical autocorrelations in Figure 3.23. Such data can be described by the methods discussed in Chapter 2. In other words, uncorrelated data are a prerequisite to apply the methods of descriptive statistics discussed in Chapter 2.

Correlated Data A random time series with low correlations between observations provides an autocorrelation function as shown in Figure 3.24. In the case of a stationary process of the first order, the function can be described by the following exponential model:

$$r(\tau) = e^{(-\tau/T)} \tag{3.43}$$

where T is the *time constant* of the process.

For determination of the time constant *T,* nonlinear or, after logarithmic transformation, linear regression can be applied. The time constant can also be evaluated at the specific correlation

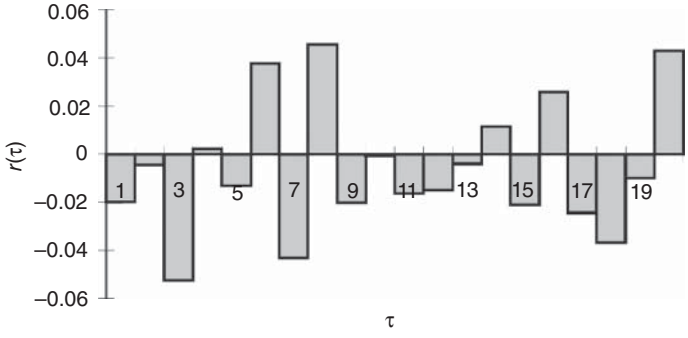


Figure 3.23 Autocorrelation function of uncorrelated data.

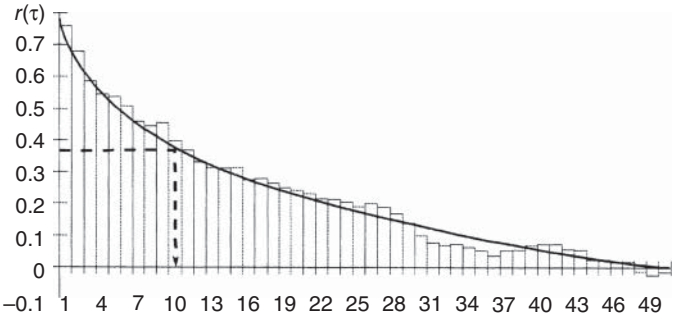


Figure 3.24 Autocorrelation function for weakly correlated data according to a process of first order. The time series is based on glucose determinations in urine over time.

coefficient $r(T) = 0.37$. At this point, $\tau = T$, and therefore,

$$r(T) = e^{(-1)} = 0.37 \tag{3.44}$$

For the data in Figure 3.24, a time constant of $T = 10$ results when calculated by Eq. (3.44).

Random Processes with Drift and Periodicities Theoretically, fluctuation about zero is expected for a random process after a decrease of $r(t)$. In the case of a *drifting* process, the autocorrelation function remains at a positive or negative correlation level (cf. Figure 3.25).

For correction of drifts, the same methods as introduced for background correction can be applied (cf. Section 3.1).

Periodic processes show characteristic dependencies as demonstrated in the example on the seasonal variations of sulfur concentrations in Figure 3.22. Periodicities or fluctuations can be recognized and quantified from autocorrelation functions much better than from the time series.

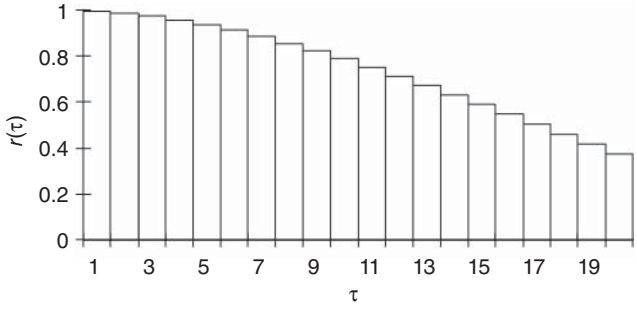


Figure 3.25 Autocorrelation function for a time series with drift, as found for measurements with a chemical sensor.

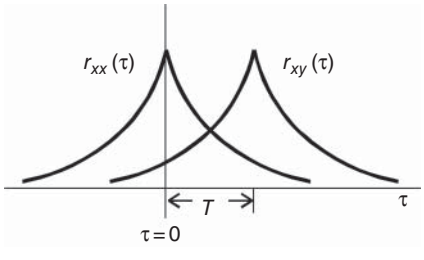


Figure 3.26 Schematic demonstration of cross-correlation between an input signal *x* and output signal *y*. The time lag can be derived from the shift in the correlation maximum.

Cross-Correlation

To describe the correlation of two different time series, $y(t)$ and $x(t)$, the *empirical cross-correlation* at time lag τ is calculated by the following equation:

$$r_{xy}(\tau) = \frac{\sum_{t=1}^{n-|\tau|} x_t y_{t+\tau}}{\sqrt{\sum_{t=1}^{n} x_t^2 \sum_{t=1}^{n} y_t^2}} \tag{3.45}$$

Cross-correlation can be used to investigate the input–output behavior of an analytical system to compare a theoretical band shape with an observed one. The greatest correspondence between the two time series is observed in the correlogram at the position where a maximum for the empirical correlation is found (Figure 3.26).

Autoregression

A close relationship exists between correlation of data and their regression on each other (cf. Section 6.1). It is therefore possible

to model the successive data of a time series by a linear regression model in order to predict future values.

For *autoregression* of a time series, we obtain

$$y(t+\tau) - \overline{y} = r(\tau)\ [y(t) - \overline{y}] + e(t+\tau) \tag{3.46}$$

The measurement at time t plus the time lag τ is predictable on the basis of the autocorrelation coefficient, $r(\tau)$, and the y value at time t. Here, e represents the random error. Note that this is a very simple model. Good predictions can only be expected for time series that really obey this simple autoregressive model.

Correlation methods are not restricted to the characterization of time-dependent measurements. They can also be used if correlations between spectra, chromatograms, or other correlated analytical measurement series are to be investigated.

General Reading

1. Bowerman, B.L. (1993) *Forecasting and Time Series*, Duxbury, Belmont, CA.
2. de Boor, C. (1978) *A Practical Guide to Splines*, Springer, Heidelberg.
3. Chau, F.-T., Liang, Y.-Z., Gao, J., and Shao, X.-G. (2004) *Chemometrics: From Basics to Wavelet Transform (Chemical Analysis: A Series of Monographs on Analytical Chemistry and Its Applications)*, John Wiley & Sons, Ltd., Chichester.
4. Kalman, R.E. (1960) *J. Basic Eng.*, **82**, 35.
5. Rutan, S.C. (1989) *Chemom. Intell. Lab. Syst.*, **6**, 191.
6. Savitzky, A. and Golay, M.J.E. (1964) *Anal. Chem.*, **36**, 1627.
7. Vaseghi, S.V. (1996) *Advanced Signal Processing and Digital Noise Reduction*, Wiley-Teubner, Chichester – Leipzig.
8. Zupan, J. (1989) *Algorithms for Chemists*, John Wiley & Sons, Ltd., Chichester.
9. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. **2**, Elsevier, Amsterdam.

Questions and Problems

1. Describe the useful and interfering frequencies in an analytical signal.
2. Explain the effect of filter width on the noise and structure of a signal trace, such as a spectrum or a chromatogram.
3. Calculate the fourth smoothed value, y_4^*, for the signal data given in Example 3.1 by means of a Savitzky–Golay filter.
4. The Kalman filter is especially useful for real-time filtering. Why?
5. What are the benefits of signal derivation and which derivatives should by applied in what situation?
6. What are the base functions of FT, HT, and WT? Describe the situations in which each would be the most suitable for use.
7. What is the difference between a polynomial and a spline function?
8. What information can be deduced from autocorrelation functions?

4
Optimization and Experimental Design

Learning Objectives

- To provide an introductory course into systematic optimization methods in analytical chemistry
- To select the most important factors that influence a given analytical problem based on statistical approaches of experimental design as well as on evaluating the factor effects and their interactions by means of statistical tests
- To discuss the design of experiments for modeling the relationship between factors and to apply response surface methods (RSMs) for locating the optimum
- To search for the optimum by sequential methods, that is, by means of the simplex method by Nelder and Mead.

Effective experimentation and the development of optimized methods are fundamental aims of any experimenter. Typical goals of the analyst in this connection are as follows:

- Investigation of factors on an analytical signal for judging the robustness of an analytical method or to test for interferences.
- Optimization of the performance of analytical methods with respect to quality criteria, such as precision, trueness, sensitivity, detection power, or signal-to-noise ratio.
- Development of an optimal composition of a digesting agent or of a chemical sensitive layer for the development of sensors.

Typical *factors* in analytical chemistry are the pH value, reagent concentration, temperature, flow rate, solvent, elution strength, mixture components, irradiation, atomization time, or sputtering rate. Typical *responses* are the analytical figures of merit and objective functions that consist of combinations of different quality criteria. Objective functions and factors are considered in Section 4.2 in detail.

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

A *systematic* optimization is always preferable to a trial-and-error approach.

In principle, two approaches in experimental optimization can be distinguished. First, selection and testing of the most important factors, as well as their subsequent optimization, are based on the subjective experience of the experimenter or analytical expert. The success will, then, be dictated by the knowledge level of the domain expert. If the know-how is low, then the workload might become large. In addition, as we will learn in the following, a nonsystematic approach does not really guarantee finding the optimum.

Second, the investigation of the factors and their optimization can be carried out in a systematic way. This systematic method is the subject of this chapter. The most successful experimenter will be the one able to support his expert-based policy by systematic investigations.

4.1 Systematic Optimization

Systematic optimizations are carried out in the following sequence:

- Choice of an objective function
- Selection of the most important factors
- Optimization.

Very often, the optimization criterion is simply an analytical signal or the analysis time. In more complex situations, however, objective functions that are composed of several criteria, such as selectivity, sensitivity, and precision, must be considered. The combination of objective criteria to give a single function is therefore an important topic in analytical chemistry.

Potential factors that affect a given objective function are best selected by the domain expert in a particular analytical field. The test for significance of the factors' influence should be performed on the basis of a simple experimental design, a screening design, by means of statistical tests. Factors should not be kept or eliminated solely for subjective reasons.

To find the most suitable factor combinations, we can distinguish between simultaneous and sequential optimization approaches.

Simultaneous Methods

With *simultaneous* strategies, the relationship between responses and factors is studied by running an experimental design,

constructing a mathematical model, and investigating the relationship by the so-called RSMs. Very often, RSMs are aimed at judging this relationship graphically and the consequences are drawn from the plots. If the optimal point is desired, it can be found by calculating the partial derivatives with respect to the individual factors or by applying a grid search over the entire response surface.

Sequential Methods

Sequential strategies of optimization are based on an initial design of an experiment followed by a sequence of further measurements in the direction of the steepest ascent or descent. That is, no quantitative relationship between factors and responses is evaluated, but the response surface is searched along an optimal (invisible) path. The two strategies are exemplified in Figure 4.1.

Simultaneous methods are based on a mathematical model for the response area, whereas sequential approaches represent *search methods.*

4.2 Objective Functions and Factors

A prerequisite for any optimization is the definition of one or several objective criteria (figures of merit). For computer-aided or automatic optimizations, the objective functions need to

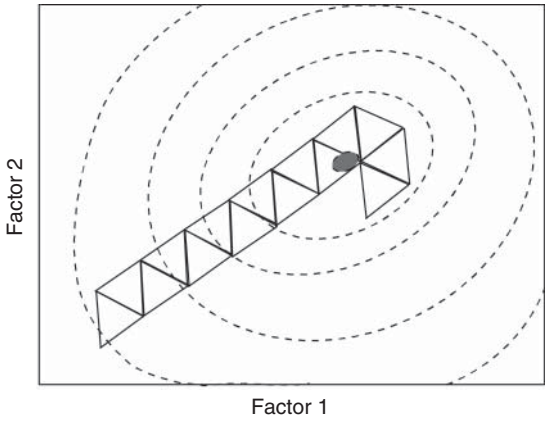


Figure 4.1 Response versus factors plot: In the case of response surface methods (RSMs), the response is described by a mathematical model (*dotted contour lines* of the response surface). By using search methods, the response is measured along a search path, here along a simplex path (cf. "Analytical Performance Characteristics" Section).

be represented in a computer-readable format. In general, the following types of objective criteria can be distinguished:

- Continuous or discrete continuous quantities, for example, yield, time demand, analytical figures of merit, or deviations between model and experiment
- Discrete (nominal scaled) quantities, for example, the number of crystallizations or extractions
- Ordinal scaled values, for example, sensory data, such as different degrees of sweetness of a raspberry jam, in the sense of a ranking order.

The optimum sought of an objective function is either the minimum, for example, minimum time demand, or the maximum, for example, the yield of a chemical reaction.

In analytics, the analytical performance characteristics constitute ever-important objective criteria that must often be considered in combination.

Analytical Performance Characteristics

Calibration Function

In connection with the calibration function performance, several characteristics are combined, such as the sensitivity of an analytical method and its working range.

The *sensitivity* corresponds to the slope of the calibration curve. The *calibration function* for the analytical signal, *y,* in dependence on the concentration (or mass), *x,* is

$$y = b_0 + b_1 x \tag{4.1}$$

The sensitivity, b_1, for this calibration function is then defined by (cf. Figure 4.2):

$$b_1 = \frac{\Delta y}{\Delta x} \tag{4.2}$$

The intercept b_0 represents an uncorrected blank or background value. If the measurement is carried out versus the blank, a model without the term b_0 would be valid.

In atomic absorption spectrometry, the use of inverse sensitivity is commonly defined by the mass or concentration per 1% absorption (0.00436 absorbance units). The sensitivity can be given as a constant value for the whole concentration range only in the case of a straight-line calibration curve. In the case of curved calibration graphs, the sensitivity varies with concentration.

In the latter case, optimization is often performed with the aim of enhancing the linear range.

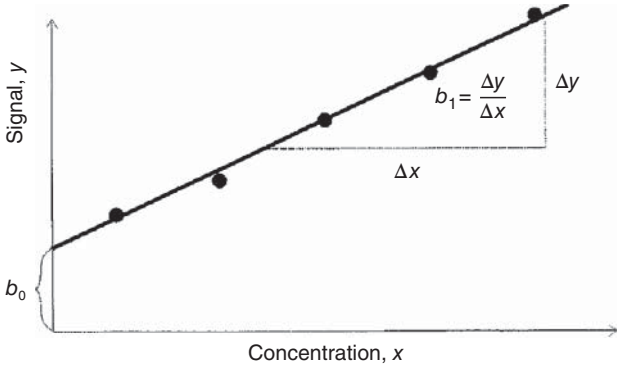


Figure 4.2 Sensitivity for a straight-line calibration as functional dependence of signal on concentration.

The *dynamic range* corresponds to the valid range of the functional relationship between the signal and the concentration or mass. The *analytical* or *working range* denotes the interval between the lowest and highest concentrations, for which accurate measurements are feasible for evaluation of random and systematic errors. Outside this interval, the measurements are considered uncertain.

In the case of *curved calibration graphs*, the sensitivity is reported together with the concentration considered.

Detection Limit and Limit of Determination

The detection limit describes the concentration that can be determined with a given analytical method. For evaluation of the *detection limit*, the signal at the detection limit, y_{DL}, is calculated from the blank mean and the standard deviation as follows:

$$y_{DL} = y_B + 3s_B \tag{4.3}$$

The factor 3 provides sufficient statistical certainty to account for the errors due to transformation from the signal to the concentration domain, for the necessary assumptions on the distribution of the blank values, and for the limited number of measurements, which only allow an estimate of the standard deviation of the blank value (s_B).

After reshaping of the calibration function in Eq. (4.1), the concentration at the detection limit, x_{DL}, is evaluated as

$$x_{DL} = \frac{y_{DL} - b_0}{b_1} \tag{4.4}$$

To characterize a boundary at which a quantitative analysis is still feasible, the limit of determination is used. The *limit of determination* is defined as the lowest analyte concentration that can be determined with an acceptable accuracy.

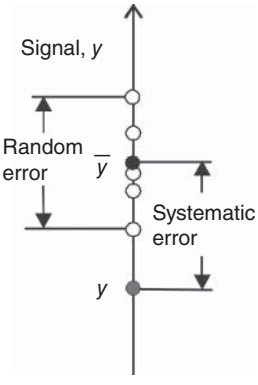


Figure 4.3 Random and systematic errors for measurements in the signal domain.

Accuracy of Analyses: Precision and Trueness

The accuracy of an analysis is determined by its precision and trueness. Precision corresponds to the fraction of random errors and trueness to that of systematic errors (bias). In Figure 4.3, the two error types for measurements of a signal, y, are illustrated.

In the concentration domain, we obtain

$$e = \underbrace{(x - \bar{x})}_{\text{random error}} + \underbrace{(\bar{x} - x_w)}_{\text{bias}} \tag{4.5}$$

Accuracy is the term placed over *precision* and *trueness.*

where e is the error for the accuracy of analysis, x the determined concentration, $\bar{x}$ the concentration mean, and x_w the true concentration.

Precision characterizes the repeatability of measurements. For n independent random measurements, it can be described by using the standard deviation as the dispersion measure. In the concentration domain, x, is calculated according to Eq. (2.4) by

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}} \tag{2.4}$$

The *trueness* for characterization of systematic errors is specified by the rate of recovery. This is the ratio of the observed mean and the true value given as a percentage by

$$\text{RR}(\%) = \frac{\bar{x}}{x} 100 \tag{4.6}$$

Specificity and Selectivity

The s*electivity* of an analytical method is a measure of the degree to which the determination of an analyte is interfered with by accompanying analytes or matrix components. A *fully selective* analytical method enables the analytes to be selected for

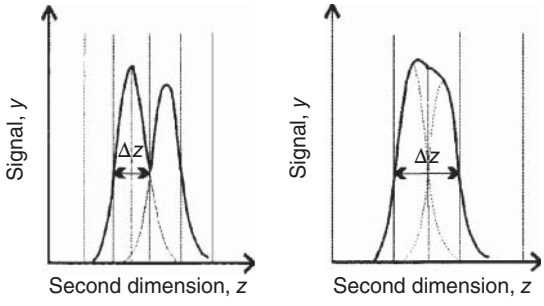


Figure 4.4 Illustration of the analytical resolution for differently separated peaks or bands.

determination without interference. One says that the method is *specific* with respect to the individual analytes.

A prerequisite for an undisturbed quantitative analysis is the existence of interference-free signals for each component to be determined. Signals related to several components are only available for the so-called two-dimensional analytical methods. The signals are monitored in dependence on a second dimension, z, such as wavelength, time, or electrode potential (cf. Figure 4.4). As a measure for the estimation of selectivity, the analytical resolution, N, is used. The analytical resolution is defined as the ratio of the signal location at the z-axis and the signal full width at half maximum, Δz:

$$N = \frac{z}{\Delta z} \tag{4.7}$$

The larger the resolution, the higher is the selectivity of the method.

If an analytical method operates fully selectively, optimization of its selectivity would be superfluous. A baseline-separated chromatogram of all components of a sample is one example of a fully selective analytical method.

For characterization of *limited selectivity,* the notions of interference, cross-sensitivity, and overlap are common.

An unselective or partially selective method is characterized by overlapping analyte signals. The aim of optimization of such methods is improvement of their selectivity, if possible, up to the development of a fully selective method. Often, however, signal overlap can only be decreased to some extent and methods of multicomponent analysis have to be used to correct for the remaining interferences computationally (cf. "Applications for Multicomponent Analysis" Section).

To characterize incomplete selectivity, measures are applied that consider only two adjacent signals related to the corresponding two components. Table 4.1 provides selectivity measures from different areas of analytics. By appropriate aggregation of

Table 4.1 Selectivity measures in analytics.

Selectivity measure	Analytical principle	Formula
Selectivity coefficient, K_{ij}^{pot}	Potentiometry	$K_{ij}^{\text{pot}} = \frac{a_i}{a_j}$
Separation factor, α	Chromatography	$\alpha = \frac{k_1'}{k_2'}$
Resolution, R_S	Chromatography	$R_S = \frac{\sqrt{N}}{4}(\alpha - 1)\frac{k'}{1+k'}$

a_i, a_j activities of ions i or j; k_i' capacity factor; N plate number.

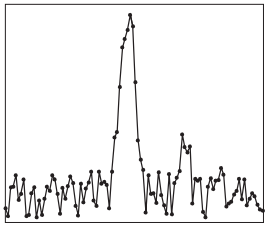
the selectivity measures, multicomponent systems can also be described, for example, in chromatography (cf. Table 4.2).

More general selectivity measures can be derived if the principles of multicomponent analysis are explored. A consequent application of the laws of error propagation enables selectivity criteria to be derived for the general case of multicomponent analysis, which can even cope with different concentration ratios of analytes in the sample.

Since the principles of multicomponent analysis are only introduced in Section 6.2, the corresponding selectivity criteria will be discussed there.

Other figures of merit are the *signal-to-noise ratio* and the *signal-to-background* ratio. A measure for the signal-to-noise (*S/N*) ratio is the quotient of the signal means, $\overline{y}$, and the standard deviation of the signal noise, s_y:

$$\frac{S}{N} = \frac{\overline{y}}{s_y}$$



Time, Cost, and Risk

Minimization of the time demand or cost of analyses might also constitute an objective function. To optimize an entire analytical procedure, methods of operational research might be needed in addition to the systematic approaches considered in this section. This is especially important in cases where risk assessment is required in connection with the analytical procedure.

Accounting for Several Performance Characteristics

In practice, it is very often important to optimize several objective criteria. The simultaneous optimization of an analytical method with respect to selectivity and time demand is a typical example of this situation.

Aggregation of Performance Characteristics

Multicriteria decision making is feasible in two ways. First, the different performance characteristics are aggregated to an objective function, most easily by a weighted sum. The objective function, Z, is obtained from the p individual objective criteria, z_i, by

$$Z = w_1 z_1 + w_2 z_2 + \cdots + w_p z_p = \sum_{i=1}^{p} w_i z_i \tag{4.8}$$

Table 4.2 Aggregation of performance characteristics to an objective function (explanation in text).

Objective function	Analytical principle	Formula
Chromatographic response function, CRF	Chromatography	$\mathrm{CRF} = \frac{1}{t}\prod_{i=1}^{m-1}\frac{f_i}{g_i + 2N_i}$
Leary criterion, Z_{AES}	Atomic emission spectrometry	$Z_{\mathrm{AES}} = \frac{p}{\sum_{i=1}^{p}\left(\frac{I_i}{I_i^{\mathrm{B}}}\right)^{-1}}$
Signal-to-background ratio, Z_{XRF}	X-ray fluorescence analysis	$Z_{\mathrm{XRF}} = \frac{1}{t}\sum_{i=1}^{p}\frac{I_i - I_i^{\mathrm{B}}}{I_i^{\mathrm{B}}}$

where w_i is the weight of objective criterion *i.*

The weights are adjusted in such a way that they reflect the real influence of the performance characteristics on the total result. This is not usually easy to do. In addition, with this method, the type of aggregation has to be decided in advance and optimization is carried out with respect to a single point.

Different aggregations of objective criteria have been developed for particular analytical methods. Table 4.2 gives examples of objective functions for chromatography and spectroscopy. The objective function for chromatography, the chromatographic response function (CRF) accounts for all m peaks of the chromatogram, the time t for elution of the last peak, the noise, N_i, at the measurement point of peak i, and the selectivity of peak separation based on Kaiser's measure for peak separation f/g (see Figure 4.5). For optimal separations, the CRF is maximized.

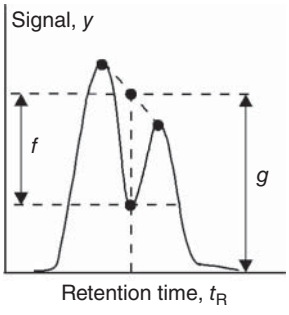


Figure 4.5 Peak separation according to Kaiser.

For judging multielement determination by means of atomic emission spectrometry (AES) or X-ray fluorescence analysis (XRF), the evaluation of the signal-to-background ratio, I to I^{B}, for all p signals is proposed.

Advantages and disadvantages of those objective functions are to be tested in the actual application of the particular method. A general disadvantage for these methods is that the weighting of the objective criteria is done by a fixed weight.

A more flexible evaluation is feasible by using cost or utility functions. They can be constructed on the basis of conventional mathematical functions or by means of fuzzy functions (Section 8.3).

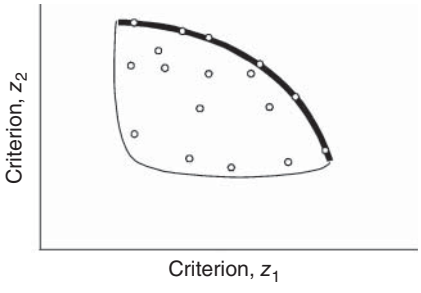


Figure 4.6 Goal region for the simultaneous investigation of two objective criteria, for example, selectivity and analysis time. The *bold line* characterizes Pareto optimal points.

The preliminary decision on the aggregation of the criteria does not allow for compromise solutions, which are possible if methods of polyoptimization are applied.

Polyoptimization

To find compromise solutions, the entire region is investigated without aggregating the individual criteria in advance.

A prerequisite is here being feasible to describe the relationship between the objective criteria and the factors by a mathematical model. The objective criteria can then be computed for all the factor combinations, and if not more than three criteria are to be considered, they can be plotted or they can be computationally investigated. In that way, the compromise set of all Pareto optimal points can be found (Figure 4.6). Pareto optimal points are those points or factor combinations for which a change of one of the objective criteria would result in worsening of at least one other criterion.

As can be seen in Figure 4.6, the objective criteria cannot assume any arbitrary value. Depending on the experimental constraints for the factors, the objective criteria lie in a bounded region. On the basis of the mathematical model, the objective region can be investigated by calculation of the criteria at as many points as needed.

4.3 Experimental Design and Response Surface Methods

Fundamentals

Ceteris Paribus Principle

Optimization of an analytical problem or of an analytical procedure has to be carried out by studying a limited number of

factors. Very often, it is easy to select the most important factors from prior and common knowledge about a given problem, for example, in developing a new spectrophotometric method, the factors pH and reagent concentration would have to be studied, or in HPLC, important factors are the constituents of the mobile phase and their actual concentrations.

In a *screening experiment*, the factor range is evaluated for the subsequent studies.

Sometimes, the effect of a factor can only be presumed and its effect would have to be assured by suitable screening experiments. Because of the complexity of most of the analytical problems, there will be additional factors that are either unknown or cannot be controlled by the experimenter. Uncontrolled factors might be the impurity of reagents, intoxication of an electrode surface, the instability of a plasma source, and the changing quality of a laboratory assistant's work.

Since the study of all potential factors is usually prohibitive, the effect of selected factors will be investigated and the remaining factors should be kept as constant as possible. This general principle is known as the *ceteris paribus principle*.

Replication

Replication of measurements at a given factor combination is necessary for estimation of the experimental error. Furthermore, the error can be decreased with repetitive measurements by averaging, that is, by a factor of $1/\sqrt{n}$ if n repetitive measurements have been carried out.

Randomization

Randomization means running the experiments in a random order. Randomized experiments are obligatory if systematic errors (bias) cannot be avoided and should be detected. Imagine the construction of a calibration graph. If the concentrations are measured in ascending order, it would not be possible to detect a systematic error such as a positive signal drift correlated with time. All experimental observations might lie on a straight line, but the measured slope of the calibration graph is shifted systematically from the true slope (Figure 4.7a). On the other hand, if the concentrations are measured in a random sequence, then you will notice that something is wrong because the variation of observations will be exceptionally high (Figure 4.7b).

In the present example, we have examined only one factor: the component concentration. In the case of studying several factors, randomization is similarly necessary. In multifactor experiments, the experimental design must be run in a randomized order, as we will see.

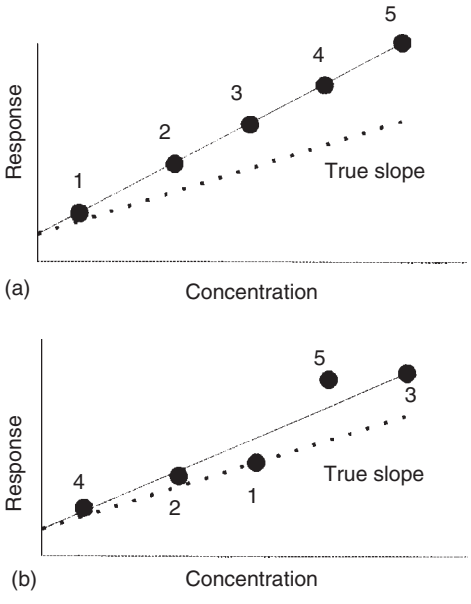


Figure 4.7 Experimental observations for constructing a calibration graph in systematic (a) and randomized order (b).

One of the basic prerequisites of all statistical tests is the existence of random and independent data. This assumption is also valid for statistical tests in connection with experimental designs, so that randomized experimentation is obligatory in this context. Carrying out the experiments in a random sequence will be one of the suppositions for being able to measure independent, uncorrelated, and usually normally distributed data.

Random sequences should be read from random number tables or, nowadays, these are read from a random number generator. This guarantees the genuine randomness of the sequence instead of subjective selections.

Blocking

Uncontrolled factors lead to higher experimental errors, which will also decrease the sensitivity of the experiments with respect to the factors to be studied. Therefore, the experiments should be designed such that uncontrolled factors can be detected and, in a further step, can be kept constant or eliminated.

There are two categories of uncontrolled factors. Uncontrolled influences might arise from either *unknown factors* or *known factors that cannot be controlled.* The eventual impurity of a reagent is an example of an unknown factor. The changing quality

of a' laboratory assistant's work is an agreed factor that is difficult to control. Completely unknown factors cannot be accounted for at all.

Known or presumably known factors can be detected by blocking the experiments. The idea is to run the experiments in blocks that show a minimum experimental variance within one block. For example, if a systematic investigation requires 12 experiments and you could run only 4 experiments a day, the experiments should be arranged in 3 blocks with 4 experiments each day. Day-to-day effects could then be detected by considering the block effects with an adequate mathematical model as given as follows.

Of course, the experiments within a block should be run at random. Randomized experimentation with respect to some factors and blocking of the experiments with respect to some other factors exclude each other. Therefore, in practice, a compromise between randomized runs and blocked experiments has to be found.

Factorial Experiments

Factorial experiments are based on varying all factors simultaneously at a limited number of factor levels. This kind of experimentation is especially important in the beginning of an experimental study, where the most influential factors, their ranges of influence, and factor interactions are not yet known. Factorial experiments allow experiments to take place in the whole range of the factors' space. They reveal high precision at a minimum experimental effort, and they enable factor interactions to be detected, such as the dependence of enzyme activity on both pH value and coenzyme concentration.

Confounding

Confounding of parameter estimations for different factors occurs if the factor combinations are highly correlated and, therefore, no difference between the factor effects can be detected. Confounding depends highly on the concrete experimental design. If, for example, the levels of two factors are changed in a constant ratio, it would not be possible to distinguish between the effects of those two factors.

Symmetry

Factorial experiments should be partitioned in the whole factor space in a balanced manner. The same is true for replications in the

experimental space. One reason for performing symmetric experiments is the avoidance of confounded factor effects. In addition, symmetric experiments might simplify data evaluation.

Two-Level Designs: Screening Designs

Full Factorial Designs

Designs on the basis of two levels for each factor are called *screening designs*. The most general design is a *full factorial design* at two levels. These designs are described as 2^k-designs where the base 2 stands for the number of factor levels and k expresses the number of factors.

Example 4.1 *2^3 Design*

As an example, a two-level three-factorial design, 2^3, is given in Table 4.3 and Figure 4.8. The factor levels are scaled here to −1 for the lower level and +1 for the higher level. Other coding schemes are also used, for example, 0 and 1, − and +, or A and B for the low and high levels, respectively. One advantage of working with scaled factor levels is the feasibility of applying the same design to several investigations. Another advantage can be seen if we model the relationship between responses and factors quantitatively. With coded levels, the size of the parameters will become comparable. Of course, the original variables can also be used in a study.

Table 4.3 Full factorial design at two levels, 2^3 design.

Experiment	Factors			Response
	x_1	x_2	x_3	
1	−1	−1	−1	y_1
2*	+1	−1	−1	y_2
3	+1	+1	−1	y_3
4*	−1	+1	−1	y_4
5*	−1	−1	+1	y_5
6	+1	−1	+1	y_6
7*	+1	+1	+1	y_7
8	−1	+1	+1	y_8

The *star*-labeled experiments represent the half-fraction factorial design of Figure 4.8.

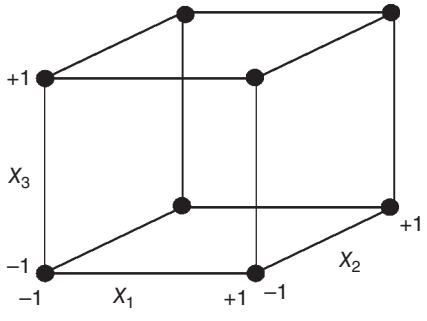


Figure 4.8 Full factorial design at two levels, 2^3 design. x_1, x_2, and x_3 represent the three factors.

Fractional Factorial Designs

As long as the number of factors is small, full factorial designs can easily be run. At higher factor numbers, however, the number of experiments will increase dramatically. For example, for the study of 7 factors in a 2^7 design, 128 experiments in total will be necessary. At this point, we have to discuss the objectives of running a factorial experimental design. One reason is the estimation of factor effects. As we will see in the section on linear models (Section 6.2), it will not be necessary to evaluate the effects of 7 factors by running 128 experiments. Another aim is to model the responses in dependence on the factors. For modeling dependences of responses on two factor levels, we would use a polynomial of first order. So in our case, we would have to estimate seven parameters linked to the effects of the seven factors and perhaps an additional parameter that models the intercept at the ordinate axis. For computing this statistical model, we deduce $128 - 8 = 120$ degrees of freedom, which are obviously too many to test for the adequacy of a simple first-order polynomial model.

The number of experiments can be reduced if *fractional factorial designs* are used. For fractional factorial designs, the number of experiments is reduced by a number p according to a 2^{k-p} design. In the case of $p = 1$, the so-called half-fraction designs result.

Example 4.2 2^{3-1} *Design*

For the example of three factors at two levels (cf. Figure 4.8), the half-fraction design consists of $2^{3-1} = 4$ experiments as given in Figure 4.9. In Table 4.4, the points of this design are

labeled by an asterisk. The experiments can be run either at the given points or at the complementary corner points.

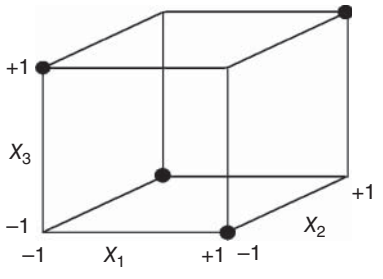


Figure 4.9 Fractional factorial design at two levels, 2^{3-1} half-fraction design. x_1, x_2, and x_3 represent the three factors.

Table 4.4 Factorial design for four factors at two levels.

Run	Factor			
	x_1	x_2	x_3	x_4
1	−1	−1	−1	−1
2*	+1	−1	−1	+1
3	−1	+1	−1	+1
4*	+1	+1	−1	−1
5*	−1	−1	+1	+1
6	+1	−1	+1	−1
7*	−1	+1	+1	−1
8	+1	+1	+1	+1

2^{4-1} – half-cell design.

Apart from saturated designs such as those in Tables 4.4 and 4.5, numerous other fractional factorial designs can be used if the effect of many factors is to be studied. Concrete designs can be taken from tables (cf. Table A.9) and can be generated with most of the software packages given in the Appendix. Special designs for estimating only the main effects have been tabulated by Plackett and Burman. As an example in Table 4.6, their fractional factorial design for 11 factors at two levels is represented.

A special case of a two-level factorial design is the *Latin square design*, which was introduced very early on to eliminate more than one blocking variable. A Latin square design for two factors is given in Table 4.7 along with the representation as a fractional factorial design.

Table 4.5 Factorial design for five factors at two levels.

Run	Factor				
	x_1	x_2	x_3	x_4	x_5
1	−1	−1	−1	−1	+1
2	+1	−1	−1	−1	−1
3	−1	+1	−1	−1	−1
4	+1	+1	−1	−1	+1
5	−1	−1	+1	−1	−1
6	+1	−1	+1	−1	+1
7	−1	+1	+1	−1	+1
8	+1	+1	+1	−1	−1
9	−1	−1	−1	+1	−1
10	+1	−1	−1	+1	+1
11	−1	+1	−1	+1	+1
12	+1	+1	−1	+1	−1
13	−1	−1	+1	+1	+1
14	+1	−1	+1	+1	−1
15	−1	+1	+1	+1	−1
16	+1	+1	+1	+1	+1

2^{5-1} – half-cell design.

Table 4.6 Plackett and Burman fractional factorial design for estimating the main effects of 11 factors at 2 levels.

Run	Factors											Response
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	
1	+	+	−	+	+	+	−	−	−	+	−	y_1
2	−	+	+	−	+	+	+	−	−	−	+	y_2
3	+	−	+	+	−	+	+	+	−	−	−	y_3
4	−	+	−	+	+	−	+	+	+	−	−	y_4
5	−	−	+	−	+	+	−	+	+	+	−	y_5
6	−	−	−	+	−	+	+	−	+	+	+	y_6
7	+	−	−	−	+	−	+	+	−	+	+	y_7
8	+	+	−	−	−	+	−	+	+	−	+	y_8
9	+	+	+	−	−	−	+	−	+	+	−	y_9
10	−	+	+	+	−	−	−	+	−	+	+	y_{10}
11	+	−	+	+	+	−	−	−	+	−	+	y_{11}
12	−	−	−	−	−	−	−	−	−	−	−	y_{12}

Table 4.7 A 2×2 Latin square design in different representations.

Conventional representation		
Blocking variable 1	**Blocking variable 2**	
	Low	**High**
Low	A	B
High	B	A

Factorial representation		
Blocking variable 1	**Blocking variable 2**	**Blocking variable 3**
Low	Low	A
Low	High	B
High	Low	B
High	High	A

Example 4.3 *Latin Square*

Four formulations of a drug are to be studied with respect to bioequivalence by treating four subjects at four periods. A 4×4 Latin square is constructed as given in Table 4.8. The formulations are coded by A, B, C, and D.

Table 4.8 4×4 Latin square.

Subject no.	**Period**			
1	A	B	D	C
2	D	C	A	B
3	B	D	C	A
4	C	A	B	D

Note the balance of the Latin square. All four drugs are given at each period to each person.

Estimation of Factor Effects

Screening designs are mainly used to estimate the effects of factors in an analytical investigation on a statistical basis. To understand the test procedure, we will consider an example from kinetic–enzymatic determinations.

Example 4.4 *Factor Effects Estimation*

The determination of the enzyme ceruloplasmin based on spectrophotometric measurements of the initial rate of *p*-phenylenediamine (PPD) oxidation is investigated at a constant enzyme concentration of 13.6 mg/l. In the first step, a screening 2^3 design is used to study the effects of the factors pH value, temperature, and substrate concentration PPD.

The factor levels are given in Table 4.9 along with the experimental design and the measured initial rates. Based on the 2^3 design, both *main* factor effects and *interactions* can be studied. The levels for factor interactions are calculated as products of the actual factor level combinations.

Table 4.9 A 2^3 screening design and factor levels for estimation of the factors pH value, temperature (*T*), and *p*-phenylenediamine concentration (PPD).

Factor	Level	
	−1	+1
T (°C)	35	40
PPD (mM)	0.5	27.3
pH	4.8	6.4

Run	Coded factor levels						y (min^{-1})
	Main effects			Interaction effects			
	T	PPD	pH	*T* × PPD	*T* × pH	PPD × pH	
1	+1	−1	−1	−1	−1	+1	6.69
2	+1	+1	−1	+1	+1	−1	11.71
3	+1	+1	+1	+1	−1	+1	14.79
4	+1	−1	+1	−1	+1	−1	8.05
5	−1	−1	−1	+1	−1	+1	6.33
6	−1	+1	−1	−1	+1	−1	11.11
7	−1	+1	+1	−1	−1	+1	14.08
8	−1	−1	+1	+1	+1	−1	7.59

The factor effects are calculated as the absolute difference, $|D|$, between the responses of a factor at high and low levels. For example, for the factor, x_1, of the full factorial 2^3 design in Table 4.4, we obtain

$$D_{x_1} = \frac{y_2 + y_4 + y_6 + y_8}{4} - \frac{y_1 + y_3 + y_5 + y_7}{4} \tag{4.9}$$

These differences are then tested against the experimental error expressed by the standard deviation s multiplied by the Student's t value:

$$|D| \geq t(P,f)\,s \tag{4.10}$$

$$\begin{aligned}
D_T &= \frac{y_1+y_2+y_3+y_4}{4} - \frac{y_5+y_6+y_7+y_8}{4} \\
&= \frac{6.69+11.71+14.79+8.05}{4} \\
&\quad - \frac{6.33+11.11+14.08+7.59}{4} = 0.53 \\
D_{\mathrm{PPD}} &= \frac{11.71+14.79+11.11+14.08}{4} \\
&\quad - \frac{6.69+6.33+8.05+7.59}{4} = 5.76 \\
D_{\mathrm{pH}} &= \frac{14.79+14.08+8.05+7.59}{4} \\
&\quad - \frac{6.69+6.33+11.71+11.11}{4} = 2.17 \\
D_{T*\mathrm{PPD}} &= \frac{11.71+14.79+6.33+7.59}{4} \\
&\quad - \frac{6.69+8.05+11.11+14.08}{4} = 0.123 \\
D_{T*\mathrm{pH}} &= \frac{14.79+8.05+6.33+11.11}{4} \\
&\quad - \frac{6.69+11.71+14.08+7.59}{4} = 0.053 \\
D_{\mathrm{PPD}*\mathrm{pH}} &= \frac{6.69+14.79+6.33+14.08}{4} \\
&\quad - \frac{11.71+8.05+11.11+7.59}{4} = 0.858
\end{aligned} \tag{4.11}$$

With a standard deviation of 0.24 and a degree of freedom of $f = 3$ measured at factor level of run 3, we calculate for the experimental error

$$t(0.95,3)\,s = 3.18 \cdot 0.24 = 0.76 \tag{4.12}$$

The comparison of the experimental error with the absolute differences reveals that the main factors pH and PPD concentration show a significant effect (D_{PPD} and D_{pH} are higher than 0.76) while the effect of the temperature can be neglected in the range studied: 35–40 °C ($D_T < 0.76$).

Graphical inspection of the main factor effects can be carried out as in Figure 4.10. As found by the calculations, there is minimal influence by the temperature. If the enzymatic reaction is run at a common 37 °C, the method will be rugged against temperature fluctuations. Note that all measured initial rates are slightly higher at 40 °C than at 35 °C. Compared to the general experimental error, however, this effect has been found to be statistically insignificant.

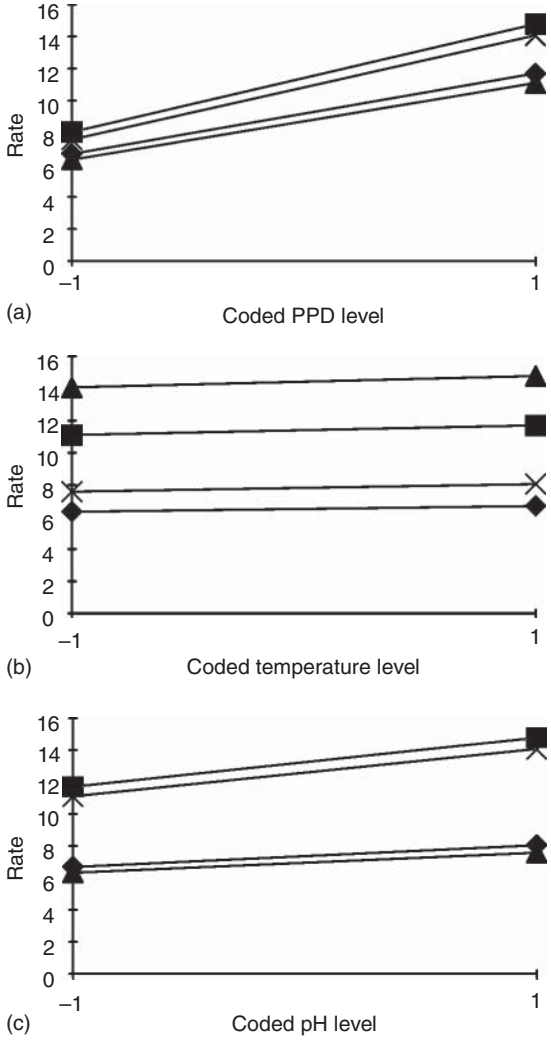


Figure 4.10 Factor effects in the kinetic–enzymatic oxidation of *p*-phenylenediamine (PPD) by the enzyme ceruloplasmin.

The enhancing effect of the substrate concentration PPD and of the pH on the rate can also be seen in Figure 4.10. As a general rule, main effects will give parallel straight lines in the factor effects plot. In the case of factor interactions, the slope of the straight line will differ if the levels of the alien factors change. The latter is the case for interactions of the pH and PPD factors. As can be seen from Figure 4.10, the changes of rates between the lower and higher levels of PPD are more pronounced, if the factor pH is at a high level (points ■ and x). A similar effect is observed if the rate changes in dependence on pH are compared at high (points ■ and x) and low (points ♦ and ▲) PPD concentrations.

The calculated interaction effects are significant for the interaction of the substrate PPD and the pH value ($D_{\mathrm{PPD}^{*}\mathrm{pH}} > 0.75$).

In contrast to the simultaneous factorial design study, experimentation by variation of one variable at a time is limited to the estimation of main effects, and no interactions, as are common in analytical chemistry, can be found. What cannot be evaluated with screening designs are curved dependences, that is, for more complicated relationships between responses and factors, designs at three or more factor levels are needed.

Three-Level Designs: Response Surface Designs

In order to describe the relationship between responses and factors quantitatively, we will use mechanistic (physicochemical) or empirical models, for example, polynomial models. These mathematical models should be able to describe linear and curved response surfaces similarly. Curved dependences can be modeled if the factor levels have at least been investigated at three levels.

The *degree of freedom* of a factorial design is the number of runs minus the number of independent factor combinations.

Three-level factorial designs are known, therefore, as response surface designs. Full-factorial three-level designs can be formalized in the same way as known for two-level designs, that is, a 3^k design means k factors at three factor levels. Figure 4.11 gives an example for a 3^2 design.

Full-factorial three-level designs are sometimes used for investigating few factors (two or three) although their statistical properties with respect to symmetry or confounding of parameter estimates are less favorable than those known for the two-level designs. In the case of many factors, the same problem as with

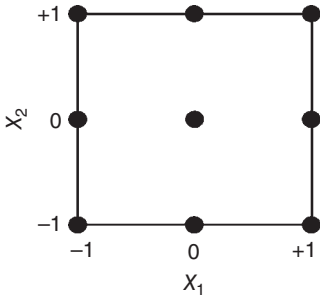


Figure 4.11 Full-factorial three-level design for two factors, 3^2 design.

two-level designs arises, that is, the number of experiments gets very high. These disadvantages led to the development of the so-called optimal designs, of which the *central composite design* and the *Box–Behnken design* are the most important ones.

Central Composite Design

Composite designs consist of a combination of a full or fractional factorial design and an additional design, often a star design. If the centers of both designs coincide, they are called *central composite designs*. Consider a design that consists of a full-factorial two-level design linked to a star design. For the number of runs r, we obtain

$$r = 2^{k-p} + 2k + n_0 \tag{4.13}$$

where k is the number of factors, p the number for reduction of the full design, and n_0 the number of experiments in the center of the design.

Example 4.5 *Runs in a Central Composite Design*

For a design with three factors, one gets 15 for the number of experiments r according to Eq. (4.13): $r = 2^3 + 2 \times 3 + 1 = 15$

A complete three-factor central composite design is depicted in Table 4.10 and Figure 4.12. The distance of the star points α from the center can be differently chosen. For a uniformly rotatable design,

$$\alpha = 2^{(k-p)/4} \tag{4.14}$$

for example, for three factors, $\alpha = 2^{(3-0)/4} = 1.682$.

To estimate the experimental error, replications of factor combinations are necessary. Usually, the center point is run thrice. The total number of runs in a central composite design with three factors amounts then to 17.

Replication here means carrying out a given factor combination several times.

Table 4.10 Central composite design for three factors consisting of a full-factorial two-level design and star design.

Run	Factors			Response
	x_1	x_2	x_3	
2^3 Kernel design				
1	−1	−1	−1	y_1
2	+1	−1	−1	y_2
3	+1	+1	−1	y_3
4	−1	+1	−1	y_4
5	−1	−1	+1	y_5
6	+1	−1	+1	y_6
7	+1	+1	+1	y_7
8	−1	+1	+1	y_8
$2k$ Star points				
9	$-\alpha$	−0	−0	y_9
10	$+\alpha$	−0	−0	y_{10}
11	−0	$-\alpha$	−0	y_{11}
12	−0	$+\alpha$	−0	y_{12}
13	−0	−0	$-\alpha$	y_{13}
14	−0	−0	$+\alpha$	y_{14}
Center point				
15, 16, 17	−0	−0	−0	y_{15}, y_{16}, y_{17}

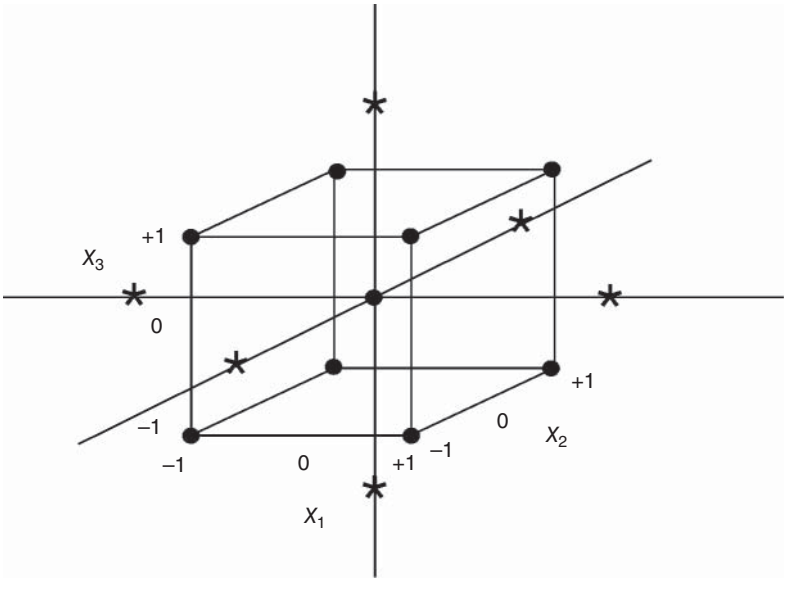


Figure 4.12 Central composite design for three factors.

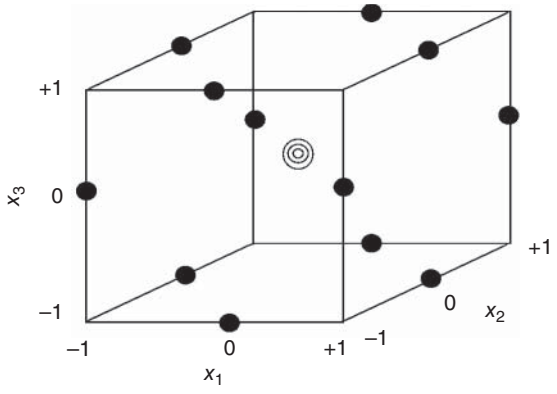


Figure 4.13 Box–Behnken design for three factors.

Apart from good statistical properties of the central composite design, there is one experimental disadvantage. Because the star points are outside the hypercube, the number of levels that have to be adjusted for every factor is actually five instead of the three in a conventional three-level design. If the adjustment of levels is difficult to achieve, an alternative response surface design would be the design introduced by Box and Behnken.

Box–Behnken Design

In a Box–Behnken design, the experimental points lie on a hypersphere equidistant from the center point as exemplified for a three-factor design in Figure 4.13 and Table 4.11. In contrast to the central composite design, the factor levels have only to be adjusted at three levels. In addition, if two replications are again performed in the center of the three-factor design, the total number of experiments is 15 compared to 17 with the central composite design.

In a Box–Behnken design, the experimental points lie on a sphere rather than on a cube.

In general, the Box–Behnken design requires few factor combinations, as is given in Table 4.12. The usage of the design is further explained in "Response Surface Methods" Section.

One disadvantage of Box–Behnken designs might be the representation of responses in dependence on a single factor. This is because the corner points of the cube are not measured but have to be computed by an appropriate response surface model.

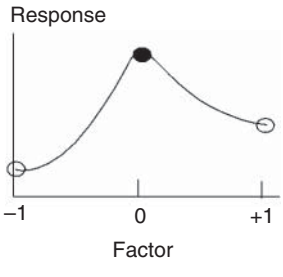


Table 4.11 Box–Behnken design for three factors.

Run	Factors			Response
	x_1	x_2	x_3	
1	+1	+1	−0	y_1
2	+1	−1	−0	y_2
3	−1	+1	−0	y_3
4	−1	−1	−0	y_4
5	+1	−0	+1	y_5
6	+1	−0	−1	y_6
7	−1	−0	+1	y_7
8	−1	−0	−1	y_8
9	−0	+1	+1	y_9
10	−0	+1	−1	y_{10}
11	−0	−1	+1	y_{11}
12	−0	−1	−1	y_{12}
13, 14, 15	−0	−0	−0	y_{13}, y_{14}, y_{15}

Table 4.12 Number of factors and experimental points for the Box–Behnken design with three replications in the center of each design.

Number of factors	Number of experiments
3	15
4	27
5	46
6	54
7	62

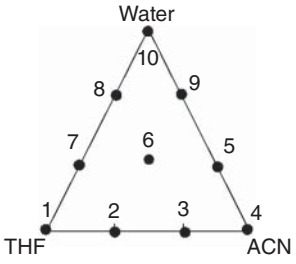
Mixture Designs

A special problem arises if additional relationships hold between the factors in an analytical investigation. This is true if mixtures, such as eluents in liquid chromatography or formulations in the pharmaceutical and textile industry, are under investigation. The constituents of a mixture given in portions of weight, volume, or mole are confined to the assumption that the amounts of the N constituents sum up to 100% or (normalized) to 1, that is,

$$\sum_{i=1}^{N} x_i = 1 \quad \text{for } x_i \geq 0 \tag{4.15}$$

The most important mixture design for an analyst is based on a so-called (k,d) lattice. For the k factors, the lattice describes all experimental points having the factor levels 0, $1/d$, $2/d, \ldots,$ $(d-1)/d$ or 1. In total, the (k, d) lattice has the following number of points:

$$\binom{d+k-1}{d} \tag{4.16}$$



Ternary mixtures of water, tetrahydrofuran (*THF*), and acetonitrile (*ACN*) as mobile-phase compositions in HPLC.

Example 4.6 *Lattice Design*

Consider a (3,3) lattice design as given in Figure 4.14. The number of points is calculated by

$$\binom{3+3-1}{3} = \binom{5}{3} = \frac{5!}{(5-3)!3!} = \frac{5\cdot4\cdot3\cdot2\cdot1}{2\cdot1\cdot3\cdot2\cdot1} = 10$$

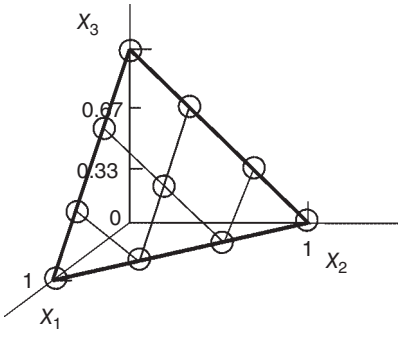


Figure 4.14 Mixture design for three factors at 10 levels based on a (3,3)-lattice design.

The factor levels are 0, 1/3, 2/3, 1, that is, 0, 0.33, 0.66, 1.00, respectively. In HPLC, the three constituents could be the solvents methanol, ACN, and water of the mobile phase.

Response Surface Methods

RSMs are very useful in order to quantify and interpret the relationships between responses and factor effects. In analytical chemistry, the relationships can be based on physical or physicochemical models that are generalized by statisticians as the so-called *mechanistic* models. Another way is empirical modeling, where the parameters have no mechanistic meaning.

General empirical models are polynomials of the second order, where the response y is related to the variables (factors) x as follows:

$$y = b_0 + \sum_{i=1}^{k} b_i x_i + \sum_{1 \le i \le j}^{k} b_{ij} x_i x_j + \sum_{i=1}^{k} b_{ii} x_i^2 \tag{4.17}$$

where k is the number of variables (factors), b_0 the intercept parameter, and b_i, b_{ij}, b_{ii} the regression parameters for linear, interaction, and quadratic factor effects, respectively.

To estimate all parameters in Eq. (4.17), the experiments have to be carried out at three factor levels, as discussed earlier for the response surface designs. The responses of experiments based on two factor levels can also of course be explored by RSM. However, no parameter estimates can then be obtained for curved (quadratic) factor effects, but only linear and interaction effects can be visualized.

The estimation of the empirical parameters is a general problem of least squares estimation by linear models. The basics are introduced in Section 6.2. With the parameters in hand, the model can be used to plot the response in dependence on the individual factors. The RSM is explained here by further exploring the enzymatic determination of ceruloplasmin from the previous section about screening designs.

Small deviations from the factor levels −1, 0, and +1 will not significantly alter the statistical properties of a design.

Example 4.7 *Response Surface Methods*

In the aforementioned study, the factors pH value and PPD were found to significantly influence the enzyme catalyzed oxidation of the substrate PPD. To study the relationship between the response (initial reaction *rate*) and the significant factors quantitatively, a design at three levels, a Box–Behnken design, is run at a temperature of 37 °C. The concentration of the enzyme ceruloplasmin (CP) is included as a third factor in order to investigate the response characteristics of the analyte ceruloplasmin. Details of the Box–Behnken experiments are outlined in Table 4.13. The three levels for each factor are given along with the concrete experimental design and the results of rate measurements.

Table 4.13 Factor levels and Box–Behnken design for studying the ceruloplasmin (CP) determination by RSM.

Factor	Level		
	−1	0	+1
PPD (mM)	0.5	14.3	27.3
pH	4.8	5.6	6.4
CP (mg l^{-1})	0.7	13.6	26.0

Run		Factors			Rate y (min^{-1})
Systematic	Randomized	PPD	pH	CP	
1	2	+1	+1	0.02	20.67
2	12	+1	−1	0.02	12.67
3	13	−1	+1	0.02	8.21
4	4	−1	−1	0.02	6.58
5	6	+1	0	+1	37.2
6	7	+1	0	−1	5.27
7	1	−1	0	+1	14.95
8	9	−1	0	−1	1.87
9	11	0.03	+1	+1	33.63
10	14	0.03	+1	−1	4.4
11	10	0.03	−1	+1	26.02
12	15	0.03	−1	−1	1.06
13	5	0.03	0	0.02	23.86
14	3	0.03	0	0.02	24.43
15	8	0.03	0	0.02	23.29

Parameter estimation with a polynomial of second order reveals the following final model:

$$y = 22.13 + 5.48\text{PPD} + 2.54\text{pH} + 12.3\text{CP} + 4.73\text{PPD} \cdot \text{CP} - 6.32\text{PPD}^2 - 5.02\text{pH}^2 \quad (4.18)$$

From the empirical model, it can be concluded that the three factors pH, PPD, and the enzyme concentration (CP) show main and quadratic effects (cf. Eq. (4.18)) on the rate of the reaction. In addition, there is a statistically significant interaction between the factors substrate (PPD) and enzyme (CP). In analytical terms, this means that for determinations of the enzyme, the substrate concentration should be kept constant with high precision.

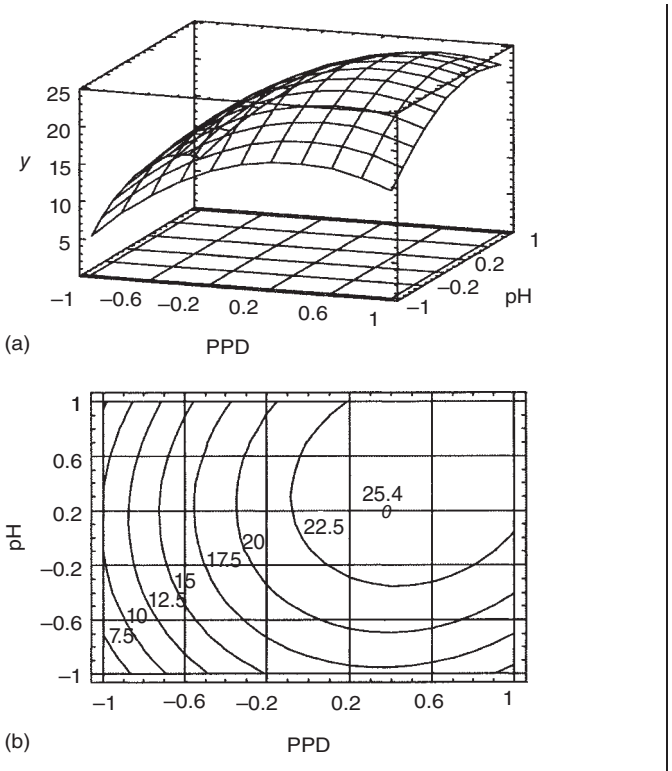


Figure 4.15 Surface (a) and contour (b) plots of enzymatic rate versus the factors PPD concentration and pH. The *contour lines* in b represent iso-rate lines.

Based on the mathematical model, the response surfaces can be explored graphically. An example plot of the response rate in dependence on PPD concentration and pH is shown in Figure 4.15a. The curved dependences in the direction of both factors lead to a maximum rate at coded levels of PPD of about 0.4 and of pH at 0.2. This relates to decoded levels of 16.6 mM PPD and a pH value of 5.95. Maxima are best found from the contour plots as represented in Figure 4.15b.

Coding of Factor Levels To convert factor levels between original and coded levels, the corresponding formulas will be given here. The coded level in the interval −1 to +1, x_i^*, is calculated from the original factor level, x_i, by

$$x_i^* = \frac{x_i - M}{H} \tag{4.19}$$

where M is the midrange and H the half-range of factor *i:*

$$M = \frac{\text{high} + \text{low}}{2} \tag{4.20}$$

$$H = \frac{\text{high} - \text{low}}{2} \tag{4.21}$$

Decoding of a factor level is carried out by converting Eq. (4.19) to the original factor level:

$$x_i = x_i^* \cdot H + M \tag{4.22}$$

Example 4.8 *Coding and Decoding of Factors*

- For PPD concentrations in the range 0.5–27.3 mM, the coded level for a concentration of 14.3 mM is to be determined (cf. Table 4.13). The midrange and half-range are obtained according to Eqs. (4.20) and (4.21):

  $$M = \frac{27.3 + 0.5}{2} = 13.9$$

  $$H = \frac{27.3 - 0.5}{2} = 13.4$$

- The resulting coded value for the PPD concentration is according to Eq. (4.19):

  $$x^* = \frac{14.3 - 13.9}{13.4} = 0.03$$

- Based on the coded value of a PPD concentration of 0.2, the original concentration is to be computed by means of the data in Table 4.13. For this, we apply Eq. (4.22) and the concentration value sought is calculated as

  $$x = 0.2 \cdot 13.4 + 13.9 = 16.6 \text{ mM}$$

Factor Effects versus Regression Parameters Although we have considered factor effect calculations and regression parameter estimation independently, it is important to understand that both concepts are linked together. More exactly, the following relationship holds:

$$\text{regression\_coefficient} = \frac{\text{factor\_effect}}{\text{factor\_range}} \tag{4.23}$$

Consider our enzyme example. Modeling the rate dependence by a polynomial that accounts for the same factors as in the screening design in Table 4.9, we will obtain the regression equation

$$y = 10.04 + 2.88\text{PPD} + 1.084\text{pH} + 0.4288\text{PPD} \cdot \text{pH} \quad (4.24)$$

The insignificant regression parameters have been eliminated in this equation. If you now compare the factor effects D in Eq. (4.11) with the regression coefficients, you will find that the latter are half as large as the factor effects, that is,

$$D_{\text{PPD}} = 5.76$$
$$D_{\text{pH}} = 2.17$$
$$D_{\text{PPH·pH}} = 0.858$$

This is because the coded factor range is equal to 2 (from −1 to +1) for all three factors and is therefore within rounding-off errors. In this example, the factor effects should be two times larger than the regression parameters.

Blocking of Experiments If a large number of experiments is to be carried out, it will be difficult to run the experiments under identical conditions. During experimentation, reagent charges might change or the activity of an enzyme might deteriorate. Often, it will be necessary to interrupt experimentation during the night so that important changes in the experimental conditions may result.

To reflect systematic changes in such situations, the sequence of experiments has so far been randomized. Strong systematic changes, however, will increase the overall experimental error to a large extent. Elimination of these systematic changes can be accounted for if the changes are taken as a discrete event and the estimation of the time-dependent effects are confounded with the estimation of unimportant interactions such as a three-factor interaction.

The experimental design is then divided into blocks.

Example 4.9 *Blocking of Experiments*

Considering the 2^3 design of Table 4.4, the blocks could be designed as the half-fraction designs given in Table 4.14.

Table 4.14 Full factorial 2^3 design arranged in two blocks.

Experiment	Factors				Response
	x_1	x_2	x_3	$x^*(x_1\ x_2\ x_3)$	
Block 1					
1	−1	−1	−1	+1	y_1
2	+1	+1	−1	+1	y_2
3	+1	−1	+1	+1	y_3
4	−1	+1	+1	+1	y_4
Block 2					
5	+1	−1	−1	−1	y_5
6	−1	+1	−1	−1	y_6
7	−1	−1	+1	−1	y_7
8	+1	+1	+1	−1	y_8

Estimation of parameters for, for example, the main effects would be performed by the following polynomial:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b^*x^* \tag{4.25}$$

Because of the orthogonality of the experimental design, the changes between the blocks will not influence the estimation of the parameters b_0 and b_3. In addition, an averaged measure for the changes between the two blocks can be derived from the size of $2b^*$.

4.4 Sequential Optimization: Simplex Method

"Change and hope" is the name sometimes given to an optimization procedure where the individual factors are changed independently of each other. As long as no interaction effects of factors are valid, the single-factor-at-a-time approach will succeed. In Figure 4.16a, this situation is explored for a response surface including curved factor effects. If one starts with variation of factor 2, keeping factor 1 at the coordinate value labeled (1), an optimal value will be found at the coordinate (2) for factor 2. In the next step, factor 1 would be investigated at a constant factor 2 value at label (2) and the optimum would be found.

However, if interaction effects become valid, the single-factor-at-a-time approach does not guarantee that the optimum is

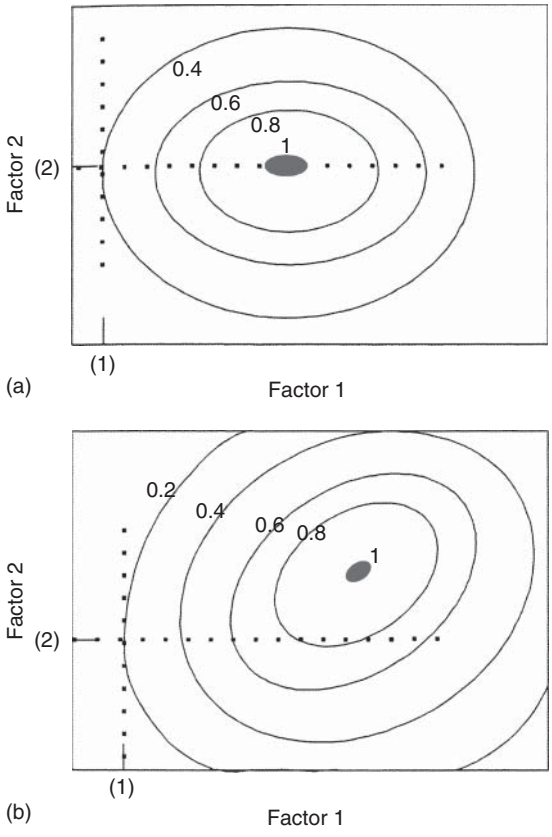


Figure 4.16 Sequential optimization with single-factor-at-a-time strategy in the case of a response surface without factor interactions (a) and for a surface with interactions (b).

reached. As seen in Figure 4.16b, the change in factor 2 will result in an optimal coordinate at label (2) that would be fixed for changing factor 1. In this case, the real optimum will never be found and the result remains suboptimal. The reason is that the ridge in Figure 4.16b does not lie parallel to the factors axis; thus, changes in factor 1 are not independent of factor 2. Instead, both factors interact and have to be considered simultaneously.

The most common sequential optimization method is based on the simplex method by Nelder and Mead. A simplex is a geometric figure having a number of vertices equal to one more than the number of factors. A simplex in one dimension is therefore a line, in two dimensions a triangle, in three dimensions a tetrahedron, and in multiple dimensions a hypertetrahedron.

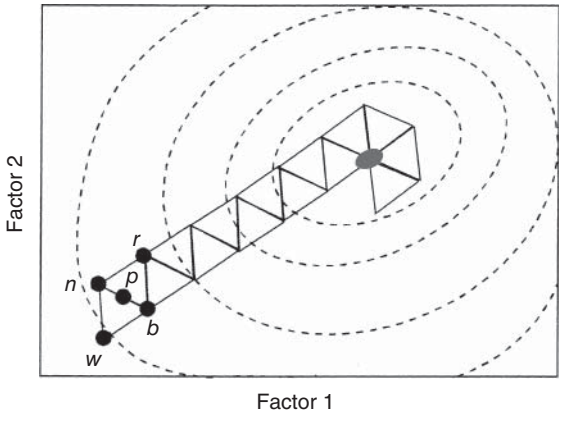


Figure 4.17 Fixed-size simplex according to Nelder and Mead along an unknown response surface.

Fixed-Size Simplex

To find the steepest path along a response surface by means of the simplex method, an algorithm has to be followed that consists of designing an initial simplex, running the experiments at the initial vertices, and calculating the new vertex point by reflection of the vertex with the worst response. Movement with steps of fixed size is called the *fixed-size simplex*.

The algorithm works as follows (cf. Figure 4.17):

- Generate the initial simplex according to the coded levels of the factors as given in Table 4.15.
- Run the experiments at the initial simplex coordinates.
- Decide from the responses which vertex represents the best response (vector $\boldsymbol{b}$), the next-to-best ($\boldsymbol{n}$), and the worst response ($\boldsymbol{w}$).
- Calculate the new experimental point by

$$\boldsymbol{r} = \boldsymbol{p} + (\boldsymbol{p} - \boldsymbol{w}) \tag{4.26}$$

where $\boldsymbol{p}$ is the centroid of the face remaining if the worst vertex $\boldsymbol{w}$ has been eliminated from the full simplex.

The centroid is calculated according to

$$\boldsymbol{p} = \frac{1}{n} \sum_{j \neq i}^{n} \boldsymbol{v}_j \tag{4.27}$$

where n is the number of factors, i the index of the worst vertex to be eliminated, and j the index of the considered vertex.

These steps are repeated until the simplex begins to rotate around the optimum or the response satisfies the experimenters'

Table 4.15 Choice of initial simplexes for up to nine variables coded in the interval between 0 and 1.

Experiments	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	0	—	—	—	—	—	—	—	—
2	1	0	—	—	—	—	—	—	—
3	0.50	0.87	0	—	—	—	—	—	—
4	0.50	0.29	0.82	0	—	—	—	—	—
5	0.50	0.29	0.20	0.79	0	—	—	—	—
6	0.50	0.29	0.20	0.16	0.78	0	—	—	—
7	0.50	0.29	0.20	0.16	0.13	0.76	0	—	—
8	0.50	0.29	0.20	0.16	0.13	0.11	0.76	0	—
9	0.50	0.29	0.20	0.16	0.13	0.11	0.094	0.75	0
10	0.50	0.29	0.20	0.16	0.13	0.11	0.094	0.083	0.75

needs. The fixed step width of the fixed-size simplex may reveal problems if the step width chosen is either too large or too small. In the former case, the optimum might be missed, and in the latter case, the number of required experiments becomes very large. These disadvantages can be circumvented if the step width is tunable as with the variable-size simplex.

Variable-Size Simplex

With the variable-size simplex, the step width is changed by expansion and contraction of the reflected vertices. The algorithm is modified as follows (cf. Figure 4.18):

$$\boldsymbol{r} = \boldsymbol{p} + (\boldsymbol{p} - \boldsymbol{w}) \tag{4.26}$$

- If $\boldsymbol{r}$ is better than $\boldsymbol{b}$, expand the simplex

  $$\boldsymbol{e} = \boldsymbol{p} + \alpha(\boldsymbol{p} - \boldsymbol{w}) \tag{4.28}$$

  with $\alpha > 1$, for example, 1.5 for all directions or α is chosen differently for each direction.
- If $\boldsymbol{r}$ lies between $\boldsymbol{b}$ and $\boldsymbol{n}$, keep the simplex $\boldsymbol{bnr}$.
- If $\boldsymbol{r}$ is worse than $\boldsymbol{n}$, contract the simplex according to

  $\boldsymbol{r}$ is worse than $\boldsymbol{n}$ but better than $\boldsymbol{w}$, then contract in the "positive" direction:

  $$\boldsymbol{c}_+ = \boldsymbol{p} + \alpha(\boldsymbol{p} - \boldsymbol{w}) \quad \text{with } 0 < \beta < 0.5 \tag{4.29}$$

  $\boldsymbol{r}$ is worse than $\boldsymbol{w}$, then contract in the "negative" direction:

  $$\boldsymbol{c}_- = \boldsymbol{p} - \alpha(\boldsymbol{p} - \boldsymbol{w}) \quad \text{with } 0 < \beta < 0.5 \tag{4.30}$$

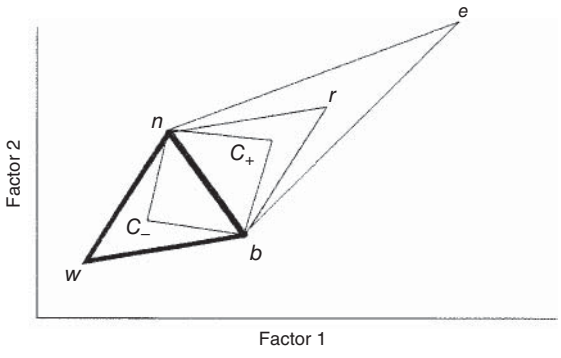


Figure 4.18 Variable-size simplex.

- At the experimental boundaries, the simplex is reflected into the space of the experimental variables.
- Stop the simplex if the signal change is less than the experimental error or if the step width is less than a given threshold.

In practice, the simplex method is the experimental optimization algorithm used most often. The main advantages are its simplicity, speed, and good convergence properties. Problems with the simplex method arise if multimodal response surfaces are investigated, that is, if several local optima exist. In this case, the simplex will climb the nearest local maximum or minimum and the global optimum might be missed. Mathematical theory provides more efficient optimization methods, such as the conjugate gradient method or Powell's method. These methods, however, are mainly used in locating optima of mathematical functions and are scarcely used in experimental optimization.

Local optima are typical for optimization of selectivity in HPLC separations. This is caused by changes of elution order of peaks in different mobile phases.

Example 4.10 *Simplex Optimization*

In this example, the performance of the variable-size simplex is demonstrated for the enzyme determination based on the problem in Examples 4.4 and 4.7. For a fixed enzyme concentration of 13.6 $mg \cdot l^{-1}$ ceruloplasmin (coded 0), the concentration of the substrate PPD and the pH value are sought for the maximum rate of the reaction, *y*. Since the simplex searches for a minimum, the rate as the objective criterion has to be

transformed. Here we use the difference of 100 min^{-1}, that is, the objective criterion is $100 - y$.

The *initial simplex* is chosen according to the scheme in Table 4.15 in coded levels. The responses for the initial simplex are as follows:

Vertex	Coded variable		$100 - y$
	PPD	pH	
1	0.0	0.0	76.547
2	1.0	0.0	77.550
3	0.5	0.87	76.450

The initial simplex is shown by bold lines in Figure 4.19.

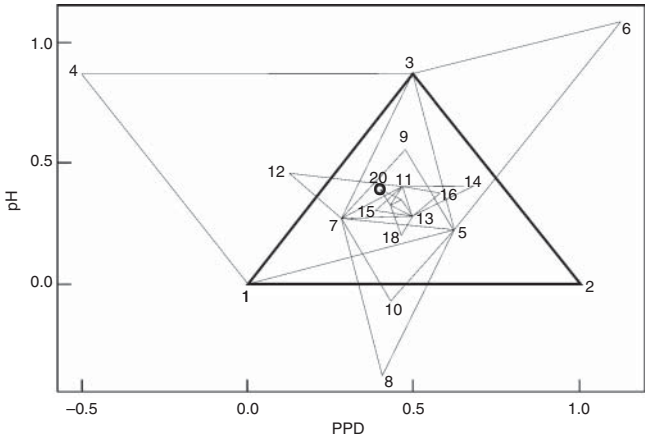


Figure 4.19 Simplex search for optimum PPD concentration and pH value for the enzymatic determination of ceruloplasmin (cf. Example 4.7).

The best (minimum) response is found for vertex 3 (best response $\boldsymbol{b}$), the next-to-best response is 1, (vertex $\boldsymbol{n}$), and the worst response is found for vertex 2 ($\boldsymbol{w}$). The latter vertex is eliminated, and a new (reflected) vertex 4 is calculated based on the centroid (Eq. (4.27)) according to Eq. (4.26):

centroid:

$$\boldsymbol{p} = \frac{1}{2}[(0, 0) + (0.5, 0.87)] = (0.25, 0.435)$$

reflection:

$$\boldsymbol{r} = \boldsymbol{p} + (\boldsymbol{p} - \boldsymbol{w}) = (0.25, 0.435) + [(0.25, 0.435) - (1, 0)]$$
$$= (-0.5, 0.87)$$

The new simplex consists now of the vertices 1, 3, and 4 having the following responses:

Vertex	Coded variable		$100 - y$
	PPD	pH	
1	0.0	0.0	76.547
3	0.5	0.87	76.450
4	−0.5	0.87	83.317

The vertex 4 ($\boldsymbol{r}$) produces a worse response than the next-to-best vertex 1 and is even worse than $\boldsymbol{w}$. Therefore, the simplex is contracted according to Eq. (4.30). At first, the centroid is calculated without the worst vertex 4. Here the centroid is the same as in the first step:

centroid:

$$\boldsymbol{p} = \frac{1}{2}[(0, 0) + (0.5, 0.87)] = (0.25, 0.435)$$

contraction:

$$\boldsymbol{c}_{-} = \boldsymbol{p} - \beta(\boldsymbol{p} - \boldsymbol{w}) = (0.25, 0.435)$$
$$- 0.5\,[(0.25, 0.435) - (-0.5, 0.87)] = (0.625, 0.218)$$

The new simplex is 1, 3, and 5. The coordinates of this simplex and the measured responses are as follows:

Vertex	Coded variable		$100 - y$
	PPD	pH	
1	0.000	0.000	76.547
3	0.500	0.870	76.450
5	0.625	0.218	75.131

After contraction, the simplex is reflected to point 6 and again contracted to vertex 7 (cf. Figure 4.19). If the calculation is continued, the simplex moves in the direction of the optimum as given in the figure. After 20 iterations, the following optimum is found in coded levels:

- PPD = 0.46
- pH = 0.316
- $100 - y = 74.88$

Thus, the maximum rate $y = 100 - 74.88 = 25.12\ \text{min}^{-1}$. This optimum compares well with the results from the response surface study in Figure 4.15.

General Reading

1. Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1987) *Statistics for Experimenters: an Introduction to Design, Data Analysis and Model Building*, John Wiley & Sons, Inc., New York.
2. Box, G.E.P. and Draper, N.R. (1987) *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, Inc., New York.
3. Deming, S.N. and Morgan, S.L. (1987) *Experimental Design: a Chemometric Approach*, Elsevier, Amsterdam.
4. Nelder, J.A. and Mead, R. (1965) *Comput. J.*, **7**, 308.
5. Massart, D.L., Vandeginste, B.G.M., Deming, S.N., Buydens, L.C.M., De Jong, S., Lewi, P.J., and Smeyers-Verbeke, J. (1998) *Handbook of Chemometrics and Qualimetrics, Parts A and B*, Elsevier, Amsterdam.
6. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. **1**, Elsevier, Amsterdam.

Questions and Problems

1. Specify the following characteristics with a standard statistical software package:
- Number of experiments
- Specific design
- Randomized run sequence
- Possible blocking
- Alias structures

for the two-level designs 2^9, 2^{7-3}, 2^{10-5}, as well as for central composite designs based on three and six factors.

2. Give definitions for the following terms: unimodal; multimodal; replication of experiments; factor level; coding of factors; local and global optimum; randomization; screening designs; factor effects.

3. Graphical inspection of response surfaces is restricted to three-dimensional plots. How do you plot response surfaces if more than two factors are included in the study?

4. How can one avoid factors and their parameter estimates being confounded?

5. How can one account for interactions of factors in a polynomial model?

6. What difficulties may arise if experimental factors are changed, one by one?

7. Why do chemists run their experiments mostly by one-factor-at-a-time methods?

5
Pattern Recognition and Classification

Learning Objectives

- To evaluate and interpret analytical data from full chromatograms, spectra, depth profiles, or electroanalytical records, from multidimensional detectors, and from samples for which concentrations of several chemical constituents or other properties have been measured
- To learn about methods for data preprocessing and for calculating distances and similarity measures
- To introduce grouping of analytical data based on unsupervised learning methods, that is, projection methods and cluster analysis
- To handle multivariate data for which their class membership is determined by means of supervised pattern recognition approaches.

Modern analytical instrumentation generates a vast amount of data. A digitized spectrum in the infrared (IR) spectral range consists of about 2000 wavenumber points. In a gas chromatography–mass spectrometry (GC-MS) experiment, it is not difficult to provide in a single run 600 000 items of data that amount to about 2.4 MB of digital information. There are different methods for dealing with this extensive amount of information. One approach is ignorance. This means that quantitative analysis in spectroscopy is restricted to data evaluation at a single wavelength or the GC-MS trace is followed at a single mass unit. However, with the invention of computers in the analytical laboratory, most of the multidimensional data are stored in instrument computers or external databases, so that important information might be wasted if only small fractions of data are evaluated. Nowadays, the extensive use of chemometrics enables

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

the evaluation and interpretation of these data to be carried out efficiently as will be examined in this chapter.

The main objectives for application of multivariate methods in analytical chemistry are aimed at the grouping and classification of objects (samples, compounds, or materials) as well as at modeling relationships between different analytical data. Some typical examples include the following:

- *Grouping* or *clustering* of rock samples according to similar elemental patterns or of material samples with respect to comparable chemical composition and technological properties.
- *Classification* of samples, such as rocks, materials, or chemical compounds, by means of analytical data (spectrum, chromatogram, or elemental pattern) on the basis of known class membership of those objects.
- Calibration of a single chemical constituent by means of full spectrum, or calibration of several components by means of mixture calibration techniques. In mathematical terms, these are problems of *parameter estimation* where the parameters represent the calibration coefficients.

Clustering and classification methods are summarized by the notion *pattern recognition.*

To understand the multidimensionality of analytical problems, let us consider a practical example. In connection with the elucidation of a crime, a hair saved at the scene of the crime is to be assigned to a subject. Apart from the common morphological investigations, the hair is analyzed for the elements copper, manganese, chlorine, bromine, and iodine. For comparison, the elemental contents of hair of three other (suspicious) subjects are available (Table 5.1).

Initially, it is determined whether the hair of the three subjects can be distinguished on the basis of their elemental patterns. For

Table 5.1 Elemental contents of hair of different subjects in parts per million.

Hair	Cu	Mn	Cl	Br	I
1	9.2	0.30	1730	12.0	3.6
2	12.4	0.39	930	50.0	2.3
3	7.2	0.32	2750	65.3	3.4
4	10.2	0.36	1500	3.4	5.3
5	10.1	0.50	1040	39.2	1.9
6	6.5	0.20	2490	90.0	4.6
7	5.6	0.29	2940	88.0	5.6
8	11.8	0.42	867	43.1	1.5
9	8.5	0.25	1620	5.2	6.2

this, chemometric methods for grouping of samples are needed. If grouping of the samples is feasible, then the found hair has to be assigned to a subject based on the construction of class models for the three subjects with subsequent classification of the unknown sample.

In general, the analytical data can be arranged as a data matrix $\boldsymbol{X}$ of n objects (rows) and p features (columns). The objects might be samples, molecules, materials, findings, or fertilizers. Typical features or variables of those objects will be elemental patterns, spectra, structural features, or physical properties. The $n \times p$ data matrix $\boldsymbol{X}$ can be written as follows:

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{5.1}$$

Feature variables may be defined on different scales. The *nominal scale* characterizes qualitative equivalence, for example, male and female. The *ordinal scale* describes ordering or ranking. The *interval scale* measures distances between values of the features. The *ratio scale* also enables quotients between feature values to be evaluated.

For some problems, these data are divided columnwise into dependent and independent variables, for example, for calibrating concentrations on spectra. The dependent variables are then renamed, for example, by the character y.

A *class* comprises a collection of objects that have similar features. The *pattern* of an object is its collection of characteristic features. For multivariate data evaluation, not all objects and features are necessarily used. On the other hand, some of the available data cannot be used as they are reported. Therefore, pretreatment of data is a prerequisite for efficient multivariate data analysis.

5.1 Preprocessing of Data

Missing Data, Centering, and Scaling

In the first step, the data have to be reviewed with respect to completeness. *Missing data* do not hinder mathematical analysis. Of course, missing data should not be replaced by zeros. Instead, the vacancies should be filled up either by the column/row mean or, in the worst case, by generating a random number in the range of the considered column/row. *Features* and/or *objects* can be removed from the data set if they are highly correlated with each other, or if they are redundant or constant.

To eliminate a constant offset, the data can be translated along the coordinate origin. The common procedure is *mean centering*, where each variable, x_{ik}, is centered by subtracting the column

Coded data are obtained by multiplying, dividing, adding, and/or subtracting a constant in order to convert the original data into more convenient values.

mean, $\overline{x}_k$, according to

$$x_{ik}^* = x_{ik} - \overline{x}_k \tag{5.2}$$

where i is the row index, k the column index, and $\overline{x}_k$ is the column mean calculated from

$$\overline{x}_k = \frac{1}{n} \sum_{i=1}^{n} x_{ik} \tag{5.3}$$

Very often, the features represent quite different properties of a sample or of an object, so that the metric might differ from column to column to a great extent. This may imply different absolute values of the variables as well as different variable ranges (variances). Both types of distortions will affect most of the statistically based multivariate methods. Elimination of these differences can be carried out by *scaling* the data to similar ranges and variances. Two scaling methods that scale the data by range or by standard deviation (autoscaling) are important, *range scaling*:

$$x_{ik}^* = \frac{x_{ik} - x_k(\min)}{x_k(\max) - x_k(\min)} \quad \in 0 \le x_{ik}^* \le 1 \tag{5.4}$$

and *autoscaling*:

$$x_{ik}^* = \frac{x_{ik} - \overline{x}_k}{s_k} \tag{5.5}$$

Dividing each variable by the square root of the standard deviation is called *Pareto scaling*.

$$\text{with} \quad s_k = \sqrt{\frac{\sum_{i=1}^{n} (x_{ik} - \overline{x}_k)}{n-1}} \tag{5.6}$$

where n is the number of objects. Autoscaling reveals data with zero mean and unit variance. The length of the vectors is scaled to $\sqrt{n-1}$.

Normalization to Length 1

In some cases, normalization of a data vector to length 1 is an important preprocessing procedure:

$$x_{ik}^* = \frac{x_{ik}}{\|\boldsymbol{x}_k\|} \tag{5.7}$$

where

$$\|\boldsymbol{x}_k\| = \sqrt{x_{1k}^2 + x_{2k}^2 + \cdots + x_{nk}^2} \tag{5.8}$$

Usually, the autoscaling method is the method of choice for scaling data. Figure 5.1 gives a graphical illustration for centering and autoscaling.

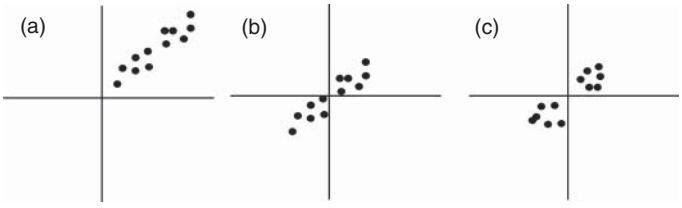


Figure 5.1 Demonstration of translation and scaling procedures: the original data in (a) are centered in (b) and autoscaled in (c). Notice that the autoscaling decreases the between-groups distance in the direction of greatest within-groups scatter and increases it in perpendicular direction in the sense of sphericization of groups.

Variance–Covariance Matrix and Correlation Matrix

Transformation of the original data to a new coordinate system is another possibility of data pretreatment. The methods are based on principal component analysis (PCA) or factor analysis (FA). The first step for these transformations is the formation of a data matrix that is derived from the original data matrix and that reflects the relationships among the data. Such derived data matrices are the variance–covariance matrix and the correlation matrix.

Variance–Covariance Matrix

The variance–covariance matrix or simply covariance matrix is computed from the data matrix, $\boldsymbol{X}$, in Eq. (5.1) from the variances of all p variables after Eq. (5.9) and their covariances according to Eq. (5.10).

$$s_{ij}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2 \quad \text{for } j = 1 \ldots p \tag{5.9}$$

$$\text{cov}(j,k) = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) \quad j,k = 1 \ldots p;\ j \neq k \tag{5.10}$$

For the variance–covariance matrix, it is valid that

$$\boldsymbol{C} = \begin{pmatrix} s_{11}^2 & \text{cov}\,(1,2) & \ldots & \text{cov}(1,p) \\ \text{cov}(2,1) & s_{22}^2 & & \text{cov}(2,p) \\ \vdots & & & \vdots \\ \text{cov}(p,1) & \text{cov}(p,2) & \ldots & s_{pp}^2 \end{pmatrix} \tag{5.11}$$

As a result, a symmetric matrix is obtained. The covariance matrix is used in cases where the metric of the variables is comparable. If widely different metrics are valid, for example, in

the case of simultaneous data treatment of main and trace constituents, the variables are scaled. If the variables are autoscaled (cf. Eq. (5.5)), the correlation matrix automatically results, as given in Eq. (5.14).

At this point, only one possibility for computing the variance–covariance matrix has been introduced. Another one we learn about in Section 5.2 in connection with FA.

Correlation Matrix

To calculate the correlation matrix, the correlation coefficients are required according to

$$r_{jk} = \frac{\mathrm{cov}(j,k)}{s_j s_k} = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\left[\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2 \sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2\right]^{1/2}} \quad j \neq k \quad (5.12)$$

In analogy to Eq. (5.9), the standard deviations s_j and s_k are calculated by

$$s_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}{n-1}} \quad s_k = \sqrt{\frac{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2}{n-1}} \quad (5.13)$$

The correlation matrix reads as follows:

$$\boldsymbol{R} = \begin{pmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{12} & 1 & & r_{2p} \\ \vdots & & & \vdots \\ r_{1p} & r_{2p} & \cdots & 1 \end{pmatrix} \quad (5.14)$$

Computations of the covariance and correlation matrix are prerequisites for the application of factorial methods.

5.2 Unsupervised Methods

Grouping of analytical data is possible either by means of clustering methods or by projecting the high-dimensional data onto lower dimensional space. Since there is no supervisor in the sense of known membership of objects to classes, these methods are performed in an unsupervised manner.

Factorial Methods

These methods are aimed at projecting the original data set from a high-dimensional space onto a line, a plane, or a three-dimensional coordinate system. Perhaps the best way would be to have a mathematical procedure that allows you to sit before the computer screen pursuing the rotation of the data into all possible directions and stopping this process when the best projection, that is, optimal clustering of data groups, has been found. In fact, such methods of projection pursuits already exist in statistics and are tested within the field of chemometrics.

At present, data projection is performed mainly by methods called PCA, FA, singular value decomposition (SVD), eigenvector projection, or rank annihilation. The different methods are linked to different science areas. They also differ mathematically in the way the projection is computed, that is, which dispersion matrix is the basis for data decomposition, which assumptions are valid, and whether the method is based on eigenvector analysis, SVD, or other iterative schemes.

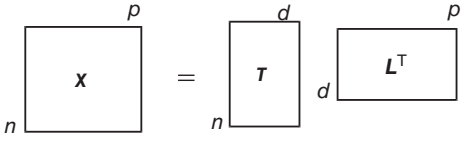
The *dispersion matrix* describes the scatter of multivariate data around the mean. For centered data, the dispersion matrix equals $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$.

Principal Component Analysis

Here, an explanation of projection methods is based on PCA in comparison to SVD. Similar methods, such as FA, are covered later in the section.

The key idea of PCA is to approximate the original matrix $\boldsymbol{X}$ by a product of two small matrices – the score and loading matrices – according to

$$\boldsymbol{X} = \boldsymbol{T}\,\boldsymbol{L}^{\mathrm{T}} \tag{5.15}$$



where $\boldsymbol{X}$ is the original data matrix consisting of n rows (objects) and p columns (features); $\boldsymbol{T}$ is the scores matrix with n rows and d columns (number of principal components (PCs)); $\boldsymbol{L}$ is the loading matrix with d columns and p rows; and T is the transpose of a matrix.

In other words, the projection of $\boldsymbol{X}$ down on a d-dimensional *subspace* by means of the projection matrix $\boldsymbol{L}^{\mathrm{T}}$ gives the object coordinates in this plane, $\boldsymbol{T}$. The columns in $\boldsymbol{T}$ are the score vectors and the rows in $\boldsymbol{P}$ are called *loading vectors*. Both vectors are orthogonal, that is, $\boldsymbol{p}_{\mathrm{i}}{}^{\mathrm{T}}\boldsymbol{p}_{\mathrm{j}} = 0$ and $\boldsymbol{t}_{\mathrm{i}}{}^{\mathrm{T}}\boldsymbol{t}_{\mathrm{j}} = 0$, for $i \neq j$.

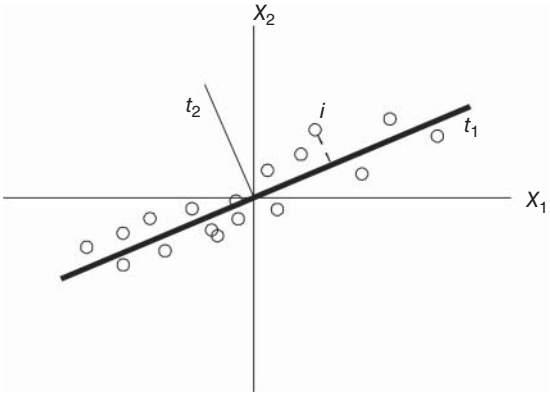
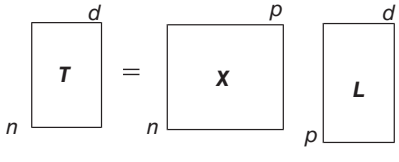


Figure 5.2 Projection of a swarm of objects from the original two dimensions onto one dimension, that is, the score vector t_1, according to the criterion of maximum variance.

In contrast to FA, the data are reconstructed such that new, uncorrelated variables are obtained. The PCs are determined on the basis of the maximum variance criterion (Figure 5.2). Each subsequent PC describes a maximum of variance that is not modeled by the former components. According to this, most of the variance of the data is contained in the first PC. In the second component, there is more information than in the third one and so on. Finally, as many PCs are computed as are needed to explain a preset percentage of variance (cf. Eq. (5.22)).

The PCs can be considered projections of the original data matrix, ***X,*** onto the scores, ***T.*** For this, Eq. (5.15) is to be converted to the scores on the left side by

$$\boldsymbol{T} = \boldsymbol{X}\,\boldsymbol{L} \tag{5.16}$$



The new coordinates are linear combinations of the original variables. For example, the elements of the first PC read as

$$\begin{aligned} t_{11} &= x_{11}l_{11} + x_{12}l_{21} + \cdots + x_{1p}l_{p1} \\ t_{21} &= x_{21}l_{11} + x_{22}l_{21} + \cdots + x_{2p}l_{p1} \\ &\vdots \\ t_{n1} &= x_{n1}l_{11} + x_{n2}l_{21} + \cdots + x_{np}l_{p1} \end{aligned} \tag{5.17}$$

Since, as a rule, a large fraction of the variance can be described by means of one, two, or three PCs, the data can be visualized by plotting the PCs against each other (cf. Example 5.2).

The simplest method for PCA used in analytics is the iterative nonlinear iterative partial least squares (NIPALS) algorithm explained in Example 5.1. More powerful methods are based on matrix diagonalization, such as SVD, or bidiagonalization, such as the partial least squares (PLS) method.

Principal components in PCA or common factors in factor analysis are sometimes called *latent variables*.

Example 5.1 *Iterative PCA by the NIPALS Algorithm*

Determine PCs for the following data matrix based on the iterative NIPALS algorithms:

$$\boldsymbol{X} = \begin{pmatrix} 2 & 1 \\ 3 & 2 \\ 4 & 3 \end{pmatrix}$$

0. Scaling by the mean and normalizing to length 1:

$$\boldsymbol{X} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

1. Estimation of the loading vector $\boldsymbol{l}^{\mathrm{T}}$. Usually, the first row of the $\boldsymbol{X}$ matrix is used:

$$\boldsymbol{l}^{T} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

2. Computation of the new score vector $\boldsymbol{t}$:

$$\boldsymbol{t} = \boldsymbol{X}\boldsymbol{l} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \tag{5.18}$$

Comparison of the new $\boldsymbol{t}$ vector with the old one. If the deviations of the elements of the two vectors are within a given threshold of 10^{-z}, for example, $z = 5$, then continue at step 5, else go to step 3.

3. Compute new loadings $\boldsymbol{l}^{\mathrm{T}}$:

$$\boldsymbol{l}^{\mathrm{T}} = \boldsymbol{t}^{\mathrm{T}}\boldsymbol{X} = \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} -\frac{2}{\sqrt{2}} & -\frac{2}{\sqrt{2}} \end{pmatrix} \tag{5.19}$$

Normalize the loading vector to length 1:

$$\boldsymbol{l}^{\mathrm{T}} = \frac{\boldsymbol{l}^{\mathrm{T}}}{\|\boldsymbol{l}^{\mathrm{T}}\|} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \tag{5.20}$$

4. Continue at 2 if the number of iterations does not exceed a predefined threshold, for example, 100; else go to step 5.
5. Determine the matrix of residuals:

$$\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{t}\boldsymbol{l}^{\mathrm{T}} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{5.21}$$

   If the number of PCs is equal to the number of previously fixed PCs or of cross-validated components, then go to step 7. Otherwise, continue at step 6.
6. Use the residual matrix $\boldsymbol{E}$ as the new $\boldsymbol{X}$ matrix and compute additional PCs $\boldsymbol{t}$ and loadings $\boldsymbol{l}^{\mathrm{T}}$ at step 1.
7. As a result, the matrix $\boldsymbol{X}$ is represented by a PC model according to Eq. (5.15), that is,

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{L}^{\mathrm{T}} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

The actual two-dimensional data can be described by just one PC.

With real data, more PCs are necessary. Therefore, there are more columns of scores in the $\boldsymbol{T}$ matrix and more rows in the $\boldsymbol{L}^{\mathrm{T}}$ matrix representing the loadings.

The factorial methods in this chapter are also called *second-order transformations*, because only two moments, mean and covariance, are needed to describe the Gaussian distribution of the variables. Other second-order transformations are FA, independent component analysis (ICA), and multivariate curve resolution (MCR).

Estimating the Number of PCs The use of all PCs after decomposition of the data matrix is usually not justified. For example, the number of pure components must be separated from the noise components in a spectrochromatogram.

To decide on the number of components in a PCA, several heuristic and statistical criteria exist:

- Percentage of explained variance
- Eigenvalue-one criterion
- Scree test
- Cross-validation.

The percentage of *explained variance* is applied in the sense of a heuristic criterion. It can be used if enough experience is gained by analyzing similar data sets. The fraction of explained (cumulative) variance, s_e^2, is calculated from the ratio of the sum of d important eigenvalues and the sum of all p eigenvalues by

$$s_e^2 = \frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \tag{5.22}$$

If all possible PCs are used in the model, 100% of the variance can be explained. Usually, a fixed percentage of explained variance is specified, for example, 90%. Equation (5.22) is then multiplied by the factor 100.

In our example of the hair data in Table 5.1, 90.7% of the data variance can be explained by two PCs (Table 5.2).

The *eigenvalue-one criterion* is based on the fact that the average eigenvalue of autoscaled data is just one. This results from the fact that for standardized data, the sum of all eigenvalues of the correlation matrix is exactly the number of all features p (cf. Eq. (5.15)). Only components with eigenvalues greater than one are considered important. According to this criterion, the eigenvalues of the hair data in Table 5.2 reveal two significant PCs.

The *scree test* is based on the phenomenon of the residual variance leveling off when the proper number of PCs is obtained. Visually, the residuals, or more often the eigenvalues, are plotted against the number of components in a scree plot. The component number is then derived from the leveling off in this dependence. Figure 5.3 demonstrates the scree plot for the hair data. The slope can be seen to change between the second and third components.

The fourth method for deciding on the number of PCs is *cross-validation*. In the simplest case, every object of the $\boldsymbol{X}$ matrix is

Table 5.2 Eigenvalues and explained variances for the hair data in Table 5.1.

Component	Eigenvalue λ	Explained variance (%)	Cumulative variance (%)
1	3.352	67.05	67.05
2	1.182	23.65	90.70
3	0.285	5.70	96.40
4	0.135	2.70	99.10
5	0.045	0.90	100.00

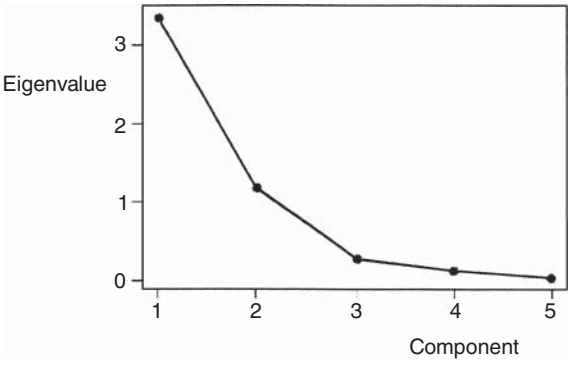


Figure 5.3 Scree plot for the principal component model of the hair data in Table 5.1.

removed from the data set once, and a model with the remaining data is computed. Then the removed data are predicted by means of the PCA model and the sum of the square root of residuals over all removed objects is calculated. The number of significant PCs is obtained from the minimum residual error.

In the case of large data sets, the leave-one-out method can be replaced by leaving out groups of objects. Other criteria for deciding on the number of PCs or factors are introduced as follows.

Graphical Interpretation of PCs Interpretation of the results of a PCA is usually carried out by visualization of the component scores and loadings. Sometimes, the data can be interpreted from a single component. Commercial software provides two- or three-dimensional plot facilities. The following example demonstrates the general procedure.

Example 5.2 *Principal Component Analysis*

The hair data in Table 5.1 on elemental contents are to be investigated by PCA. First, the grouping of the samples is recognized, and second, the importance of different elements for discrimination between the groups is discussed.

PCA on the basis of the correlation matrix of the data provides the results given in Figure 5.4 for the scores and Figure 5.5 for the loadings. Since the preliminary tests revealed only two significant PCs (cf. Table 5.1 and Figure 5.3), plots of the first two PCs are sufficient.

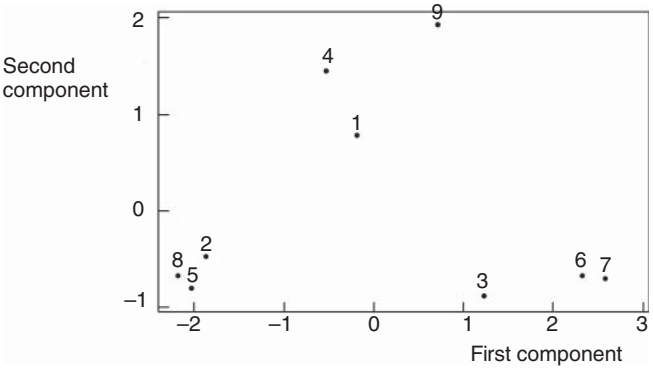


Figure 5.4 Principal component scores for the hair data in Table 5.1.

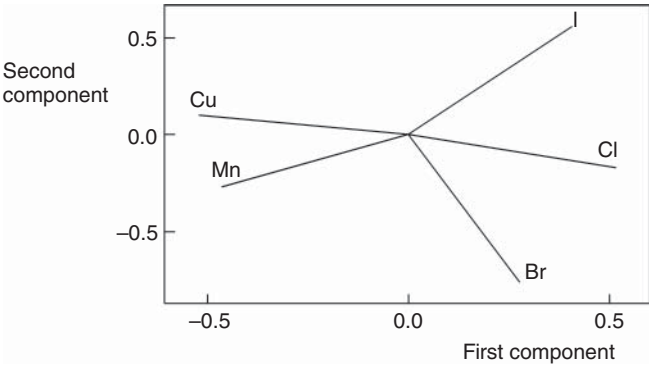


Figure 5.5 Principal component loadings of the hair data in Table 5.1.

In the score plot, the linear projection of objects is found to be representing the main part of the total variance of the data. As can be seen, there are three clusters with three objects (hair) each. These objects belong to the hair of different subjects.

The correlation and importance of feature variables are decided from plots of the PC loadings. Figure 5.5 demonstrates the loading plot of the first two components for the hair data of Table 5.1.

The loading plot provides the projection of the features onto the PCs. From this plot, information about the *correlation* of feature variables can be deduced. The correlation of features is described by the cosine of the angle between the loading vectors. The smaller the angle, the higher is the correlation between features. Uncorrelated features are orthogonal

Karhunen–Loève expansion is synonymous with PCA.

to each other. If variables are highly correlated, then it is sufficient to measure only one of the correlated variables.

The size of the loadings in relation to the considered PC is a measure of the *importance* of a feature for the PC model. Loadings in the origin of the coordinate system represent unimportant features.

In our hair data example (Figure 5.5), the elements Cu and Mn are more highly correlated than the halogens Br, Cl, and I. For Cu and I, anticorrelation is observed. All elements are important for describing the first PC. The second component is mainly characterized by the elements I and Br.

A joint interpretation of scores and loadings is possible if the loadings are properly scaled and superimposed on the score plots. The so-called *biplot* is given for the present example in Figure 5.6. From the loading direction, the discriminating ability of variables can be deduced. In our example, the features Cu, Mn, I, and Cl separate the object clusters into the groups (2,8,5) and (3,6,7), whereas the feature Br separates the left cluster (2,5,8) from the rest of the objects. The neighborhood of objects to a loading vector reflects the importance of that variable for building the PC model.

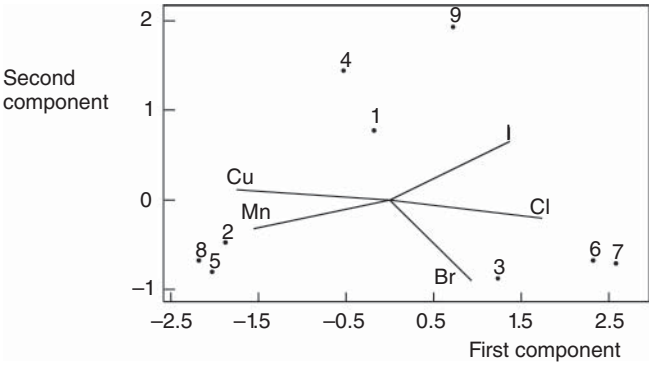


Figure 5.6 Biplot for the simultaneous characterization of the cores and loadings of two principal components of the hair data in Table 5.1.

The computation of PCs in Example 5.2 was based on SVD. Where the same task is solved by the NIPALS algorithm (Example 5.1), the projection of data onto the first two PCs reveals the result given in Figure 5.7. Apart from these two methods, many additional possibilities exist for decomposition of the original data matrix, including FA, which will be looked at later in this section.

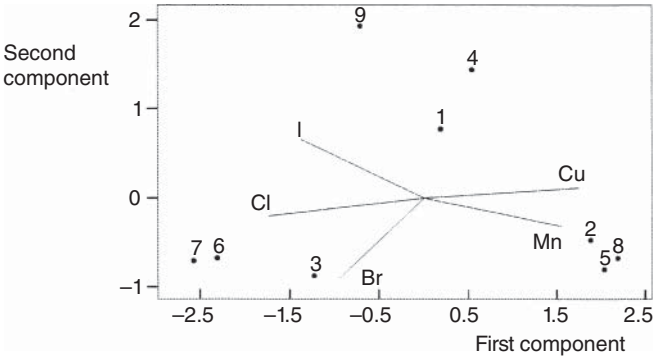


Figure 5.7 Biplot for the simultaneous characterization of the scores and loadings of two principal components of the hair data in Table 5.1 computed on the basis of the NIPALS algorithm.

To understand the relationship between PCA and FA, it is useful to outline the method of SVD in more detail here.

SVD – Singular Value Decomposition

SVD is based on the decomposition of a symmetric matrix, for example, the correlation matrix, into a threefold diagonal matrix and their diagonalization by means of the so-called QR algorithm. Details on the SVD algorithm are not important here. To understand the relationships between PC, factor, and eigenvalue analysis, however, it is useful to discuss the principle of SVD in matrix representation.

SVD decomposes the data matrix, $\boldsymbol{X}$, in Eq. (5.1) into the matrices $\boldsymbol{U}$, $\boldsymbol{W}$, and $\boldsymbol{V}$ as follows:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^{\mathrm{T}} \tag{5.23}$$

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \vdots & & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1d} \\ u_{21} & u_{22} & & u_{2d} \\ \vdots & & & \vdots \\ u_{n1} & u_{n2} & \cdots & u_{nd} \end{pmatrix} \begin{pmatrix} w_{11} & 0 & \cdots & 0 \\ 0 & w_{22} & & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & w_{dd} \end{pmatrix} \times \begin{pmatrix} v_{11} & v_{21} & \cdots & v_{p1} \\ v_{12} & v_{22} & & v_{p2} \\ \vdots & & & \\ v_{1d} & v_{2d} & & v_{pd} \end{pmatrix} \tag{5.24}$$

The matrix $\boldsymbol{U}$ contains the same column vectors as does the scores matrix $\boldsymbol{T}$ in Eq. (5.15) but normalized to length 1; $\boldsymbol{W}$ is the diagonal matrix containing the square roots of the eigenvalues or singular values. For symmetrical matrices ($n = p$), the singular

values are identical to the square roots of the eigenvalues, that is,

$$w_{ii} = \sqrt{\lambda_{ii}} \tag{5.25}$$

The dimension of the matrix of singular values corresponds to the number of columns in the scores matrix. If small singular values are not neglected, that is, not set to zero, the dimension of $\boldsymbol{W}$ is equal to the number of features, that is, $d = p$.

The matrix $\boldsymbol{V}^{\mathrm{T}}$ is identical to matrix $\boldsymbol{L}^{\mathrm{T}}$ in Eq. (5.15). $\boldsymbol{U}$ and $\boldsymbol{V}$ are also denoted as left and right vectors of singular values, respectively.

FA – Factor Analysis

The aim of FA is to express the features by a small number of *common* factors. For each factor, a property is assigned that cannot be observed directly. For example, if air is analyzed in a city, a common factor could be traffic. FA was originally developed for explaining psychological theories, for example, for description of factors such as intelligence or memory.

To transform the abstract factors determined in the first step into *interpretable* factors, rotation methods are applied. If definite target vectors can be assumed to be contained in the data, for example, a spectrum under a spectrochromatogram, the rotation of data is performed by using a target. This technique is known as *target-transform factor analysis* (*TTFA*, cf. Example 5.6).

In PCA, the objects are usually associated with samples or more generally with cases and features with properties of those cases. In contrast, in FA, the properties are arranged as the objects in the rows and the samples as features in the columns.

The general strategy of FA is again the decomposition of the data matrix, $\boldsymbol{X}$, in Eq. (5.1) into two smaller matrices $\boldsymbol{F}$ and $\boldsymbol{L}$:

$$\boldsymbol{X} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}} + \boldsymbol{E} \tag{5.26}$$

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ \vdots & & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1d} \\ f_{21} & f_{22} & & f_{2d} \\ \vdots & & & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nd} \end{pmatrix} \begin{pmatrix} l_{11} & l_{21} & \cdots & l_{p1} \\ l_{12} & l_{22} & & l_{p2} \\ \vdots & & & \vdots \\ l_{1d} & l_{2d} & \cdots & l_{pd} \end{pmatrix} + \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1p} \\ e_{21} & e_{22} & & e_{2p} \\ \vdots & & & \\ e_{n1} & e_{n2} & & e_{np} \end{pmatrix} \tag{5.27}$$

The scores matrix, $\boldsymbol{F}$, contains the values for the properties of the d causal factors. The columns of the matrix $\boldsymbol{L}$ characterize the fraction of the *loading* related to the considered factor. The matrix

$\boldsymbol{E}$ consists of the $p-d$ remaining or *specific* factors. These factors cannot be related to the sought common factors. Frequently, the specific factors represent random interferences, for example, noise factors.

Q and R Analysis The starting point in FA is, as in PCA, the covariance or correlation matrix. So far, only one possibility for computing this dispersion matrix has been exploited (Eqs. 5.11 and 5.14). In principle, the dispersion matrices can be calculated by multiplication of the data matrix by its transpose or by relating the transposed matrix with the original one.

Consider the calculation of the *variance–covariance matrix* of the scaled data of the $\boldsymbol{X}$ matrix:

$$\boldsymbol{C}_{\mathrm{Q}} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \quad \text{or} \quad \boldsymbol{C}_{\mathrm{R}} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \tag{5.28}$$

where $\boldsymbol{X}$ is a n^*p matrix (cf. Eq. (5.1)), and Q and R labels for the so-called Q and R analysis modes, respectively.

Computation of the covariance matrix based on Q mode has been tacitly used in Eqs. (5.9)–(5.11). A matrix, $\boldsymbol{C}_{\mathrm{Q}}$, of dimension p^*p is obtained. The covariance matrix for the R mode, $\boldsymbol{C}_{\mathrm{R}}$, is determined after transposing $\boldsymbol{X}$. Its dimension is n^*n.

For the correlation matrix, we obtain for the two modes

$$\boldsymbol{R}_{\mathrm{Q}} = (\boldsymbol{X}\boldsymbol{V}_{\mathrm{Q}})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{V}_{\mathrm{Q}}) \quad \text{or} \quad \boldsymbol{R}_{\mathrm{R}} = (\boldsymbol{V}_{\mathrm{R}}\boldsymbol{X})(\boldsymbol{V}_{\mathrm{R}}\boldsymbol{X})^{\mathrm{T}} \tag{5.29}$$

where

$$v_{\mathrm{Q}}(ij) = \frac{1}{\frac{1}{p-1}\sqrt{\sum_{i=1}^{p}(x_{ij}-\overline{x}_j)^2}} \quad v_{\mathrm{R}}(ij) = \frac{1}{\frac{1}{n-1}\sqrt{\sum_{i=1}^{n}(x_{ij}-\overline{x}_j)^2}} \tag{5.30}$$

Evaluations based on the Q or R mode are also needed in connection with other multivariate methods. In general, the technique is called R analysis when the relationship among p features determined by n observations is of interest. Q analysis examines the relationship among n objects characterized by p variables. For the most frequently used approach in FA, the interpretation of the loadings can be denoted as R analysis and, as seen as follows, the evaluation of the scores as Q analysis.

Communalities and Reduced Correlation Matrix As mentioned earlier, we distinguish in FA between common and specific factors. The criterion for this distinction is based on their loadings that are clearly different from zero. A common factor is found if at least two of its loadings are distinctly different from zero. For a specific factor, it is true that only one of the loadings $l_{1k} \ldots l_{pk}$ is clearly distinguished from zero. The subdivision of loadings, $\boldsymbol{L}'$,

in d common factors and $p-d$ specific factors can be expressed as sum of loadings of common factors, $\boldsymbol{L}''$, and of specific factors, $\boldsymbol{L}'''$:

$$\mathbf{L}' = \mathbf{L}'' + \mathbf{L}''' = \begin{pmatrix} l_{11} & l_{21} & \dots & l_{1d} & 0 & \dots & 0 \\ l_{21} & l_{22} & & l_{2d} & 0 & & 0 \\ & & & & & & \vdots \\ l_{p1} & l_{p2} & \dots & l_{pd} & 0 & \dots & 0 \end{pmatrix} + \begin{pmatrix} 0 & \dots & 0 & l_{1d+1} & 0 & \dots & 0 \\ 0 & & 0 & 0 & l_{2d+2} & & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & l_{pp} \end{pmatrix} \tag{5.31}$$

In the case of orthogonal factors, we obtain

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} + \boldsymbol{K}\boldsymbol{K}^{\mathrm{T}} \tag{5.32}$$

where

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & l_{21} & \dots & l_{1d} \\ l_{21} & l_{22} & & l_{2d} \\ \vdots & & & \vdots \\ l_{p1} & l_{p2} & \dots & l_{pd} \end{pmatrix} \quad \boldsymbol{K} = \begin{pmatrix} l_{1d+1} & 0 & \dots & 0 \\ 0 & l_{2d+2} & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & l_{pp} \end{pmatrix} \tag{5.33}$$

The squared elements of matrix $\boldsymbol{K}$ are exactly the specific variances.

$$\boldsymbol{K}^2 = \boldsymbol{K}\boldsymbol{K}^{\mathrm{T}} = \begin{pmatrix} l_{1d+1}^2 & 0 & \dots & 0 \\ 0 & l_{2d+2}^2 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & l_{pp}^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & \sigma_p^2 \end{pmatrix} \tag{5.34}$$

These variances correspond to the fractions of the standardized variance that can be explained by the common factors.

On the basis of the loading matrix, $\boldsymbol{L}$, for orthogonal factors, a reduced correlation matrix of the following form results:

$$\boldsymbol{R}' = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} = \boldsymbol{R} - \boldsymbol{K}\boldsymbol{K}^{\mathrm{T}} = \begin{pmatrix} h_1^2 & r_{12} & \dots & r_{1p} \\ r_{12} & h_2^2 & & r_{2p} \\ \vdots & & & \\ r_{1p} & r_{2p} & & h_p^2 \end{pmatrix} \tag{5.35}$$

where $h_j^2 = 1 - \sigma_j^2 = l_{j1}^2 + l_{j2}^2 + \dots + l_{jd}^2$ for $j = 1 \dots p$.

In Eq. (5.35), the quantities h_j^2 are the so-called *communalities*. They reflect which fraction of the variance of the jth standardized feature is explained by common factors. The communalities either are empirically known or can be estimated from random samples.

For correlated (skewed) factors, the communalities are derived from the multiple correlation coefficients according to

$$h_j^2 = 1 - \frac{1}{r_{jj}} \quad \text{for } j = 1 \ldots p \tag{5.36}$$

The first problem of an FA is that of estimation of the loading matrix, $\boldsymbol{L}$. We will consider only two methods here, that is, PCA and principal factor analysis.

Estimation of Factor Loadings FA in a more narrow sense means procedures where the *reduced* correlation matrix, $\boldsymbol{R'}$, Eq. (5.35) is reproduced. For that, the communalities are required. The reproduction of the original correlation matrix $\boldsymbol{R}$ (Eq. (5.14)), as done with PCA, is not an FA in a restrictive sense. However, because PCA has been discussed already above, we will use it here as the basis for performing a FA.

Principal Component Analysis The loading matrix has to reproduce the correlation matrix $\boldsymbol{R}$, that is,

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \tag{5.37}$$

The assumptions on specific factors that must be made in FA in the more narrow sense are not made in this context. The factorial model reads in the terminology of FA (cf. Eq. (5.15)):

$$\boldsymbol{X} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}} \tag{5.38}$$

As discussed earlier, PCA is not usually carried out to derive interpretable components. But complicated relationships should be reduced to simple ones by projecting the data from multidimensional space to two or three dimensions.

For this, the principal axes are translated and rotated without changing the distances of the features relative to each other. The pairwise, perpendicularly arranged coordinates remain orthogonal. Mathematically, this transformation can be dealt with by solving an eigenvalue problem. The loading vector $\boldsymbol{l}_k$ of the kth component corresponds to the normalized eigenvector of the related k largest eigenvalue λ_k of the empirical correlation matrix $\boldsymbol{R}$, that is,

$$\boldsymbol{l}_k^{\mathrm{T}}\boldsymbol{l}_k = 1 \tag{5.39}$$

Eigenvalue analysis is described here in more detail for a better understanding.

Eigenvector Analysis For a symmetric, real matrix, $\boldsymbol{R}$, an eigenvector $\boldsymbol{v}$ is obtained by

$$\boldsymbol{R}\boldsymbol{v} = \boldsymbol{v}\lambda \tag{5.40}$$

where λ is an unknown scalar: the *eigenvalue.* The eigenvector is to be determined such that the vector $\boldsymbol{Rv}$ is proportional to $\boldsymbol{v}$. For this, Eq. (5.40) is rewritten as

$$\boldsymbol{Rv} - \boldsymbol{v}\lambda = 0 \quad \text{or} \quad (\boldsymbol{R} - \lambda \boldsymbol{I})\boldsymbol{v} = 0 \tag{5.41}$$

where $\boldsymbol{I}$ is the identity matrix and the vector, $\boldsymbol{v}$, is orthogonal to all of the row vectors of matrix $(\boldsymbol{R} - \lambda \boldsymbol{I})$. The equation obtained is equivalent to a set of d where d is the rank of $\boldsymbol{R}$.

Unless $\boldsymbol{v}$ is the null vector, Eq. (5.41) only holds if

$$\boldsymbol{R} - \lambda \boldsymbol{I} = 0 \tag{5.42}$$

A solution to this set of equations only exists if the determinant on the left side of Eq. (5.42) is zero:

$$|\boldsymbol{R} - \lambda \boldsymbol{I}| = 0 \tag{5.43}$$

Computing the determinant reveals a polynomial in λ of degree d. Then the roots, λ_i with $i = 1 \ldots d$, of those equations have to be found. There will be an associated vector $\boldsymbol{v}_i$, such that:

$$\boldsymbol{Rv}_i - \boldsymbol{v}_i \lambda_i = 0 \tag{5.44}$$

or in matrix notation:

$$\boldsymbol{RV} = \boldsymbol{V\Lambda} \tag{5.45}$$

The matrix $\boldsymbol{V}$ is a square and orthogonal matrix of the eigenvectors. The different ways of computing the dispersion matrix by Q or R analysis techniques lead to different sets of eigenvalues, as we will see next in the comparison with SVD.

Example 5.3 *Eigenvalue Determination*

As an example of an eigenvalue analysis, we use the following data matrix consisting of three rows and two columns:

$$\begin{pmatrix} 0.9 & 4.1 \\ 1.9 & 2.9 \\ 2.9 & 2.1 \end{pmatrix}$$

The autoscaled form of this matrix is

$$\boldsymbol{X} = \begin{pmatrix} -1 & 1.059 \\ 0 & -0.132 \\ 1 & -0.927 \end{pmatrix} \tag{5.46}$$

For determination of the eigenvalues, Eqs. 5.40–5.45 are applied. First, the correlation matrix is calculated:

$$\boldsymbol{R} = \begin{pmatrix} 1 & -0.993 \\ -0.993 & 1 \end{pmatrix} \tag{5.47}$$

Insertion of the eigenvalues, λ, into Eq. (5.43) leads to

$$|\boldsymbol{R} - \lambda \boldsymbol{I}| = \begin{vmatrix} 1-\lambda & -0.993 \\ -0.993 & 1-\lambda \end{vmatrix} = 0 \tag{5.48}$$

Calculation of the determinant results in the characteristic polynomial for the root λ. In our example, a squared equation is obtained:

$$(1 \ -\lambda)^2 - 1.993 = 0 \tag{5.49}$$

The solution for Eq. (5.49) reveals the two eigenvalues $\lambda_1 = 1.993$ and $\lambda_2 = 0.115$.

For determination of the eigenvectors according to Eq. (5.44), we insert both eigenvalues into Eq. (5.48).

Calculation of the first eigenvector:

$$\begin{pmatrix} 1-1.993 & -0.993 \\ -0.993 & 1-1.993 \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.50}$$

Evaluation of Eq. (5.50) results in the following equation system with two unknowns:

$$\begin{aligned} -0.993v_{11} - 0.993v_{21} &= 0 \\ -0.993v_{11} - 0.993v_{21} &= 0 \end{aligned} \tag{5.51}$$

Further simplification leads to

$$v_{11} = -v_{21} \tag{5.52}$$

For the eigenvectors in Eq. (5.44), infinite solutions exist. For example, assume $v_{11} = 1$, then as a consequence, $v_{21} = -1$. Usually, the eigenvectors are normalized to length 1, that is,

$$v_{11}^2 + v_{21}^2 = 1 \tag{5.53}$$

The predefined values of the eigenvectors are then divided by the size of $\sqrt{1^2 + 1^2} = \sqrt{2}$. For the first eigenvector, we obtain

$$\begin{pmatrix} v_{11} \\ v_{21} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0.707 \\ -0.707 \end{pmatrix} \tag{5.54}$$

Analogous calculation of the second eigenvector reveals

$$\begin{pmatrix} 1-0.115 & -0.993 \\ -0.993 & 1-0.115 \end{pmatrix} \begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.55}$$

$$\begin{aligned} 0.885v_{12} - 0.993v_{22} &= 0 \\ -0.993v_{12} + 0.885v_{22} &= 0 \end{aligned} \tag{5.56}$$

$$v_{12} = v_{22} \tag{5.57}$$

$$v_{12}^2 + v_{22}^2 = 1 \tag{5.58}$$

The second eigenvector is then

$$\begin{pmatrix} v_{12} \\ v_{22} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0.707 \\ 0.707 \end{pmatrix} \tag{5.59}$$

The solution for the matrix in Eq. (5.45) is then

$$\begin{aligned} \boldsymbol{RV} = \boldsymbol{V\Lambda} &= \begin{pmatrix} 1 & -0.993 \\ -0.993 & 1 \end{pmatrix} \begin{pmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{pmatrix} \\ &= \begin{pmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{pmatrix} \begin{pmatrix} 1.993 & 0 \\ 0 & 0.115 \end{pmatrix} \end{aligned} \tag{5.60}$$

The matrix of eigenvectors $\boldsymbol{V}$ directly provides the loading matrix $\boldsymbol{L}$ in Eq. (5.38). In the next step, the matrix of scores $\boldsymbol{F}$ is determined, as explained in the following text in general. In Example 5.4, the score matrix will be computed for the data matrix $\boldsymbol{X}$ in Eq. (5.46).

Principal Factor Analysis PCA can be slightly modified to be applied as a factor analytical method in its genuine sense, if the specific variances are included. For this, the communalities $h_1^2 \ldots h_p^2$ of the p features (cf. Eq. (5.36)) are estimated from the correlation matrix $\boldsymbol{R}$, and the reduced correlation matrix $\boldsymbol{R'}$ is reproduced according to Eq. (5.35) from the loadings:

$$\boldsymbol{R'} = \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}} \tag{5.61}$$

The reduced correlation matrix is subsequently subjected to a PCA: the eigenvalues are determined and normalized to length 1, as explained in Example 5.3. The significant eigenvectors then determine the loading matrix $\boldsymbol{L}$. This approach is termed *principal factor analysis*.

More powerful factor analytical methods are the centroid method, maximum-likelihood method, and canonical FA. With those methods, the loading matrix is estimated iteratively. Details can be learned from the available software and the statistical literature [1].

Determination of the Scores Apart from the loadings for judging the objects, the scores are also important. Unique scores can be only determined on the basis of a complete PCA, since the loading matrix is orthonormal, which means

$$\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} = \boldsymbol{I} \tag{5.62}$$

The scores matrix can be directly estimated from the model for the standardized data matrix, $\boldsymbol{X}$, since

$$\boldsymbol{X} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}} \tag{5.63}$$

(in more detail $\boldsymbol{X}\boldsymbol{L} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L} = \boldsymbol{F}\boldsymbol{I} = \boldsymbol{F}$).

For FA in its strict sense, no unique scores matrix can be given, because the loading matrix only reproduces the reduced correlation matrix of the features. Therefore, the score matrix $\boldsymbol{F}$ has to be estimated. The estimation methods are to be found in the specialized literature.

Example 5.4 *Determination of the Factor Score Matrix*

For the data in Example 5.3, the matrix of factor scores $\boldsymbol{F}$ is to be computed.

To obtain that score matrix, the values for the standardized matrix $\boldsymbol{X}$ (Eq. (5.46)) and the loading matrix $\boldsymbol{L}$ or $\boldsymbol{V}$ computed in Example 5.3 (Eq. (5.60)) are inserted into Eq. (5.63):

$$\begin{aligned}\boldsymbol{F} = \boldsymbol{X}\boldsymbol{L} &= \begin{pmatrix} -1 & 1.059 \\ 0 & -0.132 \\ 1 & -0.927 \end{pmatrix} \begin{pmatrix} 0.707 & 0.707 \\ -0.707 & 0.707 \end{pmatrix} \\ &= \begin{pmatrix} -1.455 & 0.0422 \\ 0.095 & -0.0937 \\ 1.363 & 0.0515 \end{pmatrix}\end{aligned} \tag{5.64}$$

Decomposition of the standardized data matrix in Eq. (5.46) by PCA results in the following factor analytical model:

$$\begin{aligned}\boldsymbol{X} &= \begin{pmatrix} -1 & 1.059 \\ 0 & -0.132 \\ 1 & -0.927 \end{pmatrix} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}} \\ &= \begin{pmatrix} -1.455 & 0.0422 \\ 0.095 & -0.0937 \\ 1.363 & 0.0515 \end{pmatrix} \begin{pmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{pmatrix}\end{aligned} \tag{5.65}$$

Relationship between FA and SVD If the FA is performed on the basis of PCA, a direct relationship to SVD can be derived (cf. Eq. (5.2)).

The matrix $\boldsymbol{U}$ represents the eigenvector of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ (left eigenvector), and the matrix $\boldsymbol{V}$ the eigenvector of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ (right eigenvector).

Denote the matrix of the eigenvectors that were determined from the correlation matrix by Q analysis, $\boldsymbol{R}_{\mathrm{Q}}$, by $\boldsymbol{U}$, and the eigenvector matrix of R analysis, $\boldsymbol{R}_{\mathrm{R}}$, by $\boldsymbol{V}$, then decomposition of the matrix $\boldsymbol{X}$ by means of SVD reveals

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^{\mathrm{T}} \tag{5.66}$$

Depending on the mode, that is, whether Q or R analysis has been used for computation of Δ, the dimension of matrix Δ is either n^*n or p^*p, respectively.

The factor analytical solutions after R and Q analysis are interrelated as follows:

$$\boldsymbol{X} = \boldsymbol{L}_{\mathrm{Q}}\boldsymbol{F}_{\mathrm{Q}} \quad \text{or} \quad \boldsymbol{X} = \boldsymbol{F}_{\mathrm{R}}\boldsymbol{L}_{\mathrm{R}} \tag{5.67}$$

where

$$\boldsymbol{X} = \overbrace{\boldsymbol{U}}^{\boldsymbol{L}_{\mathrm{Q}}}\overbrace{\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^{\mathrm{T}}}^{\boldsymbol{F}_{\mathrm{Q}}}\boldsymbol{X} = \overbrace{\boldsymbol{U}\boldsymbol{\Lambda}^{1/2}}^{\boldsymbol{F}_{\mathrm{R}}}\overbrace{\boldsymbol{V}^{\mathrm{T}}}^{\boldsymbol{L}_{\mathrm{R}}} \tag{5.68}$$

The direction of the solution is less dependent on the mode, Q or R analysis, but is directed by the scaling procedures applied. The scaling may be based on scaling the columns or rows or on the scaling of both.

Example 5.5 *FA by R Mode*

Eigenvalue analysis in Example 5.3 was carried out by R analysis, that is, the matrix of scores $\boldsymbol{F}_{\mathrm{R}}$ (Eq. (5.64)) is formed according to Eq. (5.68) by the product of matrix $\boldsymbol{U}$ from SVD and the matrix of the eigenvalues (Eq. (5.49)):

$$\boldsymbol{F}_R = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2} = \begin{pmatrix} -0.729 & 0.367 \\ 0.0476 & -0.815 \\ 0.683 & 0.448 \end{pmatrix}\begin{pmatrix} 1.993 & 0 \\ 0 & 0.115 \end{pmatrix} = \begin{pmatrix} -1.455 & 0.0422 \\ 0.095 & -0.0937 \\ 1.363 & 0.0515 \end{pmatrix} \tag{5.69}$$

Compare the results to that in Eq. (5.64).

Determination of the Number of Significant Factors To decide on the number of common factors, some criteria have been introduced already in connection with PCA, such as the eigenvalue-one test or the scree test. Those criteria can be overtaken in FA, if the determination of loadings is performed by a PCA.

For FA in its genuine sense, additional criteria are used for rank analysis, that is, for determination of the number of significant factors. Most frequently, an empirical indicator function, *IND*, introduced by Malinowski [2] is used. It is computed from the real error, *RE*, or the residual standard deviation, *RSD*, as follows:

$$IND = \frac{RE}{(p-d)^2} = \frac{RSD}{(p-d)^2} = \frac{\left(\sum_{k=d+1}^{p} \frac{\lambda_k}{n\,(p-d)}\right)^{1/2}}{(p-d)^2} \quad (5.70)$$

where p is the number of features, d the number of common factors, n the number of objects, and λ_k the kth eigenvalue. The indicator function has a minimum at the most probable number of factors.

Rotation Methods An optimal loading matrix is obtained by rotation of factors. One distinguishes orthogonal and oblique (correlated) rotations. In the case of an orthogonal rotation, the coordinate system is rotated. The aim is that the new coordinate axis cut the swarm of points in an optimal way. This can be often better achieved by an oblique rotation. If the data can be described by an orthogonal rotation in an optimal way, then an oblique method will also lead to coordinate axes that are perpendicular to each other.

In the case of an orthogonal rotation of the loading matrix $\boldsymbol{L}$, a rotated loading matrix $\boldsymbol{L}_{\text{rot}}$ is obtained by multiplication of that matrix with a transformation matrix $\boldsymbol{T}$:

$$\boldsymbol{L}_{\text{rot}} = \boldsymbol{L}\boldsymbol{T} \quad (5.71)$$

For an oblique rotation by multiplication with a transformation matrix, a matrix of the so-called factor structure, $\boldsymbol{L}_{\text{fst}}$ is obtained. This matrix contains correlation of common factors and features:

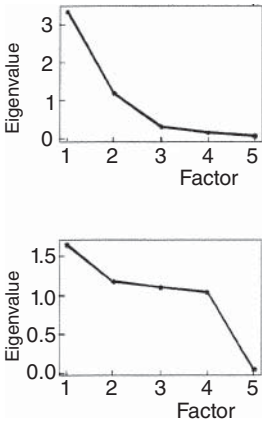
$$\boldsymbol{L}_{\text{fst}} = \boldsymbol{L}\boldsymbol{T} \quad (5.72)$$

The rotated loading matrix is only obtained after multiplication of $\boldsymbol{L}_{\text{fst}}$ by the inverse correlation matrix of factors:

$$\boldsymbol{L}_{\text{rot}} = \boldsymbol{L}_{\text{fst}}\boldsymbol{R}_{\text{F}}^{-1} \quad (5.73)$$

As examples for orthogonal and oblique factor rotations, the varimax, quartimax, and oblimax criteria will be considered.

The *varimax criterion* serves the purpose of an orthogonal rotation, where the variance of the squared loadings within a common *factor* is maximized. As a result, as many common factors should be retained that are described by as few features (variables) as possible. Large eigenvalues and loadings are increased, but small ones



Eigenvalues without (*top*) and with rotation after the varimax criterion (*bottom*) for the hair data in Table 5.1.

are decreased. The uniform weighting of the variables is guaranteed by scaling the loadings by means of the communalities (cf. Eq. (5.35)). The varimax criterion is calculated by

$$\max[\mathrm{V}] = p\sum_{j=1}^{p}\sum_{k=1}^{d}\sigma_{jk}^2 - \sum_{k=1}^{d}\left(\sum_{j=1}^{p}\sigma_{jk}^2\right)^2 \tag{5.74}$$

where p is the number of features and d the number of factors.

In contrast, the *quartimax criterion* maximizes the variance of the (squared) loadings of a *variable.* The aim is to express each variable by as few common factors as possible. This kind of rotation leads to an increase of large loadings and a decrease of small loadings of each variable. However, it might happen that a single common factor emerges from the rotation. The quartimax criterion is obtained from the communalities as follows:

$$\max[\mathrm{Q}] = \sum_{k=1}^{d}\sum_{j=1}^{p}\sigma_{jk}^4 \tag{5.75}$$

In the case of the *oblimax criterion*, a similar function as known for the quartimax criterion is maximized, termed the *kurtosis function*:

$$\max[\mathrm{K}] = \frac{\sum_{k=1}^{d}\sum_{j=1}^{p}\sigma_{jk}^4}{\left[\sum_{k=1}^{d}\sum_{j=1}^{p}\sigma_{jk}^4\right]^2} \tag{5.76}$$

More criteria for rotation of factors are given, for example, in Frank and Todeschini [3].

Target Rotation: Target Transformation Factor Analysis A special kind of rotation matches the abstract factor loadings to known patterns. For example, whether a hypothetical spectrum under an incompletely resolved chromatographic peak can be found is tested on the basis of a spectrochromatogram, or whether a compound pattern can be detected in the exhaust gas of an emittent. The hypothetical spectrum or the elemental pattern is termed the *target*, $\boldsymbol{L}^*$. The containment of a target in a data matrix is evaluated by computing the target transformation matrix, $\boldsymbol{T}$.

The starting point is the model of FA in Eq. (5.26):

$$\boldsymbol{X} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}}$$

The abstract loadings, $\boldsymbol{L}$, are tested against a hypothetical loading vector, $\boldsymbol{L}^*$, that is, against the targets. This relationship is given on the basis of the transformation matrix $\boldsymbol{T}$ by

$$\boldsymbol{L}\boldsymbol{T} = \boldsymbol{L}^* \tag{5.77}$$

Computation of the transformation matrix is feasible by multiple linear regression analysis (cf. Section 6.2):

$$\boldsymbol{T} = (\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L})^{-1}\boldsymbol{L}^{\mathrm{T}}\boldsymbol{L}^{*} \tag{5.78}$$

The fit of the target vectors in matrix $\boldsymbol{L}^{*}$ and the predicted vectors in matrix $\widehat{\boldsymbol{L}}$ based on the transformation matrix according to Eq. (5.77) is usually estimated by the average relative deviation as follows:

$$\text{relative deviation} = \frac{\sum_{j=1}^{p} |l_j^{*} - \widehat{l}_j|}{\sum_{j=1}^{p} |l_j^{*}|} \tag{5.79}$$

Let us consider the different steps of a target transformation factor analysis, for example, data from the combination of liquid chromatography with ultraviolet (UV) spectroscopy.

Example 5.6 *Target Transformation Factor Analysis*

In a HPLC chromatogram under an incompletely resolved peak, polycyclic aromatic hydrocarbons (PAHs) are expected. Since detection was based on a diode array detector, the observed peak can be evaluated at several wavelengths. The spectrochromatogram for five wavelengths and seven retention times is shown in Figure 5.8. The corresponding data are given in Table 5.3.

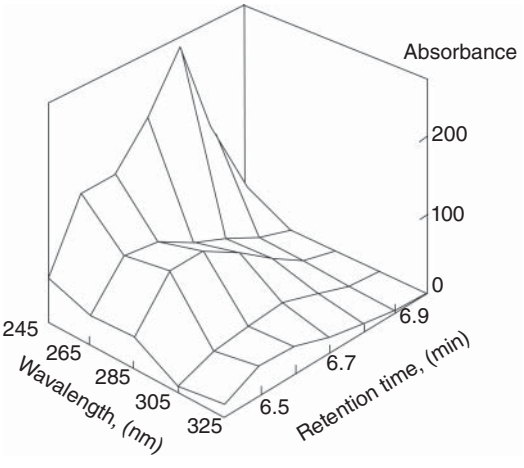


Figure 5.8 Simplified spectrochromatogram for an incompletely resolved peak in HPLC with diode array detection. Absorbance is given in milliabsorbance units.

Table 5.3 Absorbances in milliabsorbance units for the spectrochromatogram in Figure 5.8.

Retention time (min)	Wavelength (nm)				
	245	265	285	305	325
6.4	7.81	4.83	4.367	0.944	1.775
6.5	84.33	52.69	56.100	12.890	20.730
6.6	161.58	99.30	108.430	26.920	39.026
6.7	173.33	77.89	97.260	39.368	28.670
6.8	274.70	63.92	82.160	47.150	20.060
6.9	218.92	36.95	39.820	25.580	10.490
7.0	79.04	12.07	10.580	6.536	3.230

In the first step, we perform an FA based on the data for the spectrochromatogram in Table 5.3. The result is three significant common factors with the following scores and loadings:

$$\boldsymbol{X} = \boldsymbol{F}\boldsymbol{L}^{\mathrm{T}} = \begin{pmatrix} 1.431 & -0.375 & 0.839 \\ 0.183 & -0.889 & -0.266 \\ -1.077 & -1.302 & -0.977 \\ -0.881 & -0.186 & 1.509 \\ -0.868 & 1.387 & 0.480 \\ 0.110 & 1.175 & -1.297 \\ 1.102 & 0.191 & -0.286 \end{pmatrix} \times \begin{pmatrix} -0.779 & -0.958 & -0.979 & -0.892 & -0.905 \\ 0.606 & -0.282 & -0.195 & 0.417 & -0.423 \\ -0.282 & -0.050 & 0.058 & 0.177 & -0.047 \end{pmatrix} \tag{5.80}$$

To rotate the loading matrix, the target spectra of the hypothetical compounds benzo[*k*]fluoranthene (B[*k*]F), benzo[*b*]fluoranthene (B[*b*]F), perylene, and anthracene are tested. These spectra are illustrated for the considered wavelength in Figure 5.9.

On the basis of the computed loadings (Eq. (5.80)), we obtain the following matrix equation for the transformation matrix $\boldsymbol{T}$:

$$\boldsymbol{L}\boldsymbol{T} = \boldsymbol{L}^{*} = \begin{pmatrix} -0.779 & 0.606 & -0.282 & 1 \\ -0.958 & -0.282 & -0.049 & 1 \\ -0.979 & -0.195 & 0.058 & 1 \\ -0.892 & 0.417 & 0.177 & 1 \\ -0.905 & -0.423 & -0.047 & 1 \end{pmatrix}$$

$$\boldsymbol{T} = \begin{pmatrix} 111.2 & 112.6 & 282.1 & 280.0 \\ 38.2 & 87.2 & 76.4 & 2.25 \\ 52.5 & 69.4 & 12.2 & 1.0 \\ 110.6 & 33.2 & 5.1 & 1.3 \\ 14.7 & 25.0 & 6.9 & 5.5 \end{pmatrix} \tag{5.81}$$

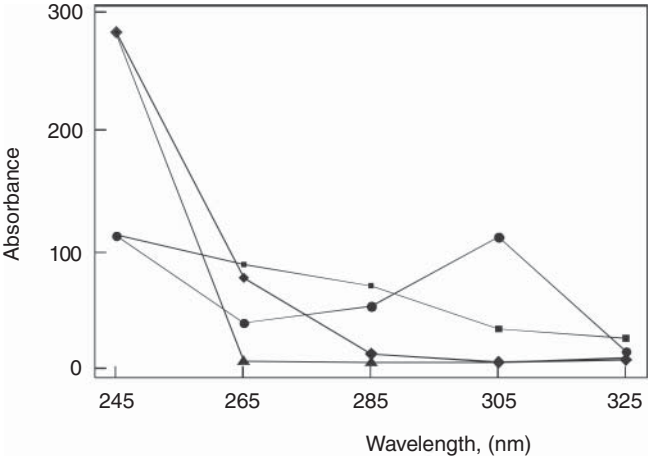


Figure 5.9 Simplified UV spectra of the compounds benzo[*k*]fluoranthene (●), benzo[*b*]fluoranthene (■), perylene (◆), and anthracene (▲) as the targets.

Note that the features (wavelengths) in the loading matrix $\boldsymbol{L}$ are arranged in the rows. The target spectra are given in the hypothetical loading matrix $\boldsymbol{L}^*$ columnwise in the order of the compounds B[*k*]F, B[*b*]F, perylene, and anthracene.

The values correspond to the absorbances shown in Figure 5.9. Since in our case the spectra need not be centralized, we have to add a column of 1s in the loading matrix. Then the intercepts for the spectra on the ordinate can be accounted for in the calculation of the transformation matrix.

The transformation matrix is computed by means of multiple linear regression according to Eq. (5.78):

$$\boldsymbol{T} = \begin{pmatrix} -141.1 & -782.5 & -589.5 & 769.1 \\ 112.5 & 134.5 & 242.7 & 111.2 \\ 14.57 & -423.5 & -888.3 & -249.7 \\ -64.69 & -645.9 & -465.2 & 748.4 \end{pmatrix} \tag{5.82}$$

Prediction of the target spectra by multiplication of the loading matrix with the transformation matrix reveals

$$\hat{\boldsymbol{L}} = \boldsymbol{LT} = \begin{pmatrix} 111.2 & 112.6 & 282.4 & 256.2 \\ 38.1 & 87.0 & 75.3 & -7.5 \\ 52.4 & 69.6 & 13.4 & -40.7 \\ 110.6 & 33.1 & 4.7 & 64.9 \\ 14.7 & 25.0 & 7.1 & 17.2 \end{pmatrix} \tag{5.83}$$

Comparison of the predicted matrix in Eq. (5.83) with the hypothetical loading matrix in Eq. (5.81) demonstrates that the spectra from the first three columns of the compounds B[*k*]F, B[*b*]F, and perylene fit quite well. The compounds can be clearly identified under the incompletely resolved peak, but for the spectrum of anthracene (fourth column in the loading matrix), an obvious deviation between the hypothetical and predicted loading vectors is observed (cf. Figure 5.10). Therefore, this compound can be excluded.

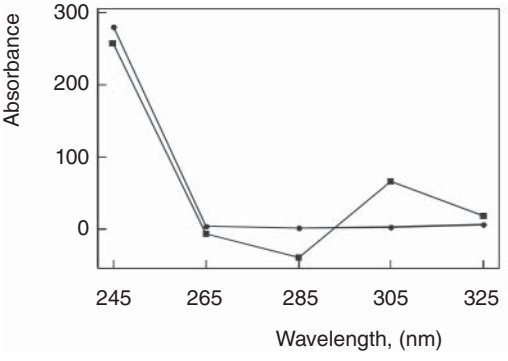


Figure 5.10 Comparison of the target spectrum anthracene (●) with the predicted spectrum (■) in Eq. (5.83).

Apart from transformations of spectra, elution profiles in chromatography, depth profiles in materials analysis, or elemental patterns in environmental analysis may also be interesting targets.

Evolving Factor Analysis (EVA) This method is a further development of FA, where intrinsically ordered data can be investigated. Multiwavelength detection of the elution of compounds in a chromatogram in dependence on time is a typical example, and the spectroscopic investigation of simultaneous equilibria in dependence on the pH value can also be carried out by evolving factor analysis (EVA).

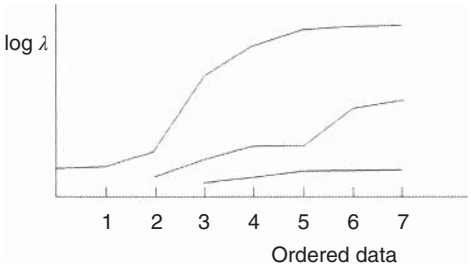


Figure 5.11 Schematic representation of evolving eigenvalues, λ, in dependence on intrinsically ordered data.

In the case of forward decomposition of the original data matrix, the eigenvalues in EVA are computed by stepwise addition of objects. The evolving eigenvalues reveal the existence of individual components (cf. Figure 5.11). In this way, for example, the intervals for the elution of compounds in a chromatogram can be evaluated.

Independent Component Analysis

The first factorial method we considered in this chapter was PCA. Although this is the most common projection method in multivariate analysis, PCA could not solve what is known as the *blind source separation problem*. Imagine a cocktail party where you record the mixed signals of the party guest speeches. To estimate the original speeches, ICA has to be used instead of PCA.

The $n \times p$ matrix of observations, $\boldsymbol{X}$, is considered to originate from the underlying $p \times m$ matrix of signal sources, $\boldsymbol{S}$, the independent components, and an $n \times m$ mixing matrix, $\boldsymbol{A}$, in a noise-free ICA model according to

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}^{\mathrm{T}} \tag{5.84}$$

There should be at least as many observations as there are source signals, that is, $n \geq m$. For the ICA model, it is then assumed that the source signals in $\boldsymbol{S}$ are statistically independent and non-Gaussian rather than uncorrelated. If the number of source signals is equal to the number of observed signals ($n = m$), estimation of the true, underlying sources, $\boldsymbol{U}$, as well as the demixing matrix $\boldsymbol{W} = \boldsymbol{A}^{-1}$ is performed by

$$\boldsymbol{U} = \boldsymbol{W}\boldsymbol{X} \tag{5.85}$$

For Gaussian underlying sources, ICA is of no use and PCA would be the method of choice. Computation of the independent components is based on entropy or negentropy measures (cf. [8]) based on the fact that, among random variables with equal variance, Gaussian variables have the maximum entropy.

The performance of ICA in case of recovering sensor signals from mixed signal observations is demonstrated in Example 5.7.

Example 5.7 *Independent Component Analysis (ICA) of Data from Three Sensor Signals*

Three source signals, ***S,*** (Figure 5.12a) have been mixed by an arbitrary mixing matrix, ***A,*** revealing the three mixed signals, ***X,*** in Figure 5.12b.

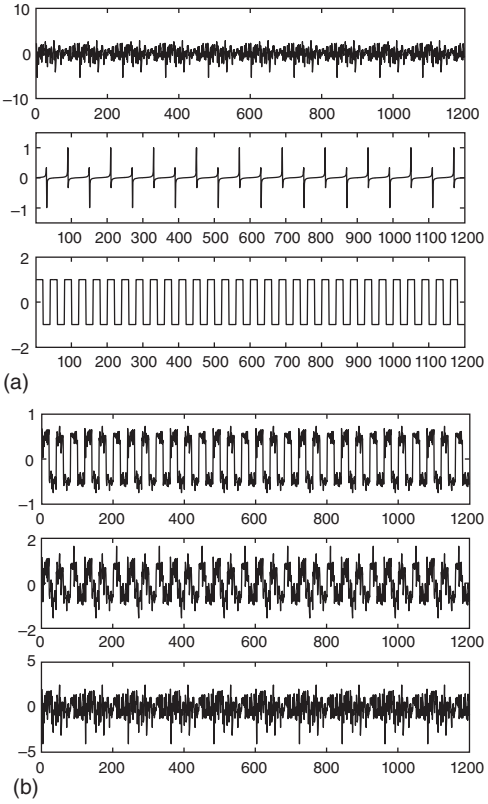


Figure 5.12 Source signals (a) and mixed signals (b).

The mixed signals were then used to estimate the original source signal traces by PCA and ICA as given in Figures 5.13a and 5.13b, respectively.

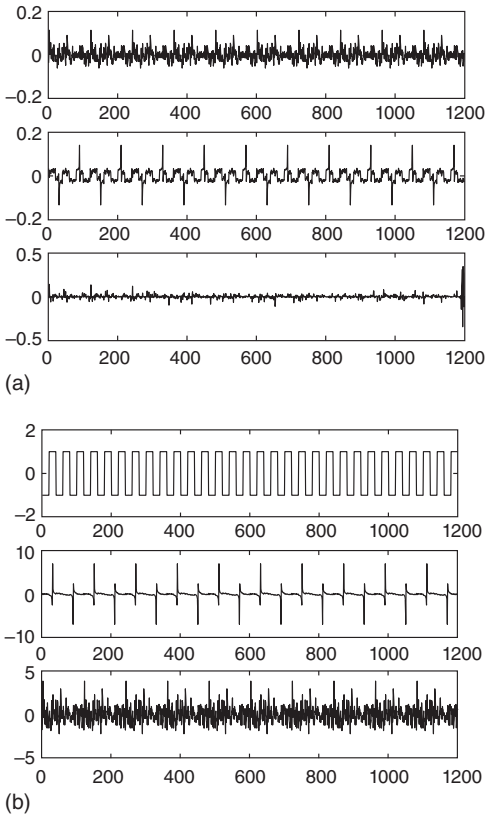


Figure 5.13 Estimation of signals by PCA (a) and ICA (b).

As a result, the PCA loadings (Figure 5.13a) reflect some more or less distorted signals of the sources, whereas ICA (Figure 5.13b) recovers the signal traces correctly.

Multiway Decompositions

The factorial models considered in this chapter so far are useful to decompose two-way data, that is, a data matrix $\boldsymbol{X}$ consisting of I rows and J columns. Typically, a set of I samples with their optical spectra measured at J wavelengths provide this kind of data. Imagine a fluorescence emission/excitation spectrum recorded for I samples. If the fluorescence spectra are measured at J emission wavelengths and K excitation wavelengths, the dimensions of the

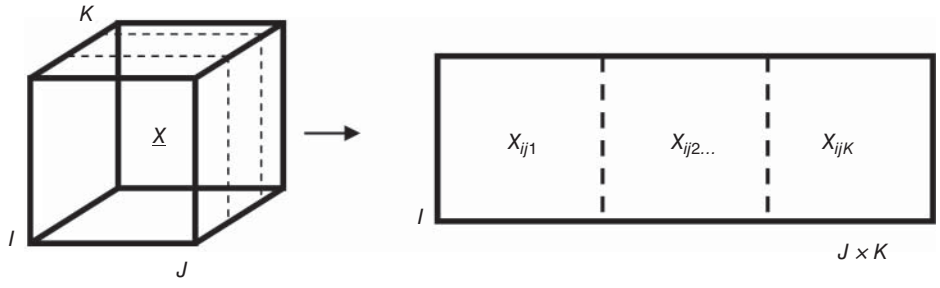


Figure 5.14 Unfolding of three-way arrays for data analysis by conventional factorial methods.

resulting data matrix $\underline{\boldsymbol{X}}$ are now $I \times J \times K$ representing a three-way array. Hyphenation of chromatography and spectroscopy leads to even higher dimensions of multiway data sets. For example, I samples applied to two-dimensional GC-MS generate a four-way data array.

In principle, three-way arrays can be unfolded (matricized) as seen in Figure 5.14 and then treated by the aforementioned factorial methods. Matricizing often results in overfitting the $\boldsymbol{X}$ matrix and simultaneously in a bad fit after cross-validation. Furthermore, the predictive power is known to be worse than by applying N-way methods.

Direct analysis of a three-way data array is feasible by parallel factor analysis (PARAFAC) or by Tucker models.

Parallel Factor Analysis The PARAFAC model for an element x_{ijk} of the three-way array $\underline{\boldsymbol{X}}$ $(I \times J \times K)$ in Figure 5.15 is as follows:

$$x_{ijk} = \sum_{f=1}^{F} a_{if} b_{jf} c_{kf} + e_{ijk} \tag{5.86}$$

where a_{if}, b_{jf}, and c_{kf} are the elements of the matrices of modes ***A, B,*** and ***C;*** e_{ijk} represents the residual; and F is the number of factors. Figure 5.15 depicts the model graphically.

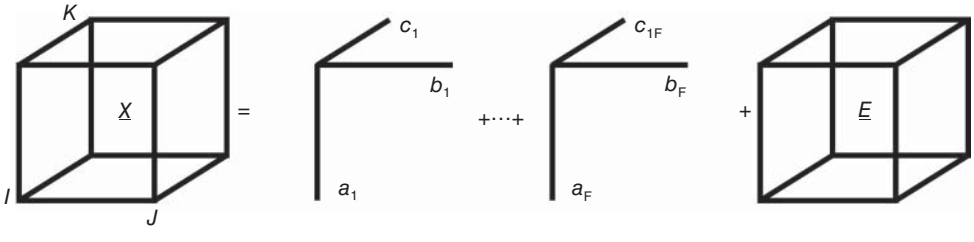


Figure 5.15 The parallel factor analysis (PARAFAC) model for F factors.

In terms of matrices $\boldsymbol{X}_k$, where $\boldsymbol{X}_k$ is the kth $(I \times J)$ slice of the three-way $\underline{\boldsymbol{X}}$ $(I \times J \times K)$, the PARAFAC model can be written by the matrices $\boldsymbol{A}$ $(I \times F)$, $\boldsymbol{B}$ $(J \times F)$, and $\boldsymbol{C}$ $(K \times F)$ that contain the elements a_{iF}, b_{jF}, and c_{kF}

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{B}^{\mathrm{T}} + \boldsymbol{E}_k = c_{k1}\boldsymbol{a}_1\boldsymbol{b}_1^{\mathrm{T}} + c_{k2}\boldsymbol{a}_2\boldsymbol{b}_2^{\mathrm{T}} + \cdots + c_{k\mathrm{F}}\boldsymbol{a}_{\mathrm{F}}\boldsymbol{b}_{\mathrm{F}}^{\mathrm{T}} + \boldsymbol{E}_k \tag{5.87}$$

where $\boldsymbol{D}_k$ is a diagonal matrix containing the kth row of $\boldsymbol{C}$ on its diagonal, that is, the elements c_{k1} to $c_{k\mathrm{F}}$; $\boldsymbol{a}$ and $\boldsymbol{b}$ represent the fth column of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively; and $\boldsymbol{E}_k$ is the residual matrix with the same dimension as $\boldsymbol{X}_k$.

The PARAFAC model can also be considered a *trilinear model.* Two sets of parameters are fixed, for example, the a's and b's, and the element x_{ijk} is estimated as a linear function of the remaining parameter c.

Estimation of the parameters $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ is usually carried out by the alternating least squares (ALS) algorithm. As an example for PARAFAC decompositions, we consider the evaluation of the folding states of a protein by means of mass spectrometry (Example 5.8).

Example 5.8 *PARAFAC Decomposition of Protein Mass Spectra*

Electrospray–ionization mass spectrometry (ESI-MS) provides information on the folding of a protein. Figure 5.16 shows an ESI-MS spectrum of the intact protein *myoglobin.* The different peaks represent differently charged protein states. As a general rule, the more charged states that are present in the spectrum, the less folded is the protein. The folding states are highly pH dependent, and this is studied in the example by varying the pH from 2.7 to 7. For a given sample, the mass spectrum in dependence on all pH values constitutes a matrix as seen in Figure 5.17. For different samples, a third dimension provides a three-way data array $\underline{\boldsymbol{X}}$ of order 18 (charge states) × 10 (pH values) × 5 (samples of different concentrations). In order to find the number of folding states and the pH ranges of their existence, the array is decomposed here by PARAFAC.

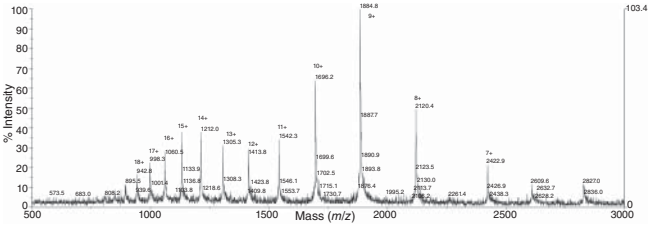


Figure 5.16 Electrospray ionization–mass spectrometry (ESI-MS) spectrum of apomyoglobin.

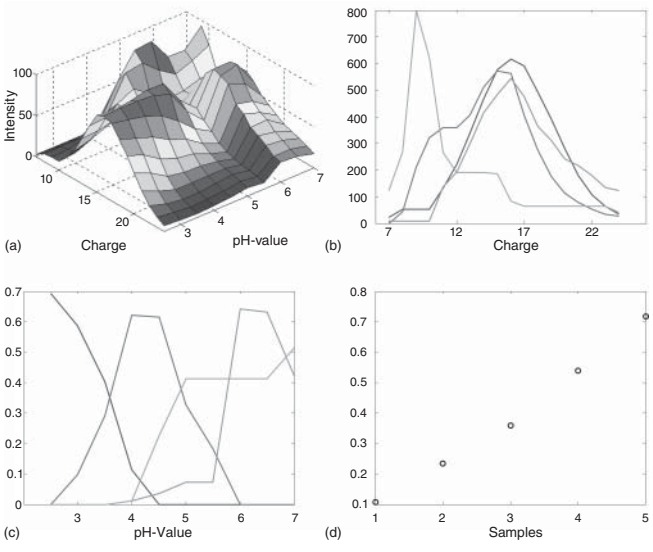


Figure 5.17 Charge states for a single sample (a) derived from ESI-MS spectra in dependence on the pH value of solution. The four factors found after decomposing the array $\underline{\boldsymbol{X}}$ into modes *A* (b), *B* (c), and *C* (d) by PARAFAC is also given.

Results of the decomposition are given in Figure 5.17 for four factors of the three modes. Mode A describes the factors with respect to the charge states, mode B those of pH dependence, and mode C shows the five samples varying in the concentration of the protein. To identify the different folding states, the decomposition is plotted as the product of modes A and B for the four factors. This is depicted in Figure 5.18, from which the four folding states of the protein can be derived in different pH ranges together with their charge state distributions.

Tucker Models In N-way PCA or Tucker models, the number of components for the different modes can vary, in contrast to PARAFAC models. Tucker models can be derived for a different number of modes and are called *Tucker1*, *Tucker2*, and so on. Consider a three-way array $\underline{\boldsymbol{X}}$. Then the corresponding Tucker3 model is given for an element, x_{ijk}, of this matrix by the loadings a, b, and c and the element g of a core matrix by

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} a_{ip} b_{jq} c_{kr} g_{pqr} + e_{ijk} \tag{5.88}$$

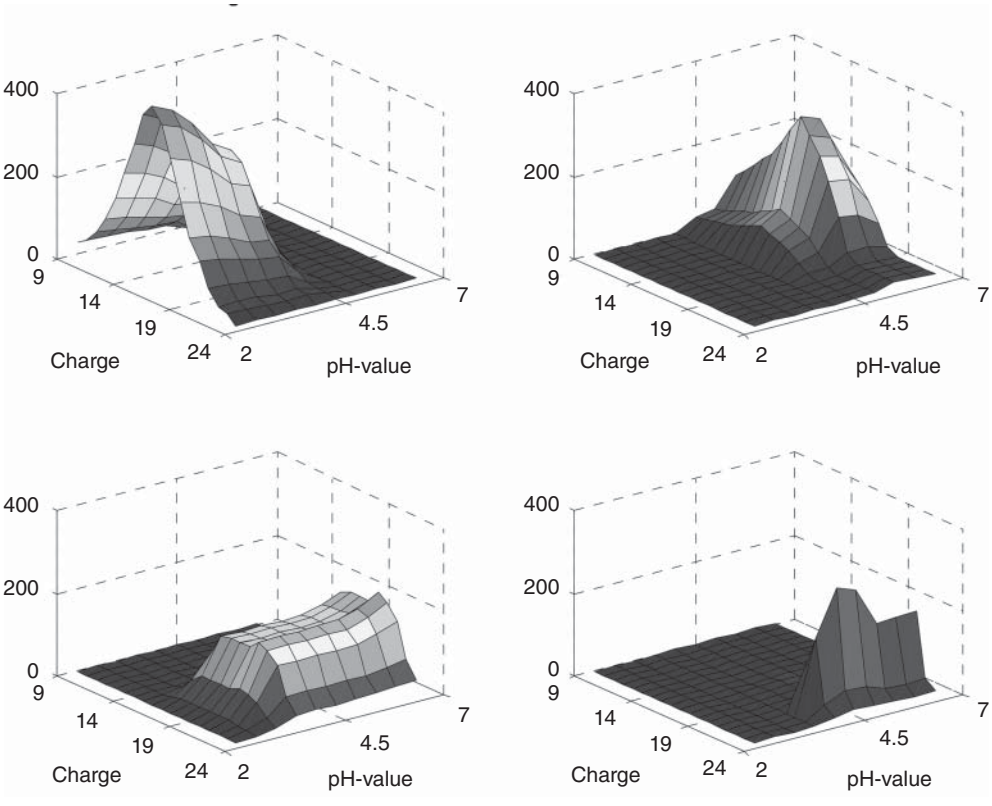


Figure 5.18 Folding states of the protein myoglobin derived from PARAFAC decomposition of charge states observed in ESI-MS spectra in dependence on pH values.

where a_{ip}, b_{jq}, and c_{kr} are the elements of the matrices of modes $\boldsymbol{A}$ $(I \times P)$, $\boldsymbol{B}$ $(J \times Q)$, and $\boldsymbol{C}$ $(K \times R)$; g_{pqr} is the element of the core matrix $\boldsymbol{G}$ $(P \times Q \times R)$ with the P, Q, R components in the different modes; and e_{ijk} represents the residual. A pictorial representation of the Tucker3 model is given in Figure 5.19. As in PARAFAC

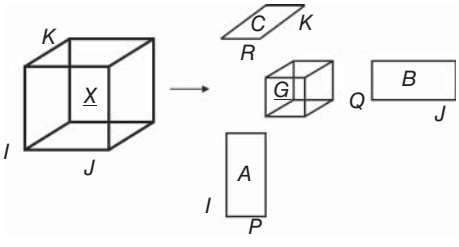


Figure 5.19 Tucker3 model with different component numbers P, Q, and R in the three modes.

modeling, an ALS algorithm can be used to estimate the parameters of a Tucker3 model.

In contrast to PARAFAC, the Tucker models do not have unique axes, so the axes can be rotated, transformed, or simplified. The PARAFAC approach also finds the best-fitting F-component model of a three-way data set, while Tucker modeling results in a finite dimensional subspace in each way.

Cluster Analysis

The second strategy of unsupervised learning is based on cluster analysis. With this method, the objects are aggregated stepwise according to the similarity of their features. As a result, hierarchically or nonhierarchically ordered clusters are formed. In order to describe the similarity of objects, we need to learn about appropriate similarity measures.

Distance and Similarity Measures

In the narrow sense, cluster analysis should not be confused with classification methods, where unknown objects are assigned to existing classes. Cluster analyses belong to the methods of unsupervised learning or unsupervised pattern recognition.

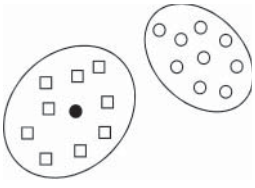
To decide on the similarity of objects, the *distance measures* as common in pattern recognition are used. The shorter the distance between objects, the more similar they are.

A general distance measure is the distance after Minkowski or L_p metric

$$d_{ij} = \left[\sum_{k=1}^{K} \left| x_{ik} - x_{jk} \right|^p \right]^{1/p} \tag{5.89}$$

where K is the number of variables and i and j the indices for objects i and j.

A *cluster* describes a group of objects that are more similar to each other than to objects outside of the group. A *seed* of a cluster serves a single object or the centroid (•).



In most cases, the Euclidean distance is applied, for which $p = 2$. For example, the Euclidean distance of two objects 1 and 2 reveals

$$d_{12} = [(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2]^{1/2} \tag{5.90}$$

In the case $p = 1$, the so-called Manhattan or city-block distance is obtained. This distance refers to the passage around a corner, that is,

$$d_{ij} = \sum_{k=1}^{K} |x_{ik} - x_{jk}| \tag{5.91}$$

Figure 5.20 demonstrates the city-block and the Euclidean distance graphically.

A disadvantage of measures based on the L_p metric is their dependence on the dimensions used. Scaling of data is frequently unavoidable if these distance measures are to be applied.

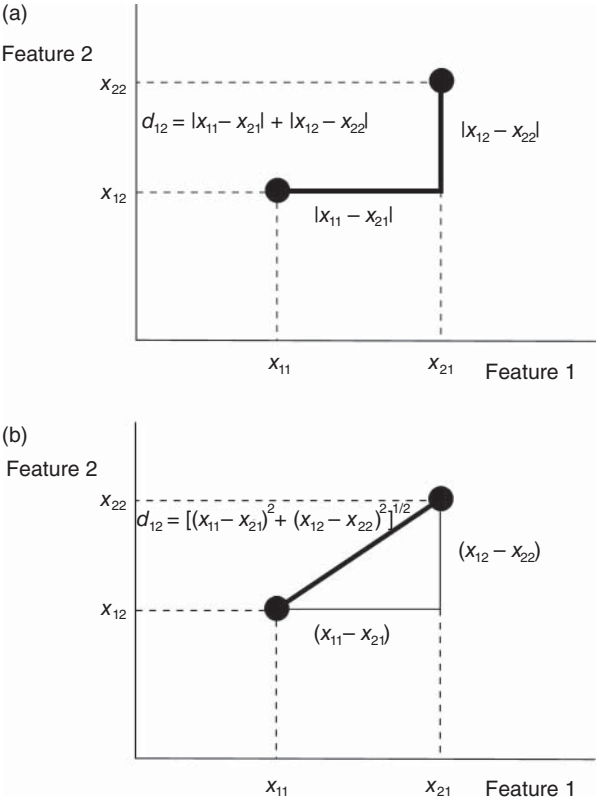


Figure 5.20 City-block distance (a) and Euclidean distance (b) for two features.

A measure that accounts for the different scales of variables and, in addition, for their correlations is the Mahalanobis distance. This invariant measure is calculated by the following formula:

$$D_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^t \mathbf{C}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{5.92}$$

where $\boldsymbol{C}$ is the covariance matrix (Eq. (5.11)), and $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the column vectors for objects i and j, respectively.

Scaling of data is not necessary if the Mahalanobis distance is used. In addition, with this measure distortion occasioned by correlations of features or feature groups is avoided. In contrast, if the Euclidean distance were applied in the case of two highly correlated variables, these variables would be used as two independent features although they provide identical information.

Complementary to distance measures are *similarity measures.* For example, the similarity measure, S_{ij}, on the basis of the

A distance measure based on the standard deviation of a variable j, s_j, is the Pearson distance:

$$d_{ij} = \sqrt{\frac{\sum_{k=1}^{K} (x_{ik} - x_{jk})^2}{s_j^2}}$$

Minkowski distance is

$$S_{ij} = 1 - \frac{d_{ij}}{d_{ij}(\max)} \tag{5.93}$$

where d_{ij} (max) represents the maximum distance of objects found in the data. Completely similar objects reveal a similarity measure of $S_{ij} = 0$. For completely dissimilar objects, a value $S_{ij} = 1$ is expected.

Hierarchical Cluster Analysis

Hierarchical cluster analysis is deduced from *taxonomy,* where biological species are ordered with respect to phenomenological similarities.

One possibility for clustering objects is their hierarchical aggregation. Here the objects are combined according to their distances or similarities to each other. We distinguish between agglomerative and divisive procedures. Divisive cluster formation is based on splitting the whole set of objects into individual clusters. In the case of the more frequently used agglomerative clustering, one starts with single objects and merges them into larger object groups.

In order to better understand the different steps of cluster analyses, hierarchical (agglomerative) clustering is demonstrated here for a data set from clinical analysis.

Example 5.9 *Cluster Analysis (Hierarchical)*

On the basis of cluster analysis, the grouping of patients' serum samples is to be investigated. Features are the concentrations of calcium and phosphate analyzed in the serum samples. The values are given in Table 5.4.

Table 5.4 Concentrations of calcium and phosphate in six blood serum samples.

Object (serum sample)	Features	
	Calcium (mg 100 ml^{-1})	Phosphate (mg 100 ml^{-1})
1	8.0	5.5
2	8.25	5.75
3	8.7	6.3
4	10.0	3.0
5	10.25	4.0
6	9.75	3.5

In the first step, we calculate the distance matrix for all the data based on one of the distance measures (Eqs. 5.89–5.92).

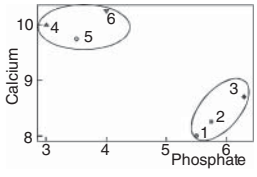
Here we use the Euclidean distance. As an example, the distance of objects 1 and 2 is evaluated by taking into account the feature values from Table 5.4:

$$d_{12} = [(8 - 8.25)^2 + (5.5 - 5.75)^2]^{1/2} = 0.354$$

Calculation of the other object distances is carried out by analogy, that is, every object is to be compared with the remaining objects. The distance between one and the same object is zero:

Object	1	2	3	4	5	6
1	0					
2	0.354	0				
3	1.063	0.711	0			
4	3.201	3.260	3.347	0		
5	2.704	2.658	2.774	1.031	0	
6	2.658	2.704	2.990	0.559	0.707	0

Clusters of objects with two or three features can be graphically represented. The visual clustering of blood sera in Table 5.4 provides



In multidimensional space, cluster analysis becomes obligatory.

Reduction of the distance matrix is performed by aggregation of objects. Objects with the shortest distance are aggregated first. In this example, the method of *weighted average linkage* is used for aggregation, where the objects are combined by averaging the calculated distances. The following steps demonstrate the building of clusters.

First reduced matrix: the shortest distance in the distance matrix is between objects 1 and 2, that is, $d_{12} = 0.354$. The two objects are aggregated to a new object 1* and their new distance is set to zero. The distance matrix is recomputed by averaging the individual distances as follows:

$$d_{1*3} = \frac{d_{13} + d_{23}}{2} = \frac{1.063 + 0.711}{2} = 1.774$$
$$d_{1*4} = \frac{d_{14} + d_{24}}{2} = \frac{3.202 + 3.260}{2} = 3.231$$
$$d_{1*5} = \frac{d_{15} + d_{25}}{2} = \frac{2.704 + 2.658}{2} = 2.681$$
$$d_{1*6} = \frac{d_{16} + d_{26}}{2} = \frac{2.658 + 2.704}{2} = 2.681$$

Clustering of objects is related to the Q technique. Usually, distance measures are used for this. Clustering of features is called R technique. The basis for that is the computation of the correlation matrix.

We obtain for the reduced matrix:

Object	1*	3	4	5	6
1*	0				
3	1.774	0			
4	3.231	3.347	0		
5	2.681	2.774	1.031	0	
6	2.681	2.990	0.559	0.707	0

Second reduced matrix: the shortest distance is observed here for objects 4 and 6 with $d_{46} = 0.559$. The two objects 4 and 6 form a new object 4*, the distance of which is again set to zero. The combination of the d values of that row reveals the actual distance matrix:

$$d_{54^*} = \frac{d_{54} + d_{56}}{2} = \frac{1.031 + 0.707}{2} = 0.869$$
$$d_{4^*3} = \frac{d_{43} + d_{63}}{2} = \frac{3.347 + 2.990}{2} = 3.169$$
$$d_{4^*1^*} = \frac{d_{41^*} + d_{61^*}}{2} = \frac{3.231 + 2.681}{2} = 2.956$$

Object	1*	3	4*	5
1*	0			
3	1.774	0		
4*	2.956	3.169	0	
5	2.681	2.774	0.869	0

Third reduced matrix: the minimal distance is now $d_{54^*} = 0.869$. A new object 5* is defined and a new distance matrix arises:

$$d_{1^*5^*} = \frac{d_{51^*} + d_{4^*1^*}}{2} = \frac{2.681 + 2.956}{2} = 2.819$$
$$d_{35^*} = \frac{d_{4^*3} + d_{53}}{2} = \frac{3.169 + 2.774}{2} = 2.972$$

Object	1*	3	5*
1*	0		
3	1.774	0	
5*	2.819	2.972	0

Fourth reduced matrix: here objects 1* and 3 are aggregated with $d_{1^*3} = 1.774$ to object 3*. Finally, the distance to the remaining object 5* is evaluated:

$$d_{3^*5^*} = \frac{d_{5^*1^*} + d_{5^*3}}{2} = \frac{2.819 + 2.972}{2} = 2.895$$

Object	3*	5*
3*	0	
5*	2.895	0

Graphically, the distances between the clusters can be demonstrated in a *dendrogram* (cf. Figure 5.21).

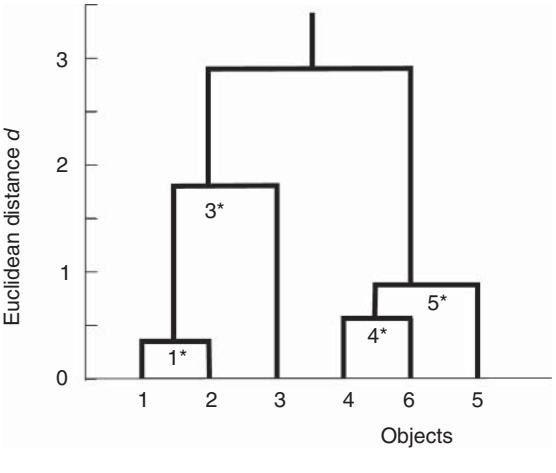


Figure 5.21 Dendrogram for the clinical–analytical data from Table 5.4.

For deciding on the number of clusters, different criteria can be exploited. Very often, the number of clusters is known. In the given example on clinical data, the number of clusters might be predefined by a given number of diseases. In some cases, the number of clusters can be deduced from a predetermined distance measure or difference between the clusters.

In Example 5.9, aggregation of clusters was carried out by the weighted average linkage methods. In general, the distance to a new object or cluster k (labeled in the example by a star) is computed by forming the average between the individual distances of objects A and B to object i.

Weighted Average Linkage

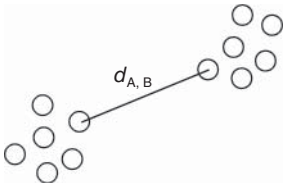
$$d_{ki} = \frac{d_{Ai} + d_{Bi}}{2} \tag{5.94}$$

The size of the clusters and their weights are assumed to be equal.

For the actualization of the distance matrix, several other formulae exist. The most important ones are considered now for the case of aggregating two clusters.

Single Linkage Here the shortest distance between the opposite clusters is calculated, that is,

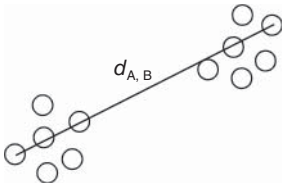
$$d_{ki} = \frac{d_{Ai} + d_{Bi}}{2} - \frac{|d_{Ai} - d_{Bi}|}{2} = \min(d_{Ai}, d_{Bi}) \tag{5.95}$$



As a result, clusters are formed that are loosely bound. The clusters are often linearly elongated in contrast to the usual spherical clusters. This chaining is occasioned by the fusion of single objects to a cluster. The procedure is related to the k-nearest neighbor (k-NN) method.

Complete Linkage This method is based on the largest distance between objects of the opposite clusters to be aggregated:

$$d_{ki} = \frac{d_{Ai} + d_{Bi}}{2} + \frac{|d_{Ai} - d_{Bi}|}{2} = \max(d_{Ai}, d_{Bi}) \tag{5.96}$$



Well-separated, small, compact spherical clusters tend to be formed.

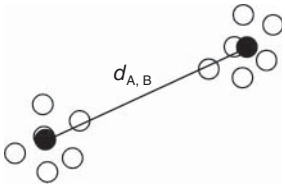
Unweighted Average Linkage With this method, the number of objects in a cluster is used for weighting the cluster distances:

$$d_{ki} = \frac{n_A}{n} d_{Ai} + \frac{n_B}{n} d_{Bi} \quad \text{with} \quad n = n_A + n_B \tag{5.97}$$

where n_A and n_B are the number of objects in cluster A and B, respectively. No deformation of the cluster is observed. To some extent, small clusters consisting of outliers might arise.

Centroid Linkage Here, the centroid calculated by the average of a cluster is applied as the basis for aggregation without distorting the cluster space:

$$d_{ki} = \frac{n_{Ai}}{n} d_{Ai} + \frac{n_B}{n} d_{Bi} - \frac{n_A n_B}{n^2} d_{AB} \tag{5.98}$$



Median Linkage For determination of the centroid, the median can also be used:

$$d_{ki} = \frac{d_{Ai}}{2} + \frac{d_{Bi}}{2} - \frac{d_{AB}}{4} \tag{5.99}$$

An advantage is the fact that the importance of a small cluster is preserved after aggregation with a large one.

Ward's Method With the Ward method, the clusters are aggregated such that a minimum increase in the within-group error sum of squares results:

$$d_{ki} = \frac{n_A + n_i}{n + n_i} d_{Ai} + \frac{n_B + n_i}{n + n_i} d_{Bi} - \frac{n_i}{n + n_i} d_{AB} \tag{5.100}$$

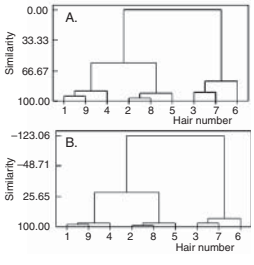
The error sum of squares is defined as the sum over the squared deviations of each from the centroid of its own cluster. This aggregation method leads to well-structured dendrograms and is probably the most frequently used method.

A more general procedure can be derived by applying a distance formula as introduced by Lance und Williams:

$$d_{ki} = \alpha_A d_{Ai} + \alpha_B d_{Bi} + \beta d_{AB} + \gamma |d_{Ai} - d_{Bi}| \tag{5.101}$$

In dependence on the choice of the parameters α, β, and γ, limiting cases result, which represent the aforementioned agglomeration methods (Table 5.5).

Dendrograms for cluster analysis of the hair data in Table 5.1 based on the Euclidean distance and the single linkage (A) or Ward method (B).



Nonhierarchical Cluster Analysis

With this method, the object clusters are not hierarchically ordered but may be partitioned independently of each other.

Commonly in nonhierarchical cluster analysis, one starts with an initial partitioning of objects to the different clusters. After that, the membership of the objects to the clusters, for example, to the cluster centroids, is determined and the objects are newly partitioned. We consider here a general method for nonhierarchical clustering that can be used for both crisp (classical) and fuzzy clustering, the *c-means algorithm.*

Table 5.5 Parameters for hierarchical cluster analysis by means of the general distance formula after Lance and Williams in Eq. (5.101).

Method	α_A	α_B	β	γ
Unweighted average	$n_A/(n_A+n_B)$	$n_B/(n_A+n_B)$	0.0	0.0
Single	0.5	0.5	0.0	−0.5
Complete	0.5	0.5	0.0	0.5
Weighted average	0.5	0.5	0.0	0.0
Centroid	$n_A/(n_A+n_B)$	$n_B/(n_A+n_B)$	$-n_A n_B/(n_A+n_B)^2$	0.0
Median	0.5	0.5	−0.25	0.0
Ward	$(n_A+n_i)/n_{ABi}$	$(n_B+n_i)/n_{ABi}$	$-n_i/n_{ABi}$	0.0

$n_{ABi} = n_A + n_B + n_i$.

At the beginning, the objects are partitioned into subsets S_i, where i is indexed from 1 to c, the number of clusters. The membership of an object with the feature vector $\boldsymbol{x}_k$ to cluster i can be characterized by means of a membership function, m_{ik}, as follows:

If methods of fuzzy clustering are used, an object can belong to different clusters to a different degree. If crisp clustering is applied, an object is uniquely assigned to a single cluster.

$$m_{ik} := m_{S_i}(\boldsymbol{x}_k) \tag{5.102}$$

In the case of a crisp set, the membership value is either 0 or 1. For fuzzy sets, the membership to an object can assume values in the interval from 0 to 1 (cf. Section 8.3).

The matrix $\boldsymbol{M} = [m_{ik}]$ is termed c-partition, if the following conditions are fulfilled:

- The membership of n objects to the clusters are either crisp or fuzzy, that is,

$$m_{ik} \in \{0,1\} \text{ or } [0,1] \quad 1 \le i \le c, 1 \le k \le n \tag{5.103}$$

- The sum of memberships of objects to a given partition is equal to 1 for crisp sets. In the case of fuzzy sets, the sum is normalized to membership values of 1, that is,

$$\sum_{i=1}^{c} m_{ik} = 1 \quad 1 \le k \le n \tag{5.104}$$

- Within a given partition, the objects are to be partitioned over all of the clusters, that is, each cluster of a partition contains at least one object. On the other hand, in a 2-partition, at best $n-1$ objects can belong to a single cluster.

$$0 < \sum_{k=1}^{n} m_{ik} < n \quad 1 \le i \le c \tag{5.105}$$

Example 5.10 *Clustering (Nonhierarchical)*

Consider as an example several 2-partitions for the three objects $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$:

$$\boldsymbol{M}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{M}_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad \boldsymbol{M}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\boldsymbol{M}_4 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{M}_5 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Each row represents a definite cluster and the columns characterize the objects, as they are assigned to the clusters. In this example, there are only two genuine 2-partitions, that is, $\boldsymbol{M}_3$ and $\boldsymbol{M}_5$. In the partition $\boldsymbol{M}_1$, the object $\boldsymbol{x}_3$ is not partitioned at all. In $\boldsymbol{M}_2$, the object $\boldsymbol{x}_2$ has a twofold presence, and in $\boldsymbol{M}_1$ and in $\boldsymbol{M}_4$, the second cluster does not contain any object.

A *partition* is a definite assignment of objects to a given cluster.

As an example of fuzzy c-partitions, the following partitions for the three objects are given:

$$\boldsymbol{M}_1 = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} \quad \boldsymbol{M}_2 = \begin{bmatrix} 0.8 & 0.5 & 0.2 \\ 0.2 & 0.5 & 0.8 \end{bmatrix}$$

$$\boldsymbol{M}_3 = \begin{bmatrix} 0.8 & 0.9 & 0.3 \\ 0.2 & 0.2 & 0.7 \end{bmatrix} \quad \boldsymbol{M}_4 = \begin{bmatrix} 0.8 & 1 & 0.9 \\ 0.2 & 0 & 0.1 \end{bmatrix}$$

As can be easily seen, a wrong partition is among the four cases, that is, $\boldsymbol{M}_3$. There, the sum of membership values exceeds 1 for object $\boldsymbol{x}_2$.

To find genuine partitions, the following scheme is applied:

- Characterization of the clusters by their centroids:

$$v_i = \frac{1}{\sum_{k=1}^{n} m_{ik}} \sum_{k=1}^{n} m_{ik}^q x_k \tag{5.106}$$

  where v_i is the centroid of cluster i; q is the exponent that expresses the degree of fuzziness, that is, for $q = 1$, the classical c-means algorithm is obtained.
- Computation of the difference between the objects and the cluster centroids by

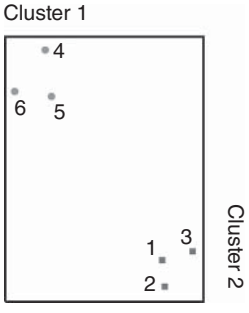
$$\|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2 = \left[\sum_{j=1}^{p} \left(x_{kj} - v_i \right)^2 \right]^{1/2} \tag{5.107}$$

  where p is the number of variables.

- Minimization of the distance function by

$$\min\ z(\boldsymbol{M}, \boldsymbol{V}) = \sum_{i=1}^{c} \sum_{k=1}^{n} m_{ik} \left\| \boldsymbol{x_k} - \boldsymbol{v_i} \right\|^2 \tag{5.108}$$

The results of a c-means clustering can be visualized by plotting the object – cluster distances. For the data in Table 5.4, we obtain



Minimization of the function $z(\boldsymbol{M},\boldsymbol{V})$ may become a computational problem. The number of all partitions to be tested is calculated according to

$$\#\text{Partitions} = \frac{1}{c!}\left[\sum_{j=1}^{c} \binom{c}{j} (-1)^{c-j} j^n\right] \tag{5.109}$$

For example, 10 clusters with 25 objects in total 10^{18} different partitions would have to be tested. Fortunately, not all partitions are to be computed, since algorithms that find an optimum partition iteratively according to predefined criteria are available.

Frequently, a threshold is defined, in order to judge the improvement by changing from a partition M^l to M^{l+1}. If the criteria fall under the threshold, the computation can be stopped.

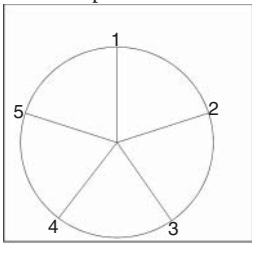
Graphical Methods

All of the methods discussed so far for grouping of data cannot surpass the abilities of humans to recognize patterns. Therefore, more and more methods are developed that exploit the human ability for *recognizing patterns.* These graphical methods are based on a compact representation of multidimensionally characterized objects. We learn here about representation of multivariate data by star and sun-ray plots as well as by Chernoff faces.

For all those methods, standardization and translation to positive feature values are a prerequisite.

Star and Sun-Ray Plots

Feature representation



If the feature values are transferred into polygons and the polar coordinates are plotted, then star-like representations emerge. Consider p features where a circle is segmented into p uniform sectors. Each sector describes an angle of $360°/p$. Each boundary line is assigned a feature. The actual value of a feature is then plotted at a distance from the mean point of the circle. After connection of the points by straight lines, a polygon is obtained, which forms a characteristic pattern.

Graphical representation is carried out in the form of *stars* (Figure 5.22A), which are also termed *diamonds*. Or, the result

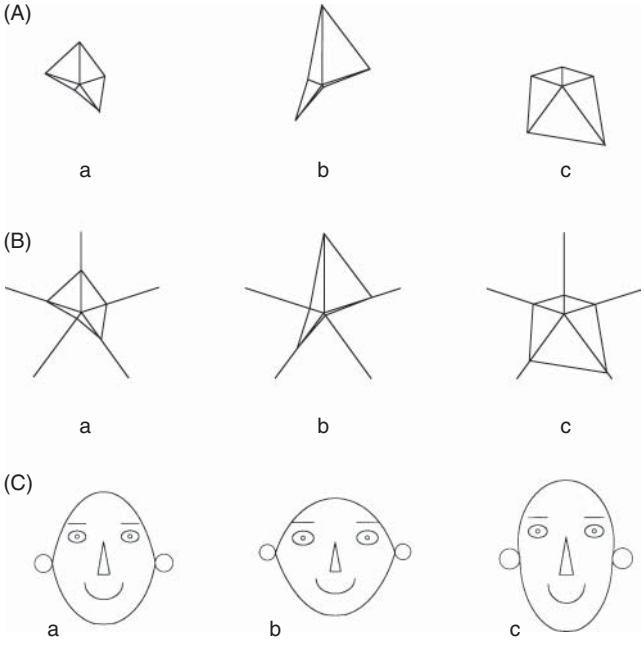


Figure 5.22 Graphical method for grouping of the hair data in Table 5.1 for representatives of the three subject groups *a*, *b*, and *c* (for assignments cf. Table 5.8). (A) Stars, (B) sun-rays, and (C) Chernoff faces.

is represented by drawing the boundaries as rays giving the so-called sun-ray plots (Figure 5.22B).

Chernoff Faces

Assignment of features to facial parts leads to representation of the objects as faces. Well known are the faces introduced by Chernoff. The features are characterized by facial parameters, such as the size or curvature of the eyes, the mouth, the eye brows, the nose, or the upper and lower half of the case. As an example, the hair data of the three subject types are plotted in Figure 5.22C, as Chernoff faces.

Frequently, up to 20 different facial parameters are used, as it is demonstrated in Figure 5.23 for distinguishing healthy and

Figure 5.23 Chernoff faces for distinguishing of serum samples of diseased and healthy patients on the basis of 20 clinical analyses.

ill patients on the basis of clinical analyses for blood serum samples.

One disadvantage of Chernoff faces is the fact that the individual faces cannot be varied independently of each other. Faces proposed by Flury–Riedwyl (cf. [1]) overcome this disadvantage.

5.3 Supervised Methods

If the membership of objects to particular clusters is known in advance, the methods of supervised pattern recognition can be used. In this section, the following methods are explained: linear learning machine (LLM), discriminant analysis, *k*-NN, the soft independent modeling of class analogies (SIMCA) method, and Support Vector Machines (SVMs).

Supervised methods for recognizing patterns can also be based on multivariate modeling methods, for example, by use of PLS as discussed in Section 6.2.2. The method is termed *discriminant analysis–partial least squares* (DA-PLS) analysis where the input feature data from the $\boldsymbol{X}$ matrix and the assignment to a class is described in the $\boldsymbol{Y}$ matrix. To avoid a ranking of classes, the containment of classes is not coded in a single classification vector, for example, classes 1–6, but is described by ones or zeros columnwise in the $\boldsymbol{Y}$ matrix.

Linear Learning Machine

The first analytical application of a pattern recognition method dates back to 1969 when classification of mass spectra with respect to certain molecular mass classes was tried with the LLM. The basis for classification with the LLM is a discriminant function that divides the *n*-dimensional space into category regions that can be further used to predict the category membership of a test sample.

Consider the data in Table 5.6 that represent the iodine content of hair samples from five different patients belonging to two categories. Since only one feature has been measured ($p = 1$), the data can be represented in one-dimensional space as given in Figure 5.24.

To find a decision boundary that separates the two groups, the data vectors have to be augmented by adding a $(n + 1)$ component equal to 1.0. This ensures that the boundary for separating the classes passes through the origin. If more than two categories

Table 5.6 Iodine content of hair samples from different patients.

Hair sample	Iodine content (ppm)	Augmented component
1	0.29	1.0
2	4.88	1.0
3	0.31	1.0
4	3.49	1.0
5	4.46	1.0

Figure 5.24 Representation of iodine data from hair samples in Table 5.6; the two groups are labeled as *open* and *filled circles*.

are to be separated, several linear discriminant functions would have to be constructed.

The boundary that separates the two categories is found iteratively by adjusting the elements of a weight vector, $\boldsymbol{w}$, which is normal to the boundary, such that the dot product of $\boldsymbol{w}$ and any vector of the full circles is positive, while that of $\boldsymbol{w}$ and the empty circles is negative (Figure 5.25). The decision boundary s is expressed by

$$s = w_1 x_1 \tag{5.110}$$

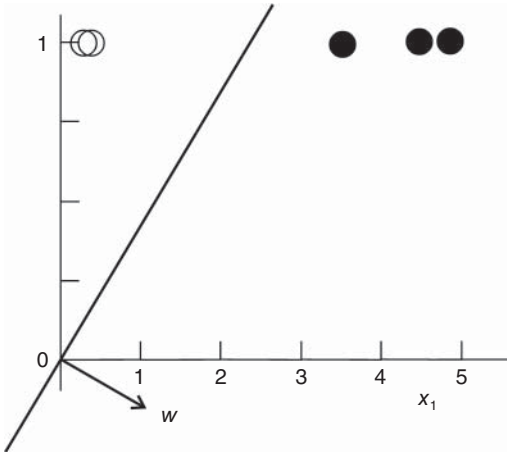


Figure 5.25 Linear learning machine (LLM): representation of iodine data of Table 5.6 augmented by an additional dimension and separated by a straight-line boundary with the normal weight vector $\boldsymbol{w}$.

or in general

$$s = \boldsymbol{w}^t \boldsymbol{x} = \|\boldsymbol{w}\| \cdot \|\boldsymbol{x}\| \cos\theta \tag{5.111}$$

where $||\cdot||$ is the vector norm, that is, $\left[\sum_i x_i^2\right]^{1/2}$, $\boldsymbol{x}$ the augmented data vector, s the scalar variable, and θ the angle between $\boldsymbol{w}$ and $\boldsymbol{x}$.

If the angle θ is less than 90°, it is obvious that the objects represented as full circles are categorized and that $s > 0.0$. Conversely, if the angle θ is greater than 90°, the scalar variable will be $s < 0$ and the empty circle objects are described.

To find the weight elements, $\boldsymbol{w}$, they are set initially to random numbers. The objects are then classified by computing s and checking against the correct answer. If all classifications are correct, the training process can be stopped and the LLM can be used for further classification purposes. However, if the response is incorrect, new weights have to be calculated by updating the old ones, for example, by

$$\boldsymbol{w}^{(\text{new})} = \boldsymbol{w}^{(\text{old})} + c\boldsymbol{x} \quad \text{with} \quad c = \frac{-2s}{\boldsymbol{x}^T \boldsymbol{x}} \tag{5.112}$$

here the constant c is chosen such that the boundary is reflected to the correct side of a data point by the same distance as it was in error. The updating of weights is repeated as long as all objects are correctly classified.

Of course, the LLM will only work properly if the data are linearly separable. One should also remember that the solution for positioning the boundary is not unique so that different solutions will emerge if the order for presenting the training objects is changed.

Additional disadvantages are the simple class boundaries, the danger of wrong assignments of outliers, or the slow convergence. In addition, LLM is restricted to the separation of only two classes (binary classifier).

Discriminant Analysis

Linear Discriminant Analysis

A decision boundary separates two or more groups of data.

A more formal way of finding a decision boundary between different classes is based on linear discriminant analysis (LDA) as introduced by Fisher and Mahalanobis. The boundary or hyperplane is calculated such that the variance between the classes is maximized and the variance within the individual classes is minimized. There are several ways to arrive at the decision hyperplanes. In

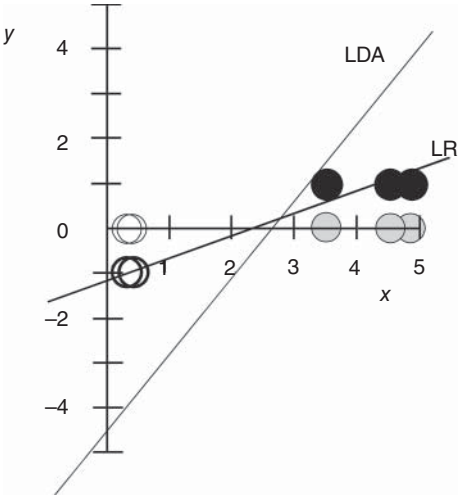


Figure 5.26 Decision lines based on linear regression (*LR*) and linear discriminant analysis (*LDA*) for separating the objects represented by *filled* and *empty circles*.

fact, one of the routes Fisher described can be understood from the principles of straight-line regression (cf. Section 6.1).

Two-Class Case Consider again the iodine data in Table 5.6. If you create an augmented variable y with values of +1 for objects from one group and −1 for objects from the other group and perform a linear regression (LR) of y on x for the five training objects, then the regression equations for y will have coefficients similar to the LDA weights, that is,

$$y = -1.082 + 0.477x$$

Comparison with the LLM reveals that the boundary is not constrained to pass through the origin but a cut-off point is included into the model. Application of LDA gives the discriminant function

$$s = -4.614 + 1.718x$$

Both boundaries are given in Figure 5.26. Depending on the class, s takes on positive or negative values.

Multiclass Case To arrive at the (nonelemental) LDA solution, an eigenvalue problem has to be solved. To generalize the problem, we have to consider a data matrix $\boldsymbol{X}$ with n objects and p feature

variables. There are g different groups or classes indexed by g_1 to g_{n_j}:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & x_{2p} \\ x_{31} & x_{32} & \cdots & x_{3p} \\ x_{41} & x_{42} & & x_{4p} \\ \vdots & & & \\ x_{j1} & x_{j2} & \cdots & x_{jp} \\ x_{n1} & x_{n2} & & x_{np} \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} g_1 \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} g_2 \\ \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\} g_{n_j} \end{matrix} \tag{5.113}$$

The weights of the linear discriminant functions are found as the eigenvectors of the following matrix:

$$\boldsymbol{G}^{-1}\boldsymbol{H}\boldsymbol{w} = \lambda \boldsymbol{w} \tag{5.114}$$

where λ is the eigenvalue.

The matrix $\boldsymbol{G}$ is derived from the covariance matrix $\boldsymbol{C}$ (Eq. (5.114)) of the different classes or groups g as follows:

The results of categorical classification are represented in the *confusion matrix,* which contains the numbers of correct classified objects in each class on the main diagonal and the misclassified objects in the off-diagonal.

$$\boldsymbol{G} = (n-g)\boldsymbol{C} = (n-g)\frac{1}{n-g}\sum_{j=1}^{g}(n_j - 1)\boldsymbol{C}_j \tag{5.115}$$

$$\boldsymbol{C}_j = \frac{1}{n_j - 1}\sum_{l \in g_j}(x_{li} - \overline{x}_{ji})(x_{lk} - \overline{x}_{jk}) \tag{5.116}$$

where n is the total number of objects, n_j the number of objects in group j, and $l \in g_j$, which means index l is an element of the jth group g_j.

Matrix $\boldsymbol{H}$ describes the spread of the group means $\overline{\boldsymbol{x}}_j$ over the grand average $\overline{\boldsymbol{x}}$, that is,

$$\boldsymbol{H} = \sum_{j=1}^{g} n_j(\overline{\boldsymbol{x}}_j - \overline{\boldsymbol{x}})(\overline{\boldsymbol{x}}_j - \overline{\boldsymbol{x}})^{\mathrm{T}} \tag{5.117}$$

$$\overline{\boldsymbol{x}} = \frac{1}{n}\sum_{j=1}^{n} n_j \overline{\boldsymbol{x}}_j \tag{5.118}$$

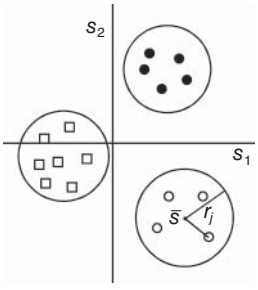
The eigenvector, $\boldsymbol{w}_1$, which is found on the basis of the greatest eigenvalue λ_1, provides the first linear discriminant function, s_1, by

$$s_1 = w_{11}x_1 + w_{12}x_2 + \cdots + w_{1p}x_p \tag{5.119}$$

With the residual x data, the second largest eigenvalue is computed and with the new eigenvector, $\boldsymbol{w}_2$, the second discriminant function, s_2, is obtained:

$$s_2 = w_{21}x_1 + w_{22}x_2 + \cdots + w_{2p}x_p \tag{5.120}$$

The procedure is continued until all discriminant functions are found for solving the discrimination problem. By plotting pairs of discriminating functions against each other, the best separation of objects into groups after the linear transformation of the initial features can be visualized (cf. Figure 5.25). The projection of a particular object onto the separating line or hyperplane is called its *score* on the linear discriminant function.



Classification of unknown objects is carried out by inserting the feature data into the discriminant functions in order to transform its coordinates in the same way as for the original data set. The object is then assigned to that class for which its centroid has the smallest Euclidian distance:

$$\min_j \|\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_u - \overline{\boldsymbol{x}}_j)\| \quad \text{with} \quad j = 1, \ldots, g \tag{5.121}$$

Example 5.11 *Linear Discriminant Analysis*

LDA is to be used to build a classification model for the hair data in Table 5.1. Based on the model, an unknown hair sample as given in Table 5.7 is to be classified.

Table 5.7 Elemental content of an unknown hair sample in parts per million.

Cu	Mn	Cl	Br	I
9.2	0.27	2200	9.8	4.7

In the first step, the discriminant function is computed by Eqs. 5.114–5.118. The membership of hair samples in the individual groups is represented by a classification vector as given in Table 5.8.

For the first two discriminant functions, we get:

$$s_1 = 0.227x_{\mathrm{Cu}} + 0.694x_{\mathrm{Mn}} - 1.200x_{\mathrm{Cl}} + 0.0394x_{\mathrm{Br}} - 0.0514x_{\mathrm{I}} \tag{5.122}$$

$$s_2 = 0.00672x_{\mathrm{Cu}} + 0.936x_{\mathrm{Mn}} - 0.211x_{\mathrm{Cl}} + 1.342x_{\mathrm{Br}} - 0.395x_{\mathrm{I}} \tag{5.123}$$

The first discriminant function describes 63.39% of the variance of data and the second one 36.61%. That is, the two discriminant functions are sufficient to explain 100% (63.39% + 36.61%) of the data variance. Discrimination of

Table 5.8 Classification vector for the hair samples in Table 5.1.

Hair no.	Subject
1	B
2	A
3	C
4	B
5	A
6	C
7	C
8	A
9	B

the data into three classes can be illustrated by plotting the discriminant functions against each other (Figure 5.27).

If, in the second step, the elemental contents of the unknown hair sample (Table 5.7) are inserted into the discriminant functions of Eqs. 5.119 and 5.120, then the following values are obtained for the scalar values $s_1 = 4.416$ and $s_2 = -7.15$. If compared to the centroids of the classes (Figure 5.27), we obtain the values 15.66, 4.93, and 10.52 for the Euclidian distances of the classes A, B, and C, respectively. The shortest distance is found between the data of the unknown hair and class B, that is, the hair has to be assigned to the subject group B.

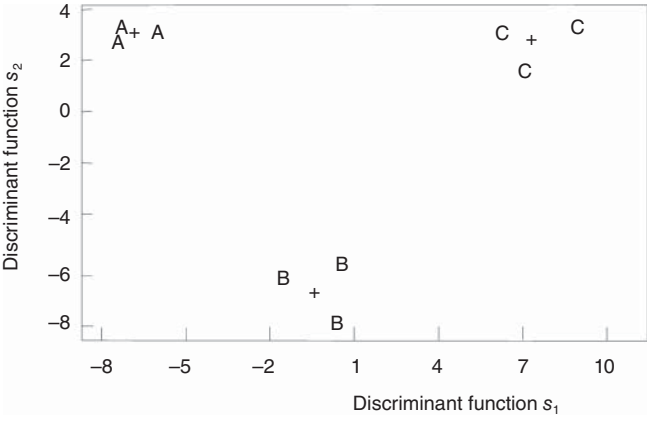
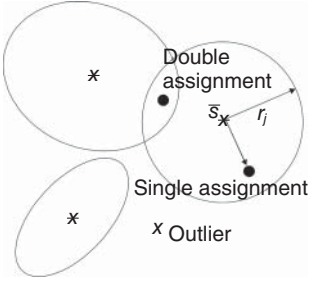


Figure 5.27 Discriminant function of the LDA of the hair data in Table 5.1 based on the classification vector in Table 5.8. The centroids of the classes are labeled by a *cross*.



Multiclass Assignment A disadvantage with this method of unique decision is the fact that simultaneous membership of an object in several classes is not detected and that outliers, which do not belong to any of the classes, will always be categorized. Therefore, the unique categorization is often replaced by assigning the object to all classes within a fixed variance range, for example, 95%. If the object lies outside of any of those variance ranges, it will not be categorized at all.

The calculation of the variance radius, r_j, is done by

$$r_j = \frac{d(n-g)}{n-g-d+1}\frac{n_j+1}{n_j}F_{(d,n-g-d+1;\alpha)} \tag{5.124}$$

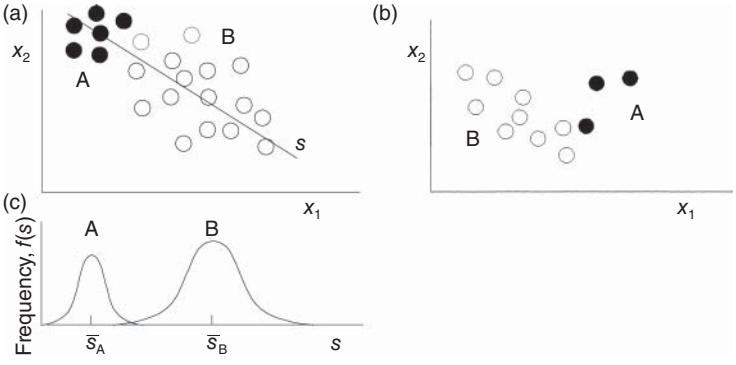


Figure 5.28 Differently spread (a) and differently directed (b) objects of the full and empty circled classes. (c) For case (a) the density function around the class centroids is given.

where d is the number of discriminant functions used and F the Fisher's F statistics with risk α.

Assignment of an object to a particular class is performed if

$$\sqrt{\sum_{i=1}^{d}(s_i - \bar{s}_{ij})^2} \leq r_j \tag{5.125}$$

here s_i represents the new coordinates of the unknown object and $\bar{s}_{ij}$ is the jth class centroid.

Necessary assumptions of LDA are the normality of data distributions and the existence of different class centroids, as well as the similarity of variances and covariances among the different groups. Classification problems therefore arise if the variances of groups differ substantially or if the direction of objects in the pattern space is different, as depicted in Figure 5.28.

Bayesian Classification

In the case that objects of all classes obey a multivariate normal distribution, an optimal classification rule can be based on *Bayes' theorem.* The assignment of a sample, $\boldsymbol{x}$, characterized by p features to a class j of all classes g is based on maximizing the *posterior probability:*

$$P(j|\boldsymbol{x}) \quad \text{for } j = 1, \ldots, g \tag{5.126}$$

Application of Bayes' theorem for calculation of the posterior probability reveals

$$P(j|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|j)P(j)}{p(\boldsymbol{x})} \tag{5.127}$$

According to Eq. (5.127), the posterior probability is computed from the probability density function for the considered class, $p(\boldsymbol{x}|j)$, the prior probability for that class $P(j)$, and the probability density function over all classes $p(\boldsymbol{x})$. A sample $\boldsymbol{x}$ is then assigned to that class j, for which the largest posterior probability is found.

For computation of the class probability density, $p(\boldsymbol{x}|j)$, the multidimensional normal distribution is assumed:

$$p(\boldsymbol{x}|j) = (2\pi)^{-p/2}|\boldsymbol{S}_j|^{-0.5}\exp[-0.5(\boldsymbol{x}-\overline{\boldsymbol{x}}_j)\boldsymbol{S}_j^{-1}(\boldsymbol{x}-\overline{\boldsymbol{x}}_j)^{\mathrm{T}}] \quad (5.128)$$

where the covariance matrix $\boldsymbol{S}_j$ based on the class centroid $\overline{\boldsymbol{x}}_j$ is obtained by

$$\boldsymbol{S}_j = \frac{1}{n_j}\sum_{i=1}^{n_j}(\boldsymbol{x}_i-\overline{\boldsymbol{x}}_j)^{\mathrm{T}}(\boldsymbol{x}_i-\overline{\boldsymbol{x}}_j) \quad (5.129)$$

$$\overline{\boldsymbol{x}}_j = \frac{1}{n_j}\sum_{i=1}^{n_j}\boldsymbol{x}_i^{(j)} \quad (5.130)$$

where n_j describes the number of samples in class j.

Maximizing the posterior probability is related to minimizing the discriminant scores obtained by

$$d_j(\boldsymbol{x}) = (\boldsymbol{x}-\overline{\boldsymbol{x}}_j)\boldsymbol{S}_j^{-1}(\boldsymbol{x}-\overline{\boldsymbol{x}}_j)^{\mathrm{T}} + \ln|\boldsymbol{S}_j| - 2\ln P(j) \quad (5.131)$$

An unknown sample is assigned to that class, j, with the shortest distance to its class centroid. The first term in Eq. (5.131) represents the Mahalanobis distance between the sample $\boldsymbol{x}$ and the class centroid $\overline{\boldsymbol{x}}_j$.

In LDA, it is assumed that the class covariance matrices are equal, that is, $\boldsymbol{S}_j = \boldsymbol{S}$ for all $j = 1$ to g. Different class covariances are allowed in quadratic discriminant analysis (QDA). The results are quadratic class boundaries based on unbiased estimates of the covariance matrix. The most powerful method is based on regularized discriminant analysis (RDA) [7]. This method seeks biased estimates of the covariance matrices, $\boldsymbol{S}_j$, to reduce their variances. This is done by introducing two regularization parameters λ and γ according to

$$\boldsymbol{S}_j(\lambda) = (1-\lambda)\boldsymbol{S}_j + \lambda\boldsymbol{S} \quad (5.132)$$

$$\boldsymbol{S}_j(\lambda,\gamma) = (1-\gamma)\boldsymbol{S}_j(\lambda) + \frac{\gamma}{p}tr[\boldsymbol{S}_j(\lambda)]\boldsymbol{I} \quad (5.133)$$

RDA is the same as QDA, if $\lambda = 0$ and $\gamma = 0$. For $\lambda = 1$ and $\gamma = 0$, the method corresponds to LDA. If $\lambda = 1$ and $\gamma = 1$, RDA is the same as the *nearest mean classifier.*

The parameters range in the interval 0 and 1; tr characterizes the trace of a matrix and $\boldsymbol{I}$ is the identity matrix.

This ensures that even in the case of ill-conditioned systems, for example, in the case of very similar spectra, good results for the

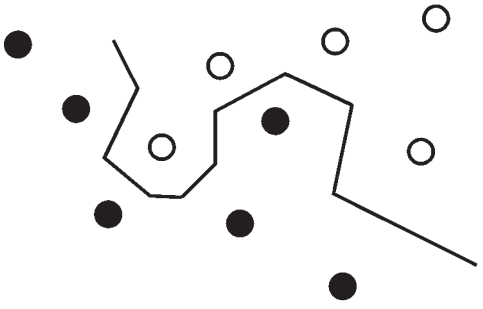


Figure 5.29 Separation boundary for classification of objects into two classes with $k = 1$.

estimation of the inverse covariance matrix in Eq. (5.131) and the subsequent classification of unknown samples are to be expected.

k-Nearest Neighbor Method

A simple nonparametric classification method is the nearest neighbor method as introduced by Fix and Hodges in 1951. For classification of an unknown object, its distance, usually the Euclidian distance, is computed to all objects. The minimum distance is selected and the object is assigned to the corresponding class.

The *k*-NN method is also used for filling in missing values or in library searches.

Typically, the number of neighboring objects k is chosen to be 1 or 3. With the k-NN method, very flexible separation boundaries are obtained as exemplified in Figure 5.29.

Unfortunately, classification is dependent on the number of objects in each class. In the case of overlapping classes, the object will be assigned to the class with the larger number of objects. This situation can sometimes be handled if no single criterion is used but alternative counting of neighborhood is allowed, for example, for class A with fewer objects, five neighbors must be found, whereas for another class B with more objects, seven neighbors would have to be considered.

In the following example, a four-class problem simulated for two-dimensional data is explored.

Example 5.12 *k-Nearest Neighbor Classification*

Here we consider simulated two-dimensional random data for four different classes with 50 objects in each class. At first,

a k-NN classification model is computed for one neighboring object revealing decision boundaries as given in Figure 5.30a. Although the resubstitution error is zero, that is, there are no misclassified objects, the error of a 10-fold cross-validated model amounts to 15.5% of misclassified objects. More stable models are obtained with more neighboring objects.

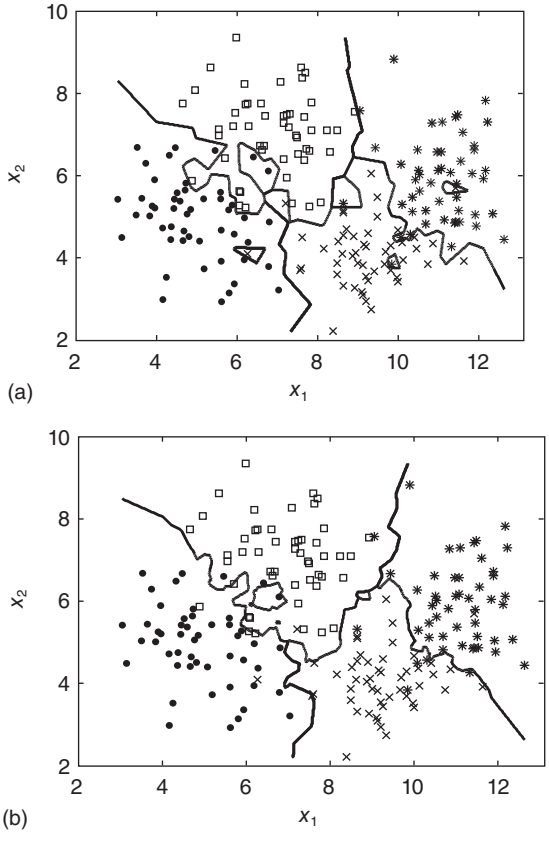


Figure 5.30 k-NN classification by cross-validated models with one neighboring object (a) and four neighboring objects (b) for simulated data.

Figure 5.31 demonstrates models for increasing numbers of neighbors up to 15. The fraction of misclassifications has a first minimum at four neighbors, where the fraction of misclassified objects is 8.0% and 9.5% for resubstitution and cross-validation, respectively. The decision boundaries for this model are given in Figure 5.30b.

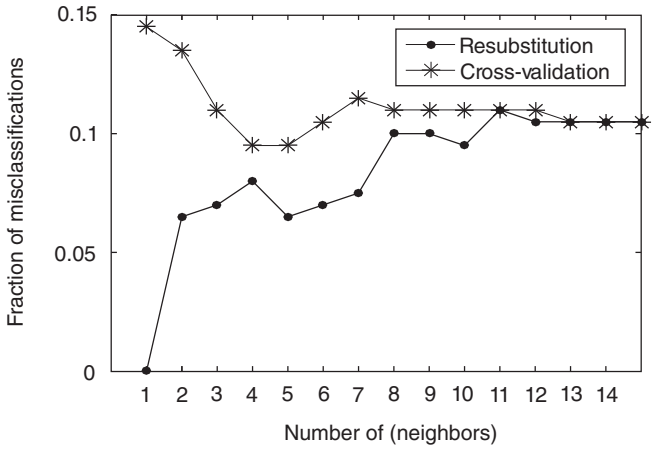


Figure 5.31 Fraction of misclassified objects in dependence on the number of neighbors used in *k*-NN classification models.

SIMCA

Apart from discrimination methods, class membership of objects can also be determined by description of the individual classes by means of a separate mathematical model independent of the model for the other classes. In terms of geometry, the model describes an envelope or a "class box" around the class, so that unknown objects can be classified according to their fit to a particular class model.

An early method developed uses multivariate normal distribution to model the classes on the basis of their data variances. Although this model has sometimes been used in analytical chemistry, it lacks general application because the method is based on the covariance matrix where it is tacitly assumed that many data exist and that the ratio between objects and variables is favorably about 6 : 1.

More often, the SIMCA method is used. This finds separate principal component models for each class. By using SIMCA, the object variable number ratio is less critical and the model is constructed around the projected, rather than the original, data. The basic steps of principal component calculations as needed for SIMCA have been outlined in the chapter on projection methods with the NIPALS algorithm (Example 5.1).

SIMCA is the abbreviation for *Soft Independent Modeling of Class Analogies.*

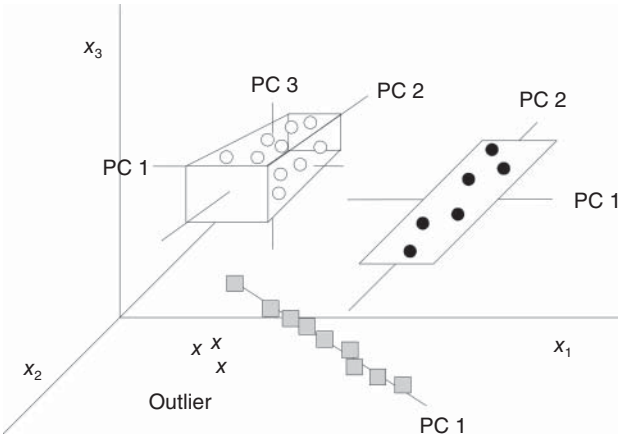


Figure 5.32 Soft independent modeling of class analogies (SIMCA) models for different numbers of significant principal components (*PCs*).

For each class q, a separate model is constructed that reveals for a single x observation:

$$x_{ij}^{q} = \overline{x}_{j}^{q} + \sum_{a=1}^{A_q} t_{ia}^{q} l_{ja}^{q} + e_{ij}^{q} \tag{5.134}$$

where $\overline{x}_{j}^{q}$ is the mean of variable j in class q, A_q the number of significant principal components in class q, t_{ia}^{q} the score of object i on component a in class q, l_{ia}^{q} the loading of variable j on principal component a in class q, and e_{ij}^{q} the residual error of object i and variable j.

The number of principal components in a class model is determined by *cross-validation.*

The principal components are found by the iterative NIPALS algorithm. Each separate model may reveal a different number of significant principal components A_q. Thus, the class models may represent lines, planes, boxes, or hyperboxes as demonstrated in Figure 5.32. The models can then be used to eliminate outlying objects, to estimate the modeling power of a particular variable, and to classify new objects.

Outliers

Objects that do not fit the estimated principal component model can be eliminated by testing the total residual variance of a class q against the residual variance of that object. The two variances are calculated as follows.

Total residual variance of class q

$$s_0^2 = \sum_{i=1}^{n} \sum_{j=1}^{p} \frac{e_{ij}^2}{(n - A_q - 1)(p - A_q)} \tag{5.135}$$

where n is the number of objects and p is the number of variables.

Residual variance for object i

$$s_i^2 = \sum_{j=1}^{p} \frac{e_{ij}^2}{p - A_q} \tag{5.136}$$

If both variances are of the same order of magnitude, the object is assigned as a typical object to the class. If $s_i^2 > s_0^2$, then the object should be eliminated to enable a more parsimonious class model to be described.

Modeling Power

The residual variance of a variable j of class q is used for estimating the modeling power of a particular variable if related to the so-called meaningful variance, the familiar expression for the variance.

Residual variance of variable j

$$s_j^2(\text{error}) = \sum_{i=1}^{n} \frac{e_{ij}^2}{n - A_q - 1} \tag{5.137}$$

Meaningful variance of variable j

$$s_j^2(x) = \sum_{i=1}^{n} \frac{(x_{ij} - \overline{x})^2}{n - 1} \tag{5.138}$$

The comparison of the two variances reveals a measure for the noise-to-signal ratio of this variable. The modeling power for variable j, R_j, is derived from the following expression:

$$R_j = 1 - \frac{s_j(\text{error})}{s_j(x)} \tag{5.139}$$

If the modeling power approaches values of 1, then the variable will be highly relevant, because the ratio between the residual error for the variable is small compared to its meaningful variance.

Classification

An unknown object with data vector $\boldsymbol{x}_u$ (dimension $1 \times p$) is determined to belong to a particular class by regression of the vector $\boldsymbol{x}_u$ on the q class models. Multiplying the data vector by the loading matrix $\boldsymbol{L}$ ($p \times A_q$) reveals an estimation for a new score vector $\widehat{\boldsymbol{t}}$ ($1 \times p$). With the score vector, the residuals are computed and

used to decide on the membership of the object to a class:

$$\hat{\boldsymbol{t}} = \boldsymbol{x}_u \boldsymbol{L} \tag{5.140}$$

$$\boldsymbol{e} = \boldsymbol{x}_u - \hat{\boldsymbol{t}} \boldsymbol{L}^{\mathrm{T}} \tag{5.141}$$

As the residual variance for object u, we get:

$$s_u^2 = \sum_{j=1}^{p} \frac{e_{uj}^2}{p - A_q} \tag{5.142}$$

The object is assigned to class q if the variances s_u^2 and s_0^2 are of similar order of magnitude. If s_u^2 is greater than s_0^2, the object is not a member of class q.

Support Vector Machines

Separation of overlapping classes is not feasible with methods such as discriminant analysis because they are based on optimal separating hyperplanes. SVMs provide an efficient solution to separating nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space.

The Support Vector Classifier

With the LLM and discriminant analysis covered in this section, classification of an object is carried out strictly by assigning it to the class on either side of the separating plane (hyperplane). To deal with overlapping classes, one approach is to allow for some objects to be on the wrong side of the margin.

Consider a hyperplane

$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w} + w_0 = 0 \tag{5.143}$$

where $\boldsymbol{w}$ denotes a weight vector and w_0 is the offset (cf. Figure 5.26). Decision rules are then defined by

$$G(\boldsymbol{x}) = sgn(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{w} + w_0) \tag{5.144}$$

Given the classification vector, $\boldsymbol{y}$, in the interval $[-1, +1]$, a function $f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w} + w_0$ with $y_i f(x_i) > 0$ can be found for all i. Then a hyperplane can be computed that creates the biggest *margin* between the training points for classes 1 and -1. The optimization problem is then given by

$$\min \left(\frac{1}{2} \|w\|^2 \right) \text{ subject to } y_i(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} + w_0) \geq 1, \quad i = 1, \ldots, n \tag{5.145}$$

The decision problem is visualized in Figure 5.33 for a separable (a) and a nonseparable case (b), where the decision boundary

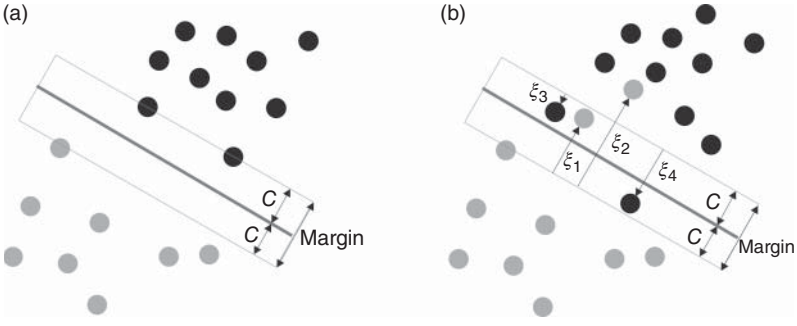


Figure 5.33 Separable case (a) and nonseparable (overlap) case (b) of the decision problem.

is represented by a solid line. The maximal margin is twice the distance C from the decision boundary. C is just the reciprocal value of the norm of the weights, that is, $1/\|\boldsymbol{w}\|$.

Now we consider overlapping classes in the feature space. One can still try to find a hyperplane allowing for some points to be on the wrong side of the margin. We define the slack variable, $\boldsymbol{\xi}$, and modify the constraints in Eq. (5.145) by

$$\min\left\{\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i\right\} \quad \text{subject to}$$
$$y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} + w_0) \geq 1 - \xi_i \quad \text{for all } i, \xi_i \geq 0 \tag{5.146}$$

The first term in Eq. (5.146) relates to maximization of the margin and the second term penalizes samples on the wrong side of the class boundaries in the nonseparable case.

The optimization problem in Eq. (5.146) is a standard situation in optimization, that is, minimization of a quadratic function with linear constraints and can be solved by applying Lagrangian theory. From this theory, it follows that the weight vector of the decision function is given by a linear combination of the training data and the Lagrange multiplier α by

$$\widehat{\boldsymbol{w}} = \sum_{i=1}^{n}\widehat{\alpha}_i y_i \boldsymbol{x}_i \tag{5.147}$$

Only those training vectors $\boldsymbol{x}_{\mathrm{i}}$ will have nonzero Lagrange multipliers α_i, which are at the class boundaries or are margin errors. These prototypes, which determine the construction of the decision function, are termed *support vectors*.

In contrast to conventional classification methods such as discriminant analysis, no assumptions about the form of the underlying class distributions are necessary with support vector classifiers.

Support vector machines can also be adapted for regression analysis (see [8]).

Support Vector Machines

The problem can be made more flexible if the feature space is enlarged using basis expansions such as polynomials or splines. Instead of the original x variables, their transforms, that is, basis functions $h(\boldsymbol{x})$, are used in the support vector classifier according to

$$f(\boldsymbol{x}) = h(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{w} + \boldsymbol{w}_0 \tag{5.148}$$

The idea of basis expansion is that linear boundaries in the enlarged space achieve better separations of the training classes and translate to nonlinear boundaries in the original space. If the dimension of the enlarged space gets very large, the classifiers are termed *SVMs*.

In a special representation of the optimization problem, the input features are represented via their inner products. Then the knowledge of transformation $h(\boldsymbol{x})$ is not required at all, but only the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This reveals the following nonlinear decision function:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + w_0 \tag{5.149}$$

Popular choices of kernel functions for SVMs are as follows:

- Polynomial of degree p with parameters a_1 and a_2:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (a_1 \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j + a_2)^p \tag{5.150}$$

- Gaussian kernel:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2}{2\sigma^2}\right) \tag{5.151}$$

- Sigmoid kernel:

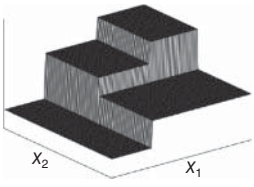
$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(a_1 \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j + a_2\right) \tag{5.152}$$

The Gaussian kernel is used in potential function classifiers, also known as *radial basis function networks*. A sigmoid kernel implements a multilayer perceptron (cf. Section 8.2) with a single hidden layer.

SVMs can be advantageously used if there are ill-behaved distributions, high dimensional data, or if there is a low ratio of training samples to the dimensionality of the input data. These situations are typically found with real-world problems and there the successful application of this approach might be expected.

Tree-Based Classification

Tree-based classification and regression are partition-based methods. Partitioning is also used in nonhierarchical cluster analysis. With classification and regression trees (CART), the feature space is partitioned into a set of rectangles and then a simple model, such as a constant, is fit to each region separately. In Figure 5.34, some fake data of a two-dimensional space of features x_1 and x_2 are partitioned into four regions. The representation of rectangles as in Figure 5.34a, is useful for two inputs, but more difficult for more than two inputs. As an alternative, a tree can be drawn independent on the number of features (Figure 5.34b).



Nonsmooth response surface for the fake data partitioning as in Figure 5.34.

Classification/regression trees partition the objects on one feature at a time.

To keep the description of the regions simple, binary decision trees with a root, nodes, and leaves, the R_m's, are favored. Each region is split into two further regions over and over again termed a *recursive binary partitioning*. The decision for the binary split is based on a constant, *s,* as shown in Figure 5.34b.

In case of regression trees, the constant could be derived from the averaged response in each region. This will not work for classification trees and, therefore, the relative frequency or proportion, *p,* of *n* responses, y_i, in a node *m,* which represents the region R_m, with class *k* is calculated as follows:

$$p_{mk} = \frac{1}{n_m} \sum_{x_i \in R_m} I(y_i = k) \tag{5.153}$$

I represents the unit step function that gives 1 for $I\ (y_\mathbf{i} = k)$ and 0 for $I\ (y_\mathbf{i} \neq k)$.

Classification of new observations in node *m* to class *k* is obtained by

$$k(m) = \arg\max_k p_{mk} \tag{5.154}$$

that is, assign an observation to class *k* with feature variables (arguments), $\boldsymbol{x}$, for which the proportion of responses, p_{mk}, is maximum.

To build the classification tree for best fit on one or both regions, the variables and the split points have to be found. Best fit is usually determined by minimizing the mean squared error of the residuals of the responses, $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2$. To find the best partition according to this criterion, however, is computationally infeasible. Several greedy algorithms are in use to estimate the tree. Because a large tree might overfit the data tree pruning is performed on the basis of a complexity criterion relying on a

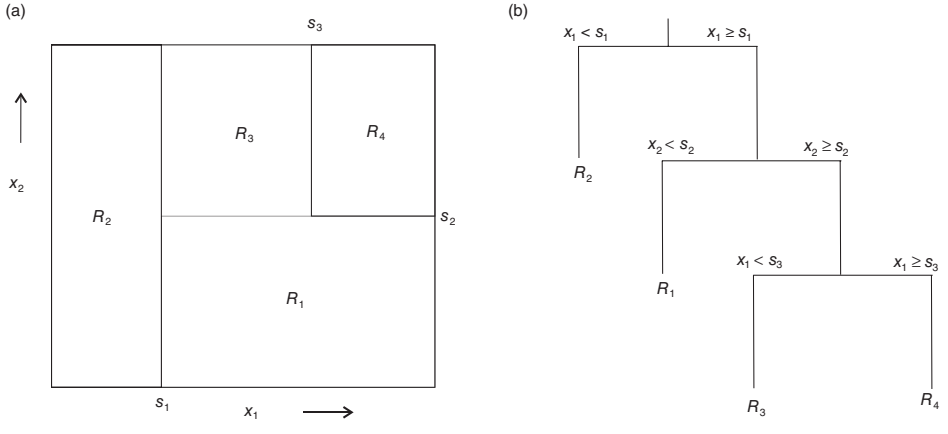


Figure 5.34 Partitioning in CART: The panel (a) demonstrates the partitioning of a two-dimensional feature space into four regions by recursive binary splitting. In the panel (b), the resulting tree is shown.

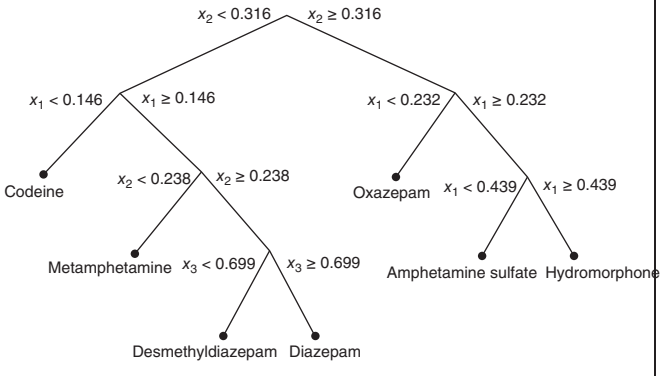
complexity parameter α and the number of terminal nodes, $|T|$:

$$C_\alpha(T) = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2 + \alpha|T| \tag{5.155}$$

An example for building a classification tree is given in Example 5.13 for classification of drugs based on their infrared spectra.

Example 5.13 *Classification Trees*

Spectra in the mid-infrared range were recorded in 70 blood serum samples representing seven classes of the drugs codeine, methamphetamine, desmethyldiazepam, oxazepam, diazepam, amphetamine sulfate, and hydromorphone. For data analysis, the absorbances at 16 wavenumbers were chosen as input to the CART algorithm. The resulting classification tree has the following shape:



All the seven drug classes can be perfectly separated. The assignment of a new drug would be possible by applying Eq. (5.154).

Although CART modeling of the classes might be perfect as in Example 5.13, the prediction of new observations is sometimes less perfect. This might be reasoned by overfitting and some further treatment, especially pruning of a tree, should be applied.

Collapsing any number of nonterminal (internal) nodes is termed *pruning*.

In CART models, no assumptions are necessary regarding the distribution of the input variables as made in many other multivariate methods. Another advantage is the treatment of missing values. In Section 5.1, we learned about column means or random numbers to deal with missing values. CART provides more sophisticated methods for this purpose, for example, by

generating surrogate variables during modeling. Also, nonadditivity of predictors and complex interactions among predictors can be handled with CART. Known disadvantages of CART methods are their instability of tree structure in case of small changes of input data and, furthermore, the fitted response surface lacks smoothness.

Ensemble methods comprise bootstrap aggregating (bagging), Bayesian model combination, bucket of models, stacking, or boosting.

Ensemble Methods

The models of tree-based methods can be improved by ensemble methods, where several different decision trees are aggregated to an ensemble and the slightly differing classification results are averaged. Currently, the most popular ensemble methods are bagging and boosting.

In *bagging*, the original feature matrix, $\boldsymbol{X}$, is used to generate different models by bootstrap sampling. As in cross-validation, in bootstrapping, samples are randomly left out and their responses are predicted by the different models. In contrast to cross-validation, however, the size of the $\boldsymbol{X}$-matrix remains constant because the missing rows are filled up with copies of the remaining samples in the $\boldsymbol{X}$-matrix. Combinations of the predictions by the different models are carried out by a voting system, where the unknown sample is assigned to the class for which most of the votes were obtained. Unfortunately, the bagged model has no longer an easy interpretation as has the single tree because the original structure is lost.

Boosting constructs sequential models by a sample reweighting function altering adaptively the $\boldsymbol{X}$-matrix. A poorly predicted sample is "boosted" by increasing its weight before the construction of the next model; thus, the models concentrate on the poorly predicted samples and generalization is improved by the sequential models. An unknown sample is predicted by all models, the results are weighted, and the outputs are combined again by a voting scheme.

Bagging and boosting of classification trees are explored in Example 5.14.

Example 5.14 *Ensemble Methods*

Here we consider again the 4-class problem used in Example 5.12. CART modeling of this data set by a single tree reveals

decision boundaries as seen in Figure 5.35. The percentage of misclassified objects amounts to 0.75% after resubstitution and to 19.0% after cross-validation, respectively.

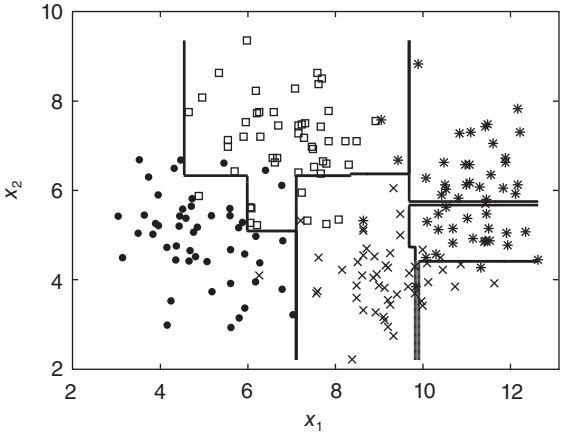


Figure 5.35 Decision boundaries for classification of four classes by CART in a single tree.

Application of ensemble methods implies variation of several parameters in order to optimize the final model. Typical parameters are minimum observations per leaf or per branch node, pruning, and split criterions as well as surrogate decision splits. In this example, the maximum number of decision splits (branch nodes) per layer – a set of nodes that are equidistant from the root node – has been chosen for improving the CART model by the ensemble methods bagging and boosting.

Figure 5.36a demonstrates the fractions of misclassifications in dependence on the maximum number of splits for both resubstitution and cross-validation in a *bagged* CART model. Although the resubstitution error might be close to zero, a more realistic model is based on the maximum number of splits for which a minimal classification error for cross-validation is observed. The smallest error for the bagged trees is found at 16 splits where the cross-validated fraction of misclassification is 12.0% (cf. Figure 5.36b for the decision boundaries).

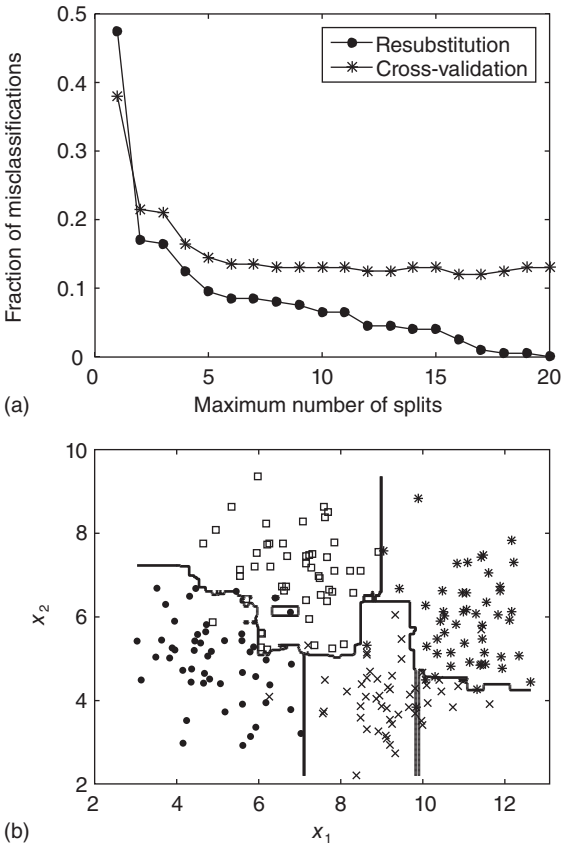


Figure 5.36 Dependence of the fraction of misclassifications in *bagged* CART models on the maximum number of splits (a) and the decision boundaries for 16 splits per layer (b).

In case of *boosting,* here the RUSBoost algorithm was used, the misclassification error versus the maximum number of splits follows a similar trend as in bagging (Figure 5.37a). The minimum for cross-validated misclassifications is at 14 splits per layer (Figure 5.37b) resulting in errors of 0.7% for resubstitution and of 14.0% for cross-validation, respectively. In summary, both ensemble methods improve the number of correct classifications in comparison with a single tree CART model.

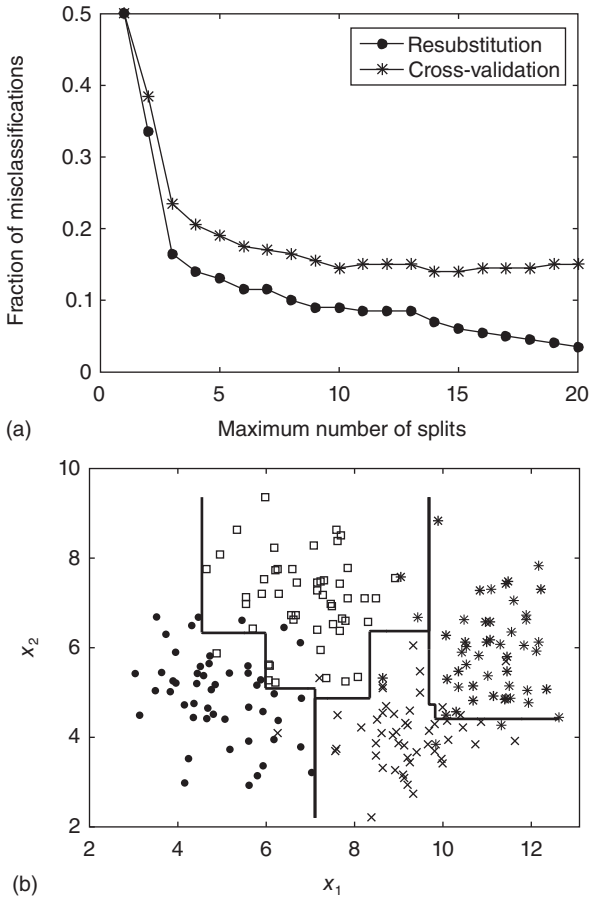


Figure 5.37 Dependence of the fraction of misclassifications in *boosted* CART models on the maximum number of splits (a) and the decision boundaries for 14 splits per layer (b).

In Example 5.15, the performance of different classification methods for a simulated data set in two dimensions is compared for LDA and QDA, SVMs, CART as well as for the k-NN method.

Example 5.15 *Comparison of Classification Methods*

Simulation of a data set in two dimensions consisting of 400 random objects divided equally into two classes revealed the data given in Figure 5.38a.

The two classes represented by filled circles and crosses have been modeled first by LDA. As can be seen in Figure 5.38b,

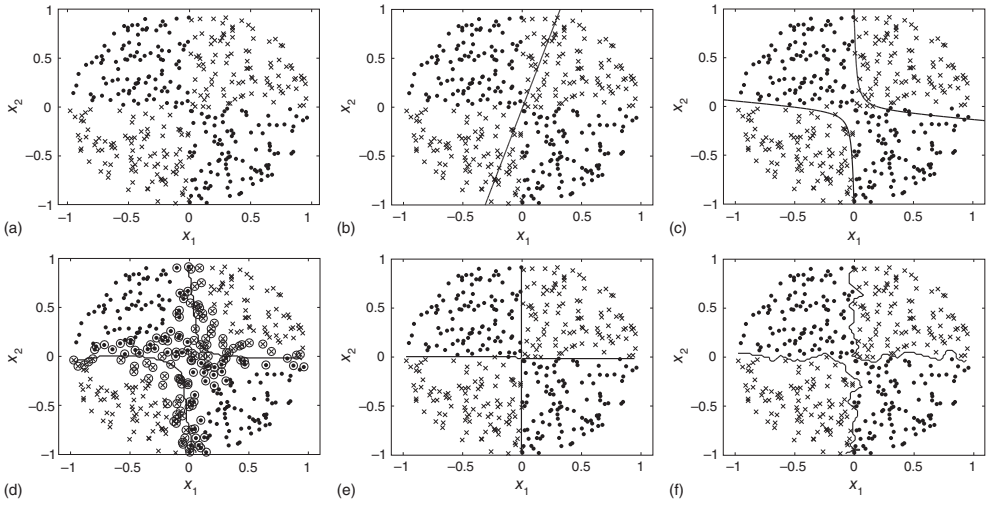


Figure 5.38 Simulated two-class data (a) and their classification by (b) LDA, (c) QDA, (d) SVM, (e) CART, and (f) *k*-NN.

LDA cannot separate these classes, and in case of *resubstitution*, the resulting fraction of misclassified objects is 49%. Much improved decision boundaries are found by using QDA with only 4.75% of misclassified objects (Figure 5.38c). A similar result is obtained by applying SVM with a fraction of misclassifications of 4.50%. The support vectors are labeled by circles in the figure for this classification method (Figure 5.38d). Even better results are obtained for CART and *k*-NN models (Figure 5.38f), where in case of resubstitution only 0.25% and 0% of objects are misclassified, respectively.

However, the instability of tree-based methods implies also here a much higher error in case of *cross-validation* by the simple leave-one-out method, that is, the cross-validated fraction of misclassified objects for CART is with 2.25%, 10 times higher than the resubstitution error. This error can be only reduced if ensemble methods are included in the model building step. A bagged CART model revealed a cross-validation error of only 1.0% (Figure 5.38e). The fraction of misclassifications for the cross-validated models increases for QDA, SVM, and *k*-NN to 5.5%, 5.0%, and 4.75%, respectively. The cross-validated classifications by LDA reveal 58.8% of misclassified objects as expected from the type of data.

General Reading

1. Varmuza, K. and Filzmoser, P. (2009) *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, Boca Raton, FL, Berlin.
2. Malinowski, E.R. (2002) *Factor Analysis in Chemistry*, 3rd edn, John Wiley & Sons, Inc., New York.
3. Frank, I.E. and Todeschini, R. (1994) *The Data Analysis Handbook*, Elsevier, Amsterdam.
4. Smilde, A., Bro, R., and Geladi, P. (2004) *Multi-Way Analysis*, John Wiley & Sons, Ltd., Chichester.
5. Massart, D.L., Vandeginste, B.G.M., Deming, S.N., Buydens, L.C.M., De Jong, S., Lewi, P.J., and Smeyers-Verbeke, J. (1998) *Handbook of Chemometrics and Qualimetrics, Parts A and B*, Elsevier, Amsterdam.
6. Sharaf, M.A., Illman, D.L., and Kowalski, B.R. (1986) *Chemometrics*, Chemical Analysis Series, vol. **82**, John Wiley & Sons, Inc., New York.
7. Friedman, J.H. (1989) *Regularized discriminant analysis. J. Am. Stat. Assoc.*, **84**, 165.
8. Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning*, Springer, New York.
9. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. **3**, Elsevier, Amsterdam.

Questions and Problems

1. Explain the following methods for data preprocessing: centering, range scaling, autoscaling, scaling to variance one, normalization, Fourier transformation (FT), principal component projection, linear transformation, logarithmic transformation.
2. Which rows and/or columns could you eliminate in matrix $\boldsymbol{X}$ in order to avoid correlated, constant, or redundant data in subsequent multivariate analyses?

$$\boldsymbol{X} = \begin{pmatrix} 2 & 1 & 3 & 5 & 7 \\ 4 & 2 & 6 & 5 & 14 \\ 3 & 7 & 4.5 & 5 & 12 \\ 1 & 3 & 1.5 & 5 & 3 \\ 9 & 4 & 13.5 & 5 & 1 \end{pmatrix}$$

   Compute the condition numbers of $\boldsymbol{X}$ before and after preprocessing.
3. Specify the following terms in multivariate analysis: principal component, eigenvector, common and unique factor, score, loading, target vector, latent variable.
4. Which methods can be used to decide on the number of abstract components in a PCA?
5. What is the difference between the principal component and factor analysis?
6. The content of heavy metals in soil samples has been determined at nine different locations. The following table provides the results of the analysis for the elements Zn, Cd, Pb, and Cu, in parts per million:

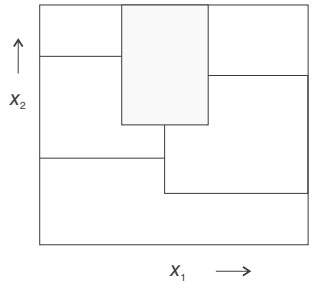
Sample	Zn	Cd	Pb	Cu
1	35.3	0.08	0.25	6.5
2	20.2	1.20	0.52	3.2
3	34.2	0.05	0.28	5.8
4	22.2	1.50	0.48	2.9
5	33.8	0.07	0.26	4.9
6	25.3	0.90	0.60	3.6
7	38.1	2.10	1.20	3.0
8	39.2	1.90	1.50	2.5
9	37.8	2.80	1.40	2.6

   (a) What are the eigenvalues of the data matrix?

(b) How many significant principal components/factors are in the data set if more than 98% of the total variance is explained?
(c) Which sample numbers belong to the found classes?
(d) Assign the following unknown sample to the appropriate class of origin: Zn, 21.8 ppm; Cd, 1.3 ppm; Pb, 0.5 ppm; Cu, 3.3 ppm.

7. Summarize the advantages and disadvantages of direct and inverse multivariate calibration.

8. For solving a classification problem by CART, the following regions are given:



Is this a valid partitioning of regions emerging from a recursive binary splitting?

6
Modeling

Learning Objectives

- To introduce univariate regression analysis for straight-line calibration and empirical model building of analytical relationships
- To understand the usage of the analysis of variance and regression diagnostics
- To model analytical relationships by multiple linear regression analysis, such as in multicomponent analysis, and by target transform factor analysis
- To present nonlinear regression methods based on parametric and nonparametric models.

Models are constructed in analytics to describe the *relationship* between *responses* and *factors.* This is, for example, important for optimization of analytical methods on the basis of response surface methods (cf. Section 4.2). Models are also needed for calibration of analytical methods. There, calibration of a single analyte in dependence on one or several wavelengths might be of interest. If, in the first example, the straight-line model would be adequate, for the second task of multiwavelength spectroscopy, multivariate approaches are needed. Calibrations in the case of unselective analytical methods must also be performed. These methods are termed simultaneous *multicomponent analysis*. In near-infrared (NIR) spectroscopy, the contents of water and protein in whole grain wheat are determined that way.

Calibration and *response surface methods* are indeed the most important applications of regression methods in analytics. Other applications are seen in environmental analysis, where receptor models are developed on the basis of multivariate relationships

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

or for risk assessment of environmental pollution on the basis of typical pollution patterns.

Linear models consist of additive terms, each of which contains only a multiplicative parameter.

Although our world in general cannot be described by simple linear relationships, most of the problems considered in the following can cope with *linear models.* More precisely, models that are linear in the parameters to be estimated are sufficient. The simplest linear model is the straight-line model. It reads for one dependent variable, *y,* and one independent variable, *x,* with the intercept b_0 and the slope b_1:

$$y = b_0 + b_1 x \tag{6.1}$$

However, a quadratic dependence of one variable y on variable x can also be characterized by a linear model, that is,

$$y = b_0 + b_1 x + b_{11} x^2 \tag{6.2}$$

All parameters (b_0, b_1, and b_{11}) can be estimated by the methods of linear algebra. Since the calculation of the parameters in multivariate linear models can be carried out by the same principles as for the straight-line model, we will begin with the problem of univariate linear regression.

6.1 Univariate Linear Regression

Straight-Line Model

The straight-line model (Eq. (6.1)) was used in Section 4.1 as calibration function (cf. Figure 4.2 and Eq. (4.1)). Estimation of the parameters b_0 and b_1 by means of linear regression for n measurements of the pairs of values (x_i, y_i) is done by

$$b_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \tag{6.3}$$

$$b_0 = \overline{y} - b_1 \overline{x} \tag{6.4}$$

where

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6.5}$$

and

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6.6}$$

The variances of the parameters are estimated according to

$$s_{b_0}^2 = \frac{s_y^2 \sum_{i=1}^{n} x_i^2}{n\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{6.7}$$

$$s_{b_1}^2 = \frac{s_y^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{6.8}$$

where s_y^2 characterizes the mean variance of *residuals* as differences between the measured values, $(y_i - \widehat{y}_i)^2$, and the values predicted by the model (Eq. (6.1)), $\widehat{y}_i$. The residual variance is calculated by

$$s_y^2 = \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}{n-p} \quad \text{with} \, p = 2 \tag{6.9}$$

The parameter estimations are frequently given as the standard error, that is, as s_{b_0} or s_{b_1} (cf. Eq. (6.23)).

For *prediction* of an x_0 value from an y_0 value by means of the straight-line model, we have

$$x_0 = \frac{y_0 - b_0}{b_1} \tag{6.10}$$

The standard deviation for prediction by using the straight-line model and performing p parallel measurements with one sample reveals

$$s_0 = \frac{s_y}{b_1}\sqrt{\frac{1}{p} + \frac{1}{n} + \frac{(\overline{y}_0 - \overline{y})^2}{b_1^2 \sum_{i=1}^{n}(x_i - \overline{x})^2}} \tag{6.11}$$

It is important for all kinds of modeling that the estimated parameters are tested for their statistical significance. To derive appropriate tests for the adequacy of a regression model, we need to generalize the straight-line regression.

Generalization of the Straight-Line Model

For generalization of the regression problem, we use matrix notation. The straight-line model

$$y = b_0 \cdot 1 + b_1 x \tag{6.1}$$

then reads

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \tag{6.12}$$

or in abbreviated notion

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} \tag{6.13}$$

The number of rows of the vector of the dependent variable $\boldsymbol{y}$ and of the matrix of the independent variables $\boldsymbol{X}$ corresponds to the number of measurements n. The parameter vector $\boldsymbol{b}$ consists of only two elements in the present case of the straight-line equation.

The parameter elements in vector $\boldsymbol{b}$ are estimated on the basis of the generalized inverse by

$$\boldsymbol{b} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{6.14}$$

We will learn the most appropriate methods for solving Eq. (6.14) in Section 6.2. The statistical tests for the adequacy of the regression model are based on an analysis of variances (ANOVAs) (cf. Section 2.3).

Analysis of Variance (ANOVA)

For an ANOVA, the observed and predicted y values are considered in dependence on the independent variables or factors (cf. Section 2.3).

Partitioning of the variances of a linear regression follows the scheme given in Table 6.1 and Figure 6.1 by the appropriate sums of squares: the total variance of the y values, SS_T, adds up from the sum of squares of the mean, SS_M, and the sum of squares corrected for the mean, SS_{corr}.

The latter sum of squares is composed of the sum of squares of the factors, SS_{fact}, and the sum of squares of the residuals, SS_R. The sum of squares of the residuals is composed of the sum of squares of the lack-of-fit, SS_{lof}, and the sum of squares of pure experimental error, SS_{pe}.

In Example 6.1, we consider the calculation of all these sums of squares for a simple regression problem. The meaning of the different y expressions is illustrated in Figure 6.2.

Table 6.1 Computation of the sums of squares (*SS*) for a complete ANOVA in linear regression.

Sum of squares	Matrix operation	Calculation	Degrees of freedom
SS_T, total	$\boldsymbol{y}^T\boldsymbol{y}$	$\sum_{i=1}^{n} y_i^2$	n
SS_M, mean	$\overline{\boldsymbol{y}}^T\overline{\boldsymbol{y}}$	$n\overline{y}^2$	1
SS_{corr}, corrected for the mean	$(\boldsymbol{y}-\overline{\boldsymbol{y}})^T(\boldsymbol{y}-\overline{\boldsymbol{y}})$	$\sum_{i=1}^{n} (y_i-\overline{y})^2$	$n-1$
SS_{fact}, factors	$(\hat{\boldsymbol{y}}-\overline{\boldsymbol{y}})^T(\hat{\boldsymbol{y}}-\overline{\boldsymbol{y}})$	$\sum_{i=1}^{n} (\hat{y}_i-\overline{y})^2$	$p-1$
SS_R, residuals	$(\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}})$	$\sum_{i=1}^{n} (y_i-\hat{y}_i)^2$	$n-p$
SS_{lof}, lack-of-fit	$(\boldsymbol{j}-\hat{\boldsymbol{y}})^T(\boldsymbol{j}-\hat{\boldsymbol{y}})$	$\sum_{i=1}^{n} (\overline{y}_i-\hat{y}_i)^2$	$f-p$
SS_{pe}, pure experimental error	$(\boldsymbol{y}-\boldsymbol{j})^T(\boldsymbol{y}-\boldsymbol{j})$	$\sum_{i=1}^{n} (y_i-\overline{y}_i)^2$	$n-f$

$\boldsymbol{j}$ – This vector contains y values averaged at each observation point i, f – number of different realizations of the independent variables (independent factor combinations), n – number of measurements, and p – number of parameters, $\overline{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n} y_i$ total mean.

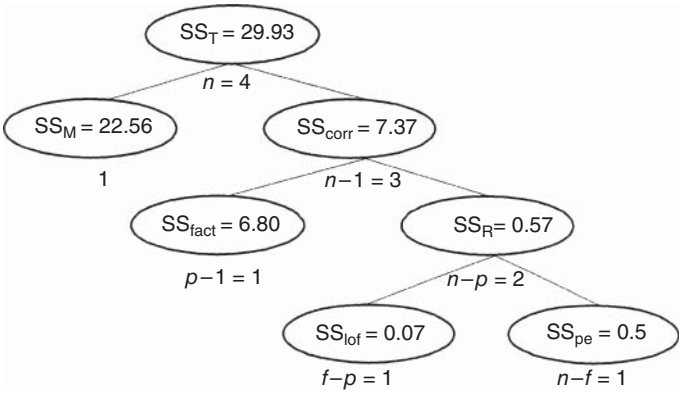


Figure 6.1 Illustration of analysis of variance (ANOVA) for linear regression on the data in Table 6.2. See Table 6.2 for abbreviations.

The *experimental error* can only be estimated if at least one replication is performed at an independent factor combination.

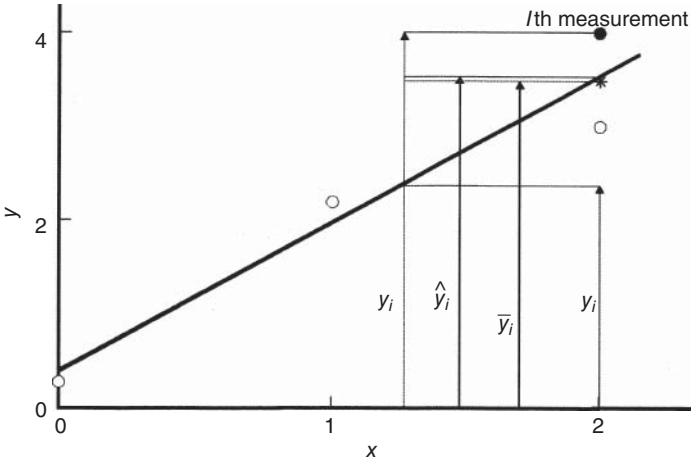


Figure 6.2 Plot of the x–y data in Table 6.2 in connection with an ANOVA according to Table 6.1.

Example 6.1 *ANOVA in Regression Analysis*

In Table 6.2, for four measurements, the x–y data are given. The data are to be fitted by the straight-line model. After that, all sums of squares are to be calculated according to Table 6.1, in order to use them for subsequent tests for adequacy of the model.

Table 6.2 x–y data.

No.	x	y
1	0	0.3
2	1	2.2
3	2	3
4	2	4

Insertion of the data in the model for the straight line (6.12) reveals

$$\begin{pmatrix} 0.3 \\ 2.2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \tag{6.15}$$

Computation of the parameter vector $\boldsymbol{b}$ is done according to Eq. (6.14)

$$\begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \boldsymbol{b} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 2 \end{pmatrix} \right)^{-1}$$
$$\times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 \end{pmatrix} \begin{pmatrix} 0.3 \\ 2.2 \\ 3 \\ 4 \end{pmatrix}$$
$$= \begin{pmatrix} 4 & 5 \\ 5 & 9 \end{pmatrix}^{-1} \begin{pmatrix} 9.5 \\ 16.2 \end{pmatrix} = \begin{pmatrix} 0.818 & -0.454 \\ -0.454 & 0.363 \end{pmatrix} \begin{pmatrix} 9.5 \\ 16.2 \end{pmatrix} = \begin{pmatrix} 0.41 \\ 1.57 \end{pmatrix} \tag{6.16}$$

The straight-line model is then

$$y = 0.41 + 1.57x$$

Based on the model, the sums of squares can be determined by performing an ANOVA:

$$\mathrm{SS_T} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = \begin{pmatrix} 0.3 & 2.2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 0.3 \\ 2.2 \\ 3 \\ 4 \end{pmatrix}$$
$$= \sum_{i=1}^{n} y_i^2 = 0.3^2 + 2.2^2 + 3^2 + 4^2 = 29.93$$

The variance of the y values results from the *total sum of squares*, $\mathrm{SS_T}$, that is:

The sum of squares of the *mean,* $\mathrm{SS_M}$, is calculated from the total mean for all y values by

$$\mathrm{SS_M} = n\bar{y}^2 = 4{\cdot}2.375^2 = 22.56$$

Correspondingly, we obtain for the sum of squares *corrected for the mean*

$$\mathrm{SS_{corr}} = \sum_{i=1}^{n} (y_i - \bar{y})^2 = (0.3 - 2.375)^2 + (2.2 - 2.375)^2$$
$$+ (3 - 2.375)^2 + (4 - 2.375)^2 = 7.37$$

The influence of the *factors* is reflected by the sum of squares due to the factors, SS_{fact}:

$$\mathrm{SS_{fact}} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 = (0.41 - 2.375)^2 + (1.98 - 2.375)^2$$
$$+ (3.55 - 2.375)^2 + (3.55 - 2.375)^2 = 6.80$$

The difference between the observed and predicted y values leads to the sum of squares of the *residuals,* SS_R (cf. Eq. (6.9)):

$$SS_R = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = (0.3 - 0.41)^2 + (2.2 - 1.98)^2 + (3 - 3.55)^2 + (4 - 3.55)^2 = 0.57$$

If the intercept is included in the model, the *sum of residuals* equals zero.

Within the residuals, the lack-of-fit to the model and the experimental error are to be accounted for. The sum of squares due to lack-of-fit, SS_{lof}, provides

$$SS_{lof} = \sum_{i=1}^{n} (\overline{y}_i - \hat{y}_i)^2 = (0.3 - 0.41)^2 + (2.2 - 1.98)^2 + (3.5 - 3.55)^2 + (3.5 - 3.55)^2 = 0.07$$

The sum of squares due to the *experimental error*, SS_{ee}, is calculated by

$$SS_{pe} = \sum_{i=1}^{n} (y_i - \overline{y}_i)^2 = (0.3 - 0.3)^2 + (2.2 - 2.2)^2 + (3 - 3.5)^2 + (4 - 3.5)^2 = 0.5$$

Table 6.3 comprises the calculation in an ANOVA table.

Table 6.3 ANOVA table for linear regression of the data in Table 6.2.

Source of variation	SS	*df*	MSS	*F* value	*p* value
Total, SS_G	29.93	4	7.48	—	—
Mean, SS_M	22.56	1	22.56	—	—
Corrected for the mean, SS_{ccrr}	7.37	3	2.47	—	—
Factors, SS_{fact}	6.80	1	6.80	24.06	0.0391
Residuals, SS_R	0.57	2	0.283	—	—
Lack-of-fit, SS_{lof}	0.07	1	0.07	0.14	0.779
Pure experimental error, SS_{pe}	0.50	1	0.5	—	—

SS Sum of squares, MSS mean sum of squares, and *df* degrees of freedom.

Coefficient of Determination and Correlation Coefficient

If the regression parameters are estimated by the least squares method, the square root of the *coefficient of determination* and the multiple c*orrelation coefficient* will be identical.

The coefficient of determination is given by the ratio of the sums of squares due to the factors and due to the sum of squares corrected for the mean:

$$R^2 = \frac{SS_{fact}}{SS_{corr}} \tag{6.17}$$

The coefficient of determination describes the fraction of the sum of squares due to the factors in relation to the sum of squares corrected for the mean. The square root of the coefficient of determination reveals the multiple correlation coefficient:

$$r = \sqrt{\frac{SS_{fact}}{SS_{corr}}} \tag{6.18}$$

The sign of the correlation coefficient is given by the slope b_1.

Example 6.2 *Correlation Coefficient*

For the regression problem in Example 6.1, the coefficient of determination we obtain according to Eq. (6.17) is

$$R^2 = \frac{SS_{fact}}{SS_{corr}} = \frac{6.80}{7.37} = 0.923$$

The two factors explain 92.3% of the sum of squares corrected for the mean.

The correlation coefficient is calculated according to Eq. (6.18):

$$r = \sqrt{\frac{SS_{fact}}{SS_{corr}}} = \sqrt{\frac{6.80}{7.37}} = 0.961$$

Note that the correlation coefficient does not provide a realistic picture for the explanation of the variance due to the factors. In our example, only 92.3% are explained, although the correlation coefficient is 0.961. An r value of, for example, 0.7 would mean that only 49% of the SS_{corr} can be explained.

Test for Adequacy of the Model

To test for the fit of data to the model, two F tests are to be carried out: the F test for goodness-of-fit and the F test for lack-of-fit. For this, the sum of squares normalized to the degrees of freedom is applied, given in Table 6.3 as the mean sum of squares (MSS).

As measure for the quality of fit, the F value for the goodness-of-fit is calculated by the linear model with the intercept, b_0, by

$$F(p-1, n-p) = \frac{SS_{fact}/(p-1)}{SS_R/(n-p)} = \frac{MSS_{fact}}{MSS_R} \tag{6.19}$$

The MSS must be greater than the residuals if the factors in the model influence the y values definitely. If an appropriate model has been found, the F test will be *significant* at the given α level.

If the experimental error is comparably large for all factor combinations, the data are termed *homoscedastic*. In the case of *heteroscedastic data,* the errors differ at different factor combinations.

The *lack-of-fit test* is based on the comparison of the MSS due to the model and the experimental error:

$$F(f-p, n-f) = \frac{\mathrm{SS}_{\mathrm{lof}}/(f-p)}{\mathrm{SS}_{\mathrm{pe}}/(n-f)} = \frac{\mathrm{MSS}_{\mathrm{lof}}}{\mathrm{MSS}_{\mathrm{pe}}} \tag{6.20}$$

In regression analysis, the term *multiple realizations* of an x value is common instead of factor combinations.

Note that this test is only applicable if the number of independent factor combinations, f, is greater than the number of model parameters, p. In addition, at least one replicate measurement is required at the same factor combination to guarantee that n becomes greater than f and the sum of squares due to the pure experimental error can be estimated.

The lack-of-fit may *not be significant* if an appropriate model has been found.

Example 6.3 *Tests for Model Adequacy*

The straight-line model in Example 6.1 is to be tested for its adequacy. First, the F test for goodness-of-fit is carried out according to Eq. (6.19):

$$F(p-1, n-p) = \frac{\mathrm{MSS}_{\mathrm{fact}}}{\mathrm{MSS}_{\mathrm{R}}} = \frac{6.80}{0.283} = 24.0$$

The critical F value at the significance level of 0.05 is $F(0.95; 1, 2) = 18.51$ (cf. Table A.4). As a consequence, the calculated F value is greater than the critical one and the goodness-of-fit test is derived to be statistically significant.

For very small experimental error, the F value in the lack-of-fit test might be extraordinarily large and the test becomes significant. It is to be decided then whether this result is also *practically significant.*

This result can also be derived from the ANOVA Table 6.3. There the p value is given with 0.0391. This value is lower than the significance level of 0.05 $(1-0.95)$ considered here, and therefore, the test is significant. In other words, the risk that all of the factors are different from zero is 3.91%. Since in our model, we only have a single factor, x, we can assume with $(100-3.91) = 96.09\%$ probability that the effect of x is realistic.

In the second step, the F test for lack-of-fit (Eq. (6.20)) is applied:

$$F(f-p, n-f) = \frac{\mathrm{MSS}_{\mathrm{lof}}}{\mathrm{MSS}_{\mathrm{pe}}} = \frac{0.07}{0.50} = 0.14$$

For the tabulated critical F value, we obtain $F(0.95; 1, 1) = 161$ (cf. Table A.4). The F test for lack-of-fit is not significant, since the calculated F value is smaller than the critical value. The significance level is only 0.779, as the p value shows in Table 6.3.

Confidence Intervals

Confidence Interval for the Parameters The error for estimation of the parameters of the straight line in Eq. (6.1) has already been used in the form of the variances for the parameter estimations (Eqs. (6.7) and (6.8)).

More generally, the errors of parameter estimation can be calculated on the basis of the variance–covariance matrix (Eq. (5.10)). The variance–covariance matrix is computed here on the basis of the MSS for the pure experimental error as follows:

$$\boldsymbol{C} = \mathrm{MSS}_{\mathrm{pe}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1} \tag{6.21}$$

where, according to Table 6.1,

$$\mathrm{MSS}_{\mathrm{pe}} = \frac{\mathrm{SS}_{\mathrm{pe}}}{n-f}$$

For the case that the sum of squares due to lack-of-fit, $\mathrm{SS}_{\mathrm{lof}}$, is small, the variance of the residuals, s_{R}^2, can be used instead of the variance due to the experimental error. The residual variance corresponds to the MSS of the residuals in Table 6.1, that is,

$$s_{\mathrm{R}}^2 = \mathrm{MSS}_{\mathrm{R}} = \frac{\mathrm{SS}_{\mathrm{R}}}{n-p}$$

The special case for $p = 2$ has already been used in Eq. (6.9). The variance–covariance matrix is then computed by

$$\boldsymbol{C} = s_{\mathrm{R}}^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1} \tag{6.22}$$

The diagonal of the variance–covariance matrix consists of the variances for the parameter estimates and the off-diagonal of those of the related covariances. In the case of the straight-line model with the parameters b_0 and b_1, the corresponding matrix is

$$\boldsymbol{C} = \begin{pmatrix} s_{b_0}^2 & s_{b_0 b_1}^2 \\ s_{b_1 b_0}^2 & s_{b_1}^2 \end{pmatrix} \tag{6.23}$$

For the confidence interval Δb related to the parameter b, we obtain by means of the F-statistic for a given significance level α

$$b \pm \sqrt{F(\alpha; 1, n-p)s_b^2} \tag{6.24}$$

The covariance matrices of Eq. (6.21) or (6.22) can be used analogously for calculations of confidence intervals, if more than two parameters are to be determined.

Example 6.4 *Confidence Interval*

For the regression parameters of the data in Example 6.1, the confidence intervals of the parameters according to Eq. (6.22) are to be estimated. The required inversion of the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ has already been introduced in Eq. (6.16). With the MSS of residuals of Table 6.3, we obtain

$$\boldsymbol{C} = s_{\mathrm{R}}^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1} = 0.283 \begin{pmatrix} 0.818 & -0.454 \\ -0.454 & 0.363 \end{pmatrix}$$
$$= \begin{pmatrix} 0.231 & -0.128 \\ -0.128 & 0.103 \end{pmatrix}$$

The standard error for the parameters of the regression equation reveals

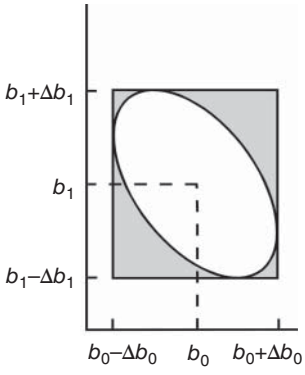
$$s_{b_0} = \sqrt{s_{b_0}^2} = \sqrt{0.231} = 0.481$$

$$s_{b_1} = \sqrt{s_{b_1}^2} = \sqrt{0.103} = 0.321$$

The confidence interval of the parameters at a significance level of 0.05 with $F(0.05;\, 1,\, 2) = 18.51$ (cf. Table A.4) is calculated by

$$b_0 \pm \sqrt{F(\alpha; 1, n-p)s_{b_0}^2} = 0.41 \pm 2.07$$

$$b_1 \pm \sqrt{F(\alpha; 1, n-p)s_{b_1}^2} = 1.57 \pm 1.38$$



We should be aware that the estimations of the confidence intervals in the given way will only be valid if the parameters are *independent* of each other. All elements in the off-diagonals of the covariance matrix in Eq. (6.23) need to be zero. In the case of two parameters, the confidence intervals describe a square (cf. figure in the margin). If dependences exist between parameters, then an ellipse is obtained for the confidence interval of the parameters. The larger the elements in the off-diagonals in Eq. (6.23), the more pronounced deviations from the squared shape of the confidence intervals are to be expected.

Confidence Bands The previous considerations can also be exploited for computation of the confidence interval for the prediction of a y value, y_0, at a given x_0 value. The predicted value

and the corresponding confidence interval are given by

$$y_0 = \boldsymbol{x}_0\boldsymbol{b} \pm \sqrt{F(\alpha; 1, n-p)s_R^2(1 + \boldsymbol{x}_0(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0^T)} \qquad (6.25)$$

In the case of the straight-line model, the vector $\boldsymbol{x}$ (dimension $1 \times p$) consists only of two elements (cf. Eq. (6.12)), that is,

$$\boldsymbol{x}_0 = \begin{pmatrix} 1 & x_0 \end{pmatrix}$$

The prediction of a single mean from several y values at a given factor combination is feasible with modification of Eq. (6.25). For m new y values, we obtain for prediction of the mean and its confidence interval

$$\overline{y}_0 = \boldsymbol{x}_0\boldsymbol{b} \pm \sqrt{F(\alpha; 1, n-p)s_R^2\left(\frac{1}{m} + \boldsymbol{x}_0\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{x}_0^T\right)} \qquad (6.26)$$

For very large values of m, Eq. (6.26) simplifies to Eq. (6.27):

$$\widehat{y}_0 = \boldsymbol{x}_0\boldsymbol{b} \pm \sqrt{F(\alpha; 1, n-p)s_R^2(\boldsymbol{x}_0(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x}_0^T)} \qquad (6.27)$$

By means of Eqs. (6.25) – (6.27), the confidence bands for prediction of the y values along the independent variables can be plotted. This is demonstrated in Figure 6.3 for the prediction of a single y value and for a mean from many y values on the basis of the data in Example 6.1.

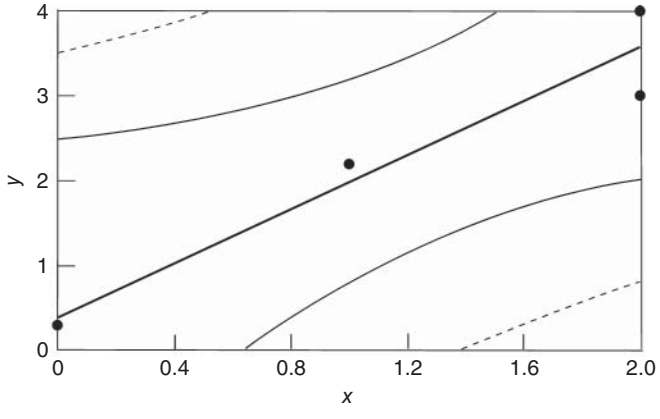


Figure 6.3 Confidence bands for the prediction of individual y values (*broken lines,* Eq. (6.25)) and for the mean from many y values (*solid lines,* Eq. (6.27)) for the data in Table 6.2 of Example 6.1.

Example 6.5 *Confidence Interval for Prediction of y Values*

Based on the straight-line model in Example 6.1,

$$y = 0.41 + 1.57x$$

The y value is to be predicted and the corresponding confidence interval is to be estimated for a value of $x = 0.9$. The confidence interval is to be given for a single value as well as for a mean from several y values. The variance for the residuals, s_R^2, is 0.283 (cf. Example 6.4). The expression $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ has already been computed in Eq. (6.16). We choose a significance level of $\alpha = 0.05$, that is, an F value of $F(0.05; 1, 2) = 18.51$ (cf. Table A.4) is to be used.

According to Eq. (6.25), we calculate for the single y value

$$y_0 = \begin{pmatrix}1 & 0.9\end{pmatrix}\begin{pmatrix}0.41\\1.57\end{pmatrix} \pm \sqrt{18.51 \cdot 0.283 \left(\begin{pmatrix}1 & 0.9\end{pmatrix}\begin{pmatrix}0.818 & -0.454\\-0.454 & 0.363\end{pmatrix}\begin{pmatrix}1\\0.9\end{pmatrix}\right)}$$

$$y_0 = 1.82 \pm 2.60$$

For prediction of the means from several y values, Eq. (6.27) is applied:

$$\hat{y}_0 = \begin{pmatrix}1 & 0.9\end{pmatrix}\begin{pmatrix}0.41\\1.57\end{pmatrix} \pm \sqrt{18.51 \cdot 0.283 \left(\begin{pmatrix}1 & 0.9\end{pmatrix}\begin{pmatrix}0.818 & -0.454\\-0.454 & 0.363\end{pmatrix}\begin{pmatrix}1\\0.9\end{pmatrix}\right)}$$

$$\hat{y}_0 = 1.82 \pm 1.24$$

As expected for prediction of a y value as the mean of several observations, a narrower confidence interval results.

Residual Analysis

For graphical inspection of regression models, the analysis of residuals is applicable. The residual, e_i, denotes the difference between the observed value, y_i, and the value estimated by the model, $\hat{y}_i$. For n observations, we get

$$e_i = y_i - \hat{y}_i \quad \text{with } i = 1, n \tag{6.28}$$

In the case of a valid model, the residuals describe the random error of the regression model. The straight-line model in Eq. (6.13)

is therefore formulated as follows:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \tag{6.29}$$

where $\boldsymbol{e}$ represents the $1 \times n$ dimensional vector of the random error.

By means of residual analysis, the assumptions for the linear regression as well as the deviations from the model can be checked.

The most important prerequisites for linear regression are as follows:

- The error of the independent variable, *x,* can be neglected, that is, x is fixed and only the dependent variable, *y,* is erroneous.
- The y values are independent of each other and normally distributed.
- The variances of y values are comparable for all x values, that is, they are homoscedastic.
- The residuals are, then, independent, normally distributed, and homoscedastic.

If the residuals are plotted in a *histogram,* for large n, the shape of a normal distribution results. From plotting the residuals in dependence on the *run order,* trends can be derived (Figure 6.4a).

In addition, the residuals are evaluated in dependence on the predictor variable, *x,* the observed variable, *y,* or the predicted variable, $\widehat{y}$. The latter possibilities reveal similar information. They will be explained in the following for plotting the residuals in dependence on the independent variable, *x.*

- In the case of changing variance of the y values (heteroscedasticity), different types of residual bands result (Figure 6.4b). For treating those data, a transformation of y values or weighted regression is required.
- A missing linear parameter representing the effect of a linear factor is recognized by linearly ascending or descending residuals (Figure 6.4c).
- Incomplete models might also be reasoned by effects of higher order. In Figure 6.4d, this is demonstrated for the lack of a quadratic term. The residuals show then the form of a parabola.

Apart from the analysis of residuals, the recognition of outliers and of influential observations is important for the selection of a regression model. We will raise those questions for the generalized regression diagnostics in Section 6.2.

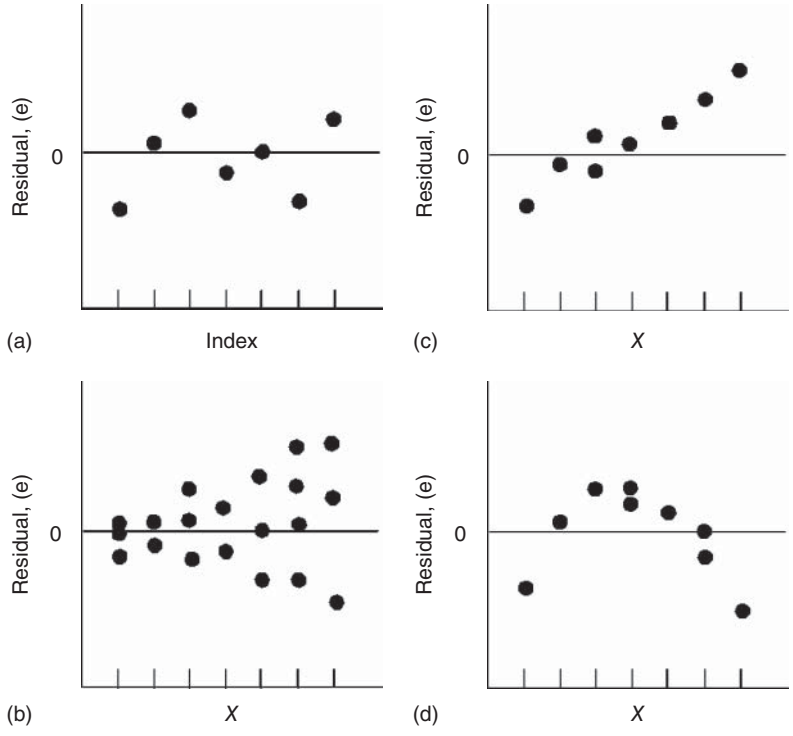


Figure 6.4 Residual analysis in linear regression. (a) Time-dependent observations. (b) Heteroscedasticity. (c) Linear effects. (d) Quadratic effect.

Weighted and Robust Regression

In residual analysis, as already mentioned, one prerequisite of conventional regression is the comparably large error in the y direction for all realizations of x (homoscedasticity). This means that each point on the straight line has the same weight in the regression analysis. In the case of heteroscedastic data, the observations will have different weights in calculating the straight line. In order to make the regression less sensitive to small deviations from the distributional assumptions, the influence of observations with large residuals has to be weighted down.

Weighted Regression To adjust the weighting, the analyst can frequently simply use his expertise. For example, the error in a calibration in y direction may increase with increasing x value, that is, the relative error is constant. Appropriate weights, w_i, would be in this case:

$$w_i = \frac{1}{x_i} \tag{6.30}$$

To weight variances changing at the different observation points x_i, a weight based on the reciprocal standard deviation (variance) could be applied:

$$w_i = \frac{1}{s_{y_i}^2} \tag{6.31}$$

The weights are usually normalized to one, that is, for n observations, we obtain

$$w_i' = \frac{1}{\sum_{i=1}^{n} w_i} \tag{6.32}$$

The model for the straight line is that given in Eq. (6.1):

$$y = b_0 + b_1 x$$

The intercept guarantees that the straight line need not pass the origin of the coordinate system, but passes the centroids of the variables, that is, $\overline{x}$ and $\overline{y}$. For estimation of the parameters for a weighted regression, first the weighted centroids are calculated as follows:

$$\overline{y}_w = \frac{1}{n}\sum_{i=1}^{n} w_i y_i \tag{6.33}$$

$$\overline{x}_w = \frac{1}{n}\sum_{i=1}^{n} w_i x_i \tag{6.34}$$

The regression parameters are then calculated by

$$b_1 = \frac{\sum_{i=1}^{n} w_i x_i y_i - n\overline{x}_w \overline{y}_w}{\sum_{i=1}^{n} w_i x_i^2 - n(\overline{x}_w)^2} \tag{6.35}$$

$$b_0 = \overline{y}_w - b_1 \overline{x}_w \tag{6.36}$$

For the standard deviation in the case of prediction from x_0 values based on p parallel measurements of y_0 values, it is valid for the weighted case (cf. Eq. (6.11)) that

$$s_0 = \frac{s_{y_w}}{b_1}\sqrt{\frac{1}{p} + \frac{1}{n} + \frac{(\overline{y}_0 - \overline{y}_w)^2}{b_1^2\left(\sum_{i=1}^{n} w_i x_i^2 - n\overline{x}_w\right)}} \tag{6.37}$$

where

$$s_{y_w} = \sqrt{\frac{\left(\sum_{i=1}^{n} w_i y_i^2 - b_1^2\left(\sum_{i=1}^{n} w_i x_i^2 - n\bar{x}_w^2\right)\right)}{n-2}} \tag{6.38}$$

As result of a weighted regression, the standard deviations for the parameters should decrease compared to a conventional regression. This is advantageous, for example, for better predictions of y values.

Example 6.6 *Weighted Regression*

The x–y data in Figure 6.5 show an increasing random error in the y direction with ascending x values. To fit those data, conventional and weighted regression is to be carried out.

In the case of *conventional* regression, we obtain the following model, with the corresponding standard errors of the parameters, for the straight-line regression:

$$y = (1.24 \pm 0.71) + (0.732 \pm 0.091)x$$

For weighted regression, the reciprocal x values are used as weights according to Eq. (6.30). The following regression model results:

$$y = (1.08 \pm 0.33) + (0.754 \pm 0.060)x$$

The standard errors for estimation of the two parameters of the straight-line model were obviously decreased by applying weighted regression.

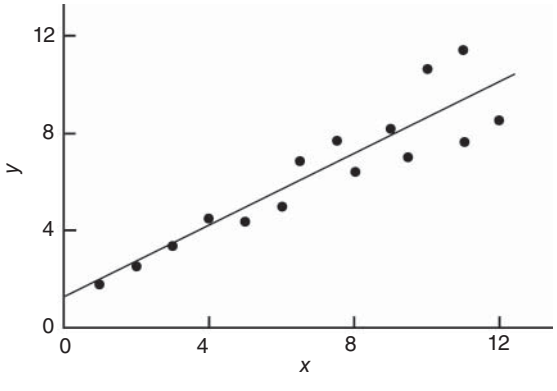


Figure 6.5 x–y-Values for the case of heteroscedastic data.

Robust Regression Robust regression is based at an iterative weighting of observations. As appropriate weights, Tukey suggests

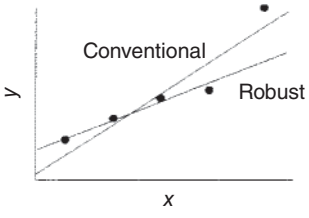
$$w_i = \begin{cases} 1 - \left(\frac{e_i}{kS^2}\right) & |e_i| < kS \\ 0 & |e_i| > kS \end{cases} \tag{6.39}$$

where e_i is again the residual of observation *i;* k is a constant, which controls the weighting of the residuals; and S is a scaling factor, for example,

$$S = \text{median}\{|e_i|\} \tag{6.40}$$

Computation is started by conventional regression analysis. Subsequently, the residuals are determined and the weights for a value of $k > 1$ are calculated for each observation. Regression is repeated as long as the parameters change only by a predefined small amount. The smaller the value for *k,* the more the residuals are weighted down.

Robustness means tolerance against outliers, that is, a model that describes the majority of data well is to be found.



6.2 Multiple Linear Regression

In the case of multivariate modeling, several independent as well as several dependent variables may operate. Out of the many regression methods, we will learn about the conventional method of ordinary least squares (OLS) as well as methods that are based on biased parameter estimations reducing simultaneously the dimensionality of the regression problem, that is, principal component regression (PCR) and the partial least squares (PLS) method.

As an example of the application of these methods, spectrometric multicomponent analysis will be considered, leading to an introduction to regression diagnostics in multiple linear regression.

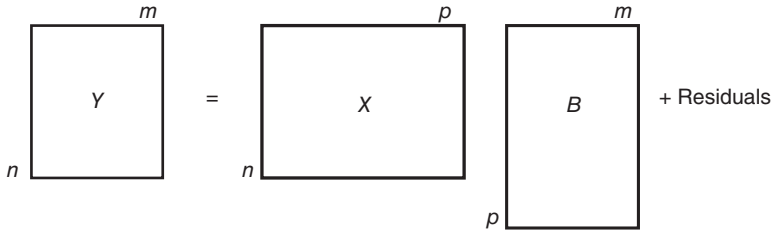
Ordinary Least Squares Regression

The general least squares problem that relates a matrix of dependent variables $\boldsymbol{Y}$ to a matrix of independent variables $\boldsymbol{X}$ can be stated as follows:

$$\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & & & \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & & & \\ b_{p1} & b_{p2} & \cdots & b_{pm} \end{pmatrix}$$

In matrix notation, we get

$$\boldsymbol{Y} = \boldsymbol{XB} \tag{6.41}$$



where $\boldsymbol{Y}$ is the $n \times m$ matrix of dependent variables, $\boldsymbol{X}$ the $n \times p$ matrix of independent variables, $\boldsymbol{B}$ the $p \times m$ matrix of regression parameters, and *residuals* are the differences between measured and predicted data, that is, $\boldsymbol{Y} - \boldsymbol{XB}$.

For example, in multivariate calibration, this equation will be used to model the concentrations of m constituents in n samples ($\boldsymbol{Y}$ matrix) on the n spectra recorded at p wavelengths ($\boldsymbol{X}$ matrix).

In OLS, the number of columns in the $\boldsymbol{X}$ matrix is maintained. As an example, the set of linear algebraic equations for the first column of Eq. (6.41) resembles the following:

OLS is synonymous with the following terms: least squares regression, linear least squares regression, multiple least squares regression, multivariate least squares regression.

$$\begin{aligned} y_{11} &= b_{11}x_{11} + b_{12}x_{12} + \cdots + b_{p1}x_{1p} \\ y_{21} &= b_{11}x_{21} + b_{12}x_{22} + \cdots + b_{p1}x_{2p} \\ &\vdots \\ y_{n1} &= b_{11}x_{n1} + b_{12}x_{n2} + \cdots + b_{p1}x_{np} \end{aligned} \tag{6.42}$$

OLS provides the *best linear unbiased estimator* (BLUE) that has the smallest variance among all linear and unbiased estimators.

Parameter Estimation

Usually, the matrix of the independent variables, $\boldsymbol{X}$, is not square, so that the regression parameters $\boldsymbol{B}$ have to be estimated by the *generalized inverse*. $\boldsymbol{B}$ is given by

$$\boldsymbol{B} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} \tag{6.43}$$

In principle, this equation could be solved by directly inverting the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. This, however, will only work if no linear dependences are valid and the system is, in a mathematical sense, well conditioned. The conditioning of the system is given by the *condition number:*

$$\mathrm{cond}(\boldsymbol{B}) = \|\boldsymbol{C}\| \cdot \|\boldsymbol{B}^{-1}\| \tag{6.44}$$

where $\|\boldsymbol{B}\|$ is the norm of matrix $\boldsymbol{B}$.

The matrix norm of $\boldsymbol{B}$ is computed as its largest singular value (square root of eigenvalue λ) and the norm of $\boldsymbol{B}^{-1}$ as the reciprocal

smallest singular value of $\boldsymbol{B}$, that is,

$$\text{cond}(\boldsymbol{B}) = \sqrt{\lambda_{\max}} \cdot \frac{1}{\sqrt{\lambda_{\min}}} \tag{6.45}$$

This definition holds for exactly determined systems where the number of rows in $\boldsymbol{X}$ is equal to the number of columns, that is, $n = p$ (cf. Eq. (6.41)). In the case of overdetermined systems, with $n > p$, the condition number is obtained from

$$\text{cond}(\boldsymbol{B}) = [\text{cond}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})]^{1/2} \tag{6.46}$$

Well-conditioned systems have condition numbers close to 1. A matrix is singular if its condition number is infinite, and it is ill-conditioned if its condition number is too large, that is, if its reciprocal approaches the machine's floating point precision.

Linear dependences between rows or columns lead to singularities. This might even happen if there is no exact linear dependency, and round-off errors in the machine render some of the equations linearly dependent.

Solution Methods

Typical procedures to solve the OLS problem are Gaussian elimination and Gauss–Jordan elimination. More efficient solutions are based on decomposition of the $\boldsymbol{X}$ matrix by algorithms, such as LU decomposition, Householder reduction, or singular value decomposition (SVD). One of the most powerful methods, SVD, is outlined as follows (cf. Section 5.2 and "Biased Parameter Estimations: PCR and PLS" Section).

Significance of Parameters

To test for the significance of the parameters of a model, we have already used the F test for goodness-of-fit and for lack-of-fit (Section 6.1). In this type of test, a whole set of parameters is investigated.

In statistical programs, the test for *individual parameters* on the basis of the t or F statistics can also be found. The null hypothesis is that the considered parameter differs only randomly from 0. The tests are based on the confidence intervals for the considered parameters that include the corresponding value $b_i = 0$.

Too many parameters in a model lead to *overfitting* of the experimental observations.

For a t-test on the parameters b_0 of the straight-line model in Eq. (6.1), we get

$$t = \frac{|b_0 - 0|}{s_{b_0}} \tag{6.47}$$

where s_{b_0} is the standard error for the parameter (Eq. (6.7)). The calculated t value is then compared with the critical value at the predefined significance level as discussed in Section 2.2.

As long as the model consists of only a single parameter, this kind of testing is adequate. In the case of several parameters, the risk for applying a one-parameter test is no longer simply α. The probability of rejecting at least one null hypothesis falsely, if all k null hypotheses were true, is $[1-(1-\alpha)^k]$. This means, for example, if the model has two parameters, the risk at significance level of $\alpha = 0.05$ is accordingly $[1-(1-0.05)^2] = 0.0975$ or, in the case of three parameters, $[1-(1-0.05)^3] = 0.1426$.

The F tests demonstrated in Section 6.1 deal with a whole set of parameters. The selected significance level is therefore valid independent of the number of parameters in the model. However, one should bear in mind that, with those tests, by rejecting the null hypothesis, the parameters different from zero cannot be recognized individually.

Prediction

In regression analysis, the dependent variable y is also called a *response* variable and the independent variable x is denoted as *predictor* variable or *regressor*.

Modeling of analytical relationships by estimating the regression parameters in OLS is one of the objectives. Most often, in a second step, the model parameters are used to predict some unknown x or y values from the measured y or x values, for example, in multivariate calibration, the concentrations are predicted from the recorded spectra.

Prediction of a single $\boldsymbol{y}_0$ vector from an $\boldsymbol{x}_0$ vector is easily performed by Eq. (6.41). For predicting an $\boldsymbol{x}_0$ vector (dimension $1 \times p$) from a $\boldsymbol{y}_0$ vector ($1 \times m$), the following expression is used:

$$\boldsymbol{y}_0 = \boldsymbol{x}_0 \boldsymbol{B}$$

For the confidence intervals of the predicted y values, the approaches given for the straight-line model hold (Eqs. (6.25) – (6.27)). Instead of the parameter vector $\boldsymbol{b}$, the matrix of the parameter estimations $\boldsymbol{B}$ is now used. For prediction of a single y value according to Eq. (6.25), we get

$$\boldsymbol{y}_0 = \boldsymbol{x}_0 \boldsymbol{B} \pm \sqrt{F(\alpha; p, n-p) s_{\mathrm{R}}^2 (1 + \boldsymbol{x}_0 (\boldsymbol{X}^{\mathrm{T}} \boldsymbol{X})^{-1} \boldsymbol{x}_0^{\mathrm{T}})} \qquad (6.48)$$

Prediction of an $\boldsymbol{x}_0$ vector (dimension $1 \times p$) from a $\boldsymbol{y}_0$ vector ($1 \times m$) is done by solving a least-squares problem, that is,

$$\boldsymbol{x}_0 = \boldsymbol{y}_0 \boldsymbol{B}^{\mathrm{T}} (\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})^{-1} \qquad (6.49)$$

To estimate the upper bound error for prediction, the following relationship holds:

$$\frac{\|\delta \boldsymbol{x}_0\|}{\|\boldsymbol{x}_0\|} = \mathrm{cond}(\boldsymbol{B}) \left(\frac{\|\delta \boldsymbol{y}\|}{\|\boldsymbol{y}\|} + \frac{\|\delta \boldsymbol{B}\|}{\|\boldsymbol{B}\|} \right) \qquad (6.50)$$

where $\|\delta \boldsymbol{x}_0\| / \|\boldsymbol{x}_0\|$ is the relative error for prediction, $\|\delta \boldsymbol{y}\| / \|\boldsymbol{y}\|$ is the relative error of measurements y, and $\|\delta \boldsymbol{B}\| / \|\boldsymbol{B}\|$ is the relative error of parameter estimation.

From Eq. (6.50), it is obvious that the prediction error will be small if the error in the dependent variable y and the modeling error can be kept low.

Biased Parameter Estimations: PCR and PLS

PCR – Principal Component Regression

PCR is best performed by means of SVD, a method that was introduced in Section 5.2. With this method, the matrix $\boldsymbol{X}$ is decomposed into two orthonormal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ that are joined by a diagonal matrix $\boldsymbol{W}$ of singular values:

Orthonormal means a set of vectors that are orthogonal and have their norm equal to 1.

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^{\mathrm{T}} \tag{6.51}$$

Computation of the regression coefficients $\boldsymbol{b}$ vectorwise is carried out by formation of the pseudo-inverse matrix $\boldsymbol{X}^{+}$ (Moore – Penrose matrix) according to

In OLS, the *pseudo-inverse matrix* is equivalent to the generalized inverse:

$$\boldsymbol{X}^{+} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}$$

$$\boldsymbol{X}^{+} = \boldsymbol{V}(\mathrm{diag}(1/w_{ij})\boldsymbol{U}^{\mathrm{T}}) \tag{6.52}$$

$$\boldsymbol{b} = \boldsymbol{X}^{+}\boldsymbol{y} \tag{6.53}$$

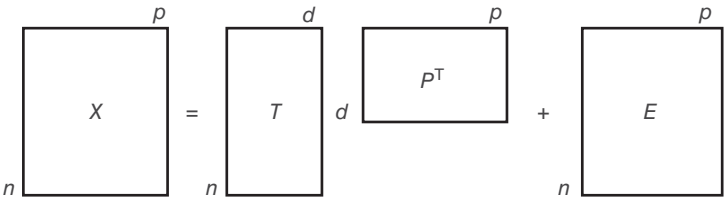
In the case of full rank, all singular values will be obviously different from zero and the SVD solution equals that of OLS. However, one often comes up with several small singular values because of ill-conditioned systems. Therefore, the main goal of PCR is not to keep all singular values for an exact representation of the Moore – Penrose matrix, but to select a subset of singular values that best guarantee predictions of unknown cases.

PLS – Partial Least Squares Regression

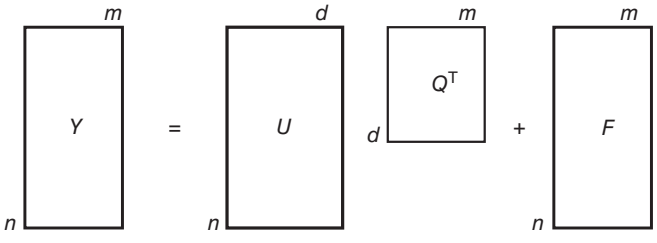
Regression of $\boldsymbol{Y}$ on $\boldsymbol{X}$ by OLS and PCR is based on solving the linear equations columnwise with respect to the $\boldsymbol{Y}$ matrix in order to estimate the regression coefficients in the columns of the matrix $\boldsymbol{B}$ in Eq. (6.41). The decomposition of the $\boldsymbol{X}$ matrix is performed independently of the $\boldsymbol{Y}$ matrix. A method for using the information from the $\boldsymbol{Y}$ matrix is the PLS algorithm as developed by H. Wold and propagated by his son S. Wold. Each PLS latent variable direction of the $\boldsymbol{X}$ matrix is modified so that the covariance between it and the $\boldsymbol{Y}$ matrix vector is maximized. The PLS method is based on a bilinear model with respect to the objects and the variables of the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrix. Both the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrices are decomposed into smaller matrices according to the following scheme:

PCR and PLS are examples of *biased* regression methods where the expected estimated value might differ from the true value of the parameter. Another powerful biased regression method is *ridge regression.*

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{E} \tag{6.54}$$



$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}} + \boldsymbol{F} \tag{6.55}$$



where ***X, Y,*** *n, p, m,* and *d* have the same meanings as given in Eq. (6.41), ***T*** and ***U*** are the $n \times d$ score matrices containing orthogonal rows, ***P*** are the $p \times d$ loadings of the ***X*** matrix, ***E*** is the $n \times p$ error (residual) matrix of the ***X*** matrix, ***Q*** is the $m \times d$ loading of the ***Y*** matrix, and ***F*** is the $n \times m$ error (residual) matrix for the ***Y*** matrix.

To compute the ***B*** coefficients for the general model of Eq. (6.41), the matrices ***P, Q,*** and ***W*** are required:

$$\boldsymbol{B} = \boldsymbol{W}(P^{\mathrm{T}}W)^{-1}Q^{\mathrm{T}} \tag{6.56}$$

with ***W*** being the $p \times d$ matrix of PLS weights. The meaning and estimation of the weight matrix ***W*** can be understood from the PLS algorithm in the following example.

Example 6.7 *PLS Algorithm*

The first dimension *d* (index *l*) is computed from the column mean vectors of the ***X*** and ***Y*** matrix as follows (for centering cf. Eq. (5.2)):

$l = 0$:

$$\boldsymbol{X} = \boldsymbol{X}_{\text{original}} - \overline{\boldsymbol{x}} \tag{6.57}$$

$$\boldsymbol{Y} = \boldsymbol{Y}_{\text{original}} - \overline{\boldsymbol{y}} \tag{6.58}$$

Next, the dimensions $l = 1$ to $l = d$ are computed, based on a suitable stopping criterion, usually the standard error of prediction due to cross-validation (SEP_{CV}, Eq. (6.68)):

Loop for the number of dimensions : $l = l + 1$

The principal components are estimated iteratively, for example, by the nonlinear iterative partial least squares (NIPALS) algorithm. Iteration is halted if the computer precision is reached.

Iteration loop for NIPALS:

1. Use the first column of the actual $\boldsymbol{Y}$ matrix as a starting vector for the y score vector $\boldsymbol{u}$:

$$\boldsymbol{u} = \boldsymbol{y}_1$$

2. Compute the $\boldsymbol{X}$ weights:

$$\boldsymbol{w}^{\mathrm{T}} = \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}} \tag{6.59}$$

3. Scale the weights to a vector of length 1:

$$\boldsymbol{w}^{\mathrm{T}} = \frac{\boldsymbol{w}^{\mathrm{T}}}{(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w})^{1/2}} \tag{6.60}$$

4. Estimate the scores of the $\boldsymbol{X}$ matrix:

$$\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w}^{\mathrm{T}} \tag{6.61}$$

5. Compute the loadings of the $\boldsymbol{Y}$ matrix:

$$\boldsymbol{q}^{\mathrm{T}} = \frac{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{Y}}{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{t}} \tag{6.62}$$

6. Generate the y score vector $\boldsymbol{u}$:

$$\boldsymbol{u} = \frac{\boldsymbol{Y}\boldsymbol{q}}{\boldsymbol{q}^{\mathrm{T}}\boldsymbol{q}} \tag{6.63}$$

Compare $\boldsymbol{u}$(old) with $\boldsymbol{u}$(new). If $||\boldsymbol{u}(\text{old}) - \boldsymbol{u}(\text{new})|| < ||\boldsymbol{u}(\text{new})||$ * THRESHOLD, convergence is obtained. Otherwise, iteration is continued at step 1. The threshold can be chosen on the basis of computer precision.

7. Determine the inner relationship in the form of a scalar b:

$$b = \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{t}}{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{t}} \tag{6.64}$$

8. Compute the loadings of the $\boldsymbol{X}$ matrix:

$$\boldsymbol{p}^{\mathrm{T}} = \frac{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{X}}{\boldsymbol{t}^{\mathrm{T}}\boldsymbol{t}} \tag{6.65}$$

9. Form new residuals for the $\boldsymbol{X}$ and $\boldsymbol{Y}$ matrix:

$$\boldsymbol{E} = \boldsymbol{X} - b\boldsymbol{t}\boldsymbol{p} \tag{6.66}$$

$$\boldsymbol{F} = \boldsymbol{Y} - b\boldsymbol{t}\boldsymbol{q} \tag{6.67}$$

Compute the SEP_{CV}. If the SEP_{CV} is greater than for the actual number of factors, then the optimum number of dimensions has been found. Otherwise, the next dimension is computed. The SEP_{CV} is obtained by

$$SEP_{CV} = \left[\frac{\sum_{j=1}^{m} \sum_{i=1}^{n} \left(y_{ij}^{\text{estimated}} - y_{ij}^{\text{true}} \right)^2}{n \cdot m} \right] \tag{6.68}$$

The B coefficients are finally computed according to Eq. (6.56), that is,

$$\boldsymbol{B} = \boldsymbol{W}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W})^{-1}\boldsymbol{Q}^{\mathrm{T}}$$

One can show that the matrix $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}$ is an upper *bidiagonal* matrix so that the PLS algorithm represents just a variation of diagonalizing a matrix before its inversion.

To estimate the error of the model, there are two possibilities. Either the objects or samples are predicted by *resubstitution* into the model equation (Eq. (6.41)), giving an estimate of the standard error of modeling, or the model is built by leaving out objects randomly and predicting the left-out samples from the reduced model. The latter prediction is known as *cross-validation* and reveals the standard error of prediction from cross-validation, SEP_{CV} (cf. Eq. (6.68)).

The PLS algorithm is one of the standard methods used for *two-block* modeling, for example, for multivariate calibration as given as follows.

OPLS – Orthogonal Partial Least Squares

Orthogonal partial least squares (OPLS) or orthogonal projection to latent structures (O-PLS) has been developed in order to separate information in the $\boldsymbol{X}$ matrix that is correlated with $\boldsymbol{Y}$ from $\boldsymbol{Y}$-uncorrelated (orthogonal) information [6]. The PLS model from Eq. (6.54) is extended by an extra component of scores, $\boldsymbol{T}_{\mathrm{o}}$, and loadings, $\boldsymbol{P}_{\mathrm{o}}$, which account for the orthogonal variations:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{T}_{\mathrm{o}}\boldsymbol{P}_{\mathrm{o}}{}^{\mathrm{T}} + \boldsymbol{E} \tag{6.69}$$

This method provides less complex and easier way to interpret models. It can also be considered as a preprocessing step to remove all $\boldsymbol{Y}$-orthogonal variation in $\boldsymbol{X}$ by calculating $\boldsymbol{X} - \boldsymbol{T}_{\mathrm{o}}\boldsymbol{P}_{\mathrm{o}}{}^{\mathrm{T}}$ followed by PLS analysis in the usual way.

Generalization of OPLS to a bidirectional version is known as *O2PLS* and modeled in the following way:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{W}^{\mathrm{T}} + \boldsymbol{T}_{\mathrm{o}}\boldsymbol{P}_{\mathrm{Yo}}{}^{\mathrm{T}} + \boldsymbol{E} \tag{6.70}$$

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{U}_{\mathrm{o}}\boldsymbol{P}_{\mathrm{Xo}}{}^{\mathrm{T}} + \boldsymbol{F} \tag{6.71}$$

here, $\boldsymbol{T}$ and $\boldsymbol{U}$ denote the predictive scores and $\boldsymbol{T}_{\mathrm{o}}$ and $\boldsymbol{U}_{\mathrm{o}}$ the orthogonal scores, $\boldsymbol{W}$ and $\boldsymbol{C}$ are the predictive weight matrices, and $\boldsymbol{P}_{\mathrm{Yo}}$ and $\boldsymbol{P}_{\mathrm{Xo}}$ the orthogonal weight matrices as well as $\boldsymbol{E}$ and $\boldsymbol{F}$ the residual matrices of the $\boldsymbol{X}$- and $\boldsymbol{Y}$-matrices, respectively.

Example 6.8 *Orthogonal Partial Least Squares (OPLS)*

Here, we consider NIR spectra that have been recorded in order to analyze the protein content in wheat (cf. Example 6.10). The original 30 spectra shown in Figure 6.6a are of less regular structure in comparison to the spectra of Figure 6.6b corrected by the OPLS model of Eq. (6.69).

Multivariate calibration of the spectra on protein contents by OPLS will reveal better predictions with fewer principal components compared to the original PLS algorithm used in Example 6.10.

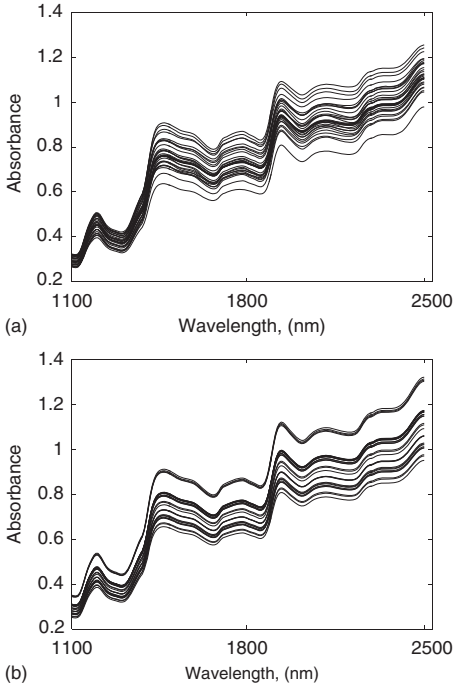


Figure 6.6 Original NIR spectra (a) and OPLS corrected spectra (b).

Applications for Multicomponent Analysis

As an example for multivariate modeling, we consider the simultaneous determination of several components in low-selective analytical systems (multicomponent analysis). These components can be elements, compounds, or chemical/physical properties. By means of multicomponent analysis, constituents of pharmaceutical formulations can be determined in the UV range, the water and protein content of cereal can be estimated from NIR spectra or chemical elements, and technological parameters of coal are predictable on the basis of infrared (IR) spectra. The limited selectivity of chemical sensors can also be overcome by applying the principles of multicomponent analysis.

The principles are introduced on the basis of spectrometric multicomponent analysis, since this method currently dominates in applications.

Spectrometric multicomponent analysis is based on Beer's law that is formulated for a single component:

$$A_\lambda = \varepsilon_\lambda d c \tag{6.72}$$

where A_λ is the absorbance at wavelength, λ; ε_λ is the molar absorption coefficient at wavelength, λ, in liters per mole per centimeter; d is the cell thickness in centimeters; and c is the molar concentration in moles per liter.

The absorbances can be normalized to a constant cell thickness, so that a simplified Beer's law results:

$$A_\lambda = k_\lambda c \tag{6.73}$$

where k_λ is the normalized absorption coefficient.

In a multicomponent system, it is assumed that the absorbances at a specific wavelength i can be represented by the sum of absorbances of the m individual components according to

$$A_i = k_{i1}c_1 + k_{i2}c_2 + \cdots + k_{im}c_m = \sum_{j=1}^{m} k_{ij}c_j \tag{6.74}$$

where A_i is the absorbance at wavelength, i; k_{ij} is the normalized absorption coefficient of the jth component at wavelength, i; and c_j is the concentration of the jth component.

In multiwavelength spectroscopy, the spectra are acquired at p wavelengths, so that either exactly determined linear equation

systems ($p = m$) or overdetermined systems ($p > m$) emerge, that is,

$$\begin{aligned} A_1 &= k_{11}c_1 + k_{12}c_2 + \cdots + k_{1m}c_m \\ A_2 &= k_{21}c_1 + k_{22}c_2 + \cdots + k_{2m}c_m \\ &\vdots \\ A_p &= k_{p1}c_1 + k_{p2}c_2 + \cdots + k_{pm}c_m \end{aligned}$$

In matrix notation, we obtain

$$\boldsymbol{a} = \boldsymbol{K}\boldsymbol{c} \tag{6.75}$$

where $\boldsymbol{a}$ is a $p \times 1$ dimensional vector that represents the spectrum; $\boldsymbol{K}$ a $p \times m$ matrix of normalized molar absorptivities; and $\boldsymbol{c}$ an $m \times 1$ dimensional vector of concentrations.

Direct Calibration Method

This method is used if all the absorptivities are known. This implies that the pure component spectra can be measured or can be obtained elsewhere, that there is no interaction between the different components in the sample or between constituents and the solvent, and that unknown matrix constituents do not interfere with the determination.

Analysis of unknown samples is based on the sample spectrum, $\boldsymbol{a}_0$, and the known absorptivities $\boldsymbol{K}$ according to (cf. Eq. (6.49))

$$\boldsymbol{c}_0^{\mathrm{T}} = \boldsymbol{a}_0^{\mathrm{T}}\boldsymbol{K}^{-1} \quad \text{for } m = p \tag{6.76}$$

or

$$\boldsymbol{c}_0 = \boldsymbol{K}^{\mathrm{T}}(\boldsymbol{K}\boldsymbol{K}^{\mathrm{T}})^{-1}\boldsymbol{a}_0 \quad \text{for } m > p \tag{6.77}$$

where $\mathbf{c}_0$ is the vector of predicted concentrations.

Example 6.9 *Direct Calibration*

Two constituents in a sample are to be determined from their absorbances at either two or three wavelengths. The following $\boldsymbol{K}$ matrix for the absorptivities (arbitrary units) is given:

Wavelength	Constituent	
	1	2
1	3.00	2.00
2	3.00	4.00
3	2.00	6.00

The absorbance data measured on the sample are

$$\boldsymbol{a}_0 = \begin{pmatrix} 0.71 \\ 1.09 \\ 1.41 \end{pmatrix}$$

The true concentrations of that sample are

$$\boldsymbol{c}_0 = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 0.100 \\ 0.200 \end{pmatrix}$$

Determination at the *two wavelengths* 1 and 2 (exactly determined linear system, $p = m$) reveals

$$\begin{aligned} \boldsymbol{c}_0^{\mathrm{T}} &= \boldsymbol{a}_0^{\mathrm{T}} \boldsymbol{K}^{-1} = (0.71, 1.09) \begin{bmatrix} 3.0 & 3.0 \\ 2.0 & 4.0 \end{bmatrix}^{-1} \\ &= (0.71, 1.09) \begin{bmatrix} 0.666 & -0.5 \\ -0.333 & 0.5 \end{bmatrix} = (0.110,\ 0.190) \end{aligned}$$

or expressed for the single concentrations:

$$c_1 = 0.110 \,\text{and}\, c_2 = 0.190$$

As a result, the relative deviation between estimated and true concentrations is obtained as

$$\delta c_1 = \frac{c_1^{\text{predicted}} - c_1^{\text{true}}}{c_1^{\text{true}}} \cdot 100 = \frac{0.110 - 0.100}{0.100} \cdot 100 = 10\%$$

$$\delta c_2 = \frac{c_2^{\text{predicted}} - c_2^{\text{true}}}{c_2^{\text{true}}} \cdot 100 = \frac{0.190 - 0.200}{0.200} \cdot 100 = 5\%$$

This relatively high error can be decreased if all *three wavelengths* are used in the analysis:

$$\begin{aligned} \boldsymbol{c}_0 = \boldsymbol{K}^{\mathrm{T}} (\boldsymbol{K}\boldsymbol{K}^{\mathrm{T}})^{-1} \boldsymbol{a}_0 &= \begin{bmatrix} 3 & 3 & 2 \\ 2 & 4 & 6 \end{bmatrix} \left(\begin{bmatrix} 3 & 2 \\ 3 & 4 \\ 2 & 6 \end{bmatrix} \begin{bmatrix} 3 & 3 & 2 \\ 2 & 4 & 6 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0.71 \\ 1.09 \\ 1.41 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 3 & 2 \\ 2 & 4 & 6 \end{bmatrix} \left(\begin{bmatrix} 13 & 17 & 18 \\ 17 & 25 & 30 \\ 18 & 30 & 40 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0.71 \\ 1.09 \\ 1.41 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 3 & 2 \\ 2 & 4 & 6 \end{bmatrix} \left(\begin{bmatrix} 0.125 & 0.0479 & -0.0967 \\ 0.0479 & 0.0209 & -0.0309 \\ -0.0967 & -0.0309 & 0.0889 \end{bmatrix} \right) \begin{bmatrix} 0.71 \\ 1.09 \\ 1.41 \end{bmatrix} \\ &= \begin{pmatrix} 0.0995 \\ 0.2002 \end{pmatrix} \end{aligned}$$

For the relative deviations from the true concentrations, we now obtain

$$\delta c_1 = \frac{c_1^{\text{predicted}} - c_1^{\text{true}}}{c_1^{\text{true}}} \cdot 100 = \frac{0.0995 - 0.100}{0.100} \cdot 100 = 0.5\%$$

$$\delta c_2 = \frac{c_2^{\text{predicted}} - c_2^{\text{true}}}{c_2^{\text{true}}} \cdot 100 = \frac{0.2002 - 0.200}{0.200} \cdot 100 = 0.1\%$$

Notice that the error has been decreased by adding just one wavelength to the analysis scheme. With higher floating point precision, the result could be even further improved. Therefore, *multiwavelength spectrometry* can be a powerful alternative to systems where, for every component, only a single wavelength is applied.

In practice, some or even all of the aforementioned prerequisites for direct calibration on the basis of OLS regression are often not obeyed. Therefore, more sophisticated calibration procedures have to be carried out based on mixture or multivariate calibration.

Indirect Calibration Methods

Indirect methods are based on estimating the calibration parameters from calibration mixtures. These methods offer the following advantages:

- Interactions between constituents or between constituents and the sample matrix can be accounted for in the calibration. Thus, the validity of Beer's law, that is, the additivity of spectra for every single component and linear response–concentration relationships, is not a prerequisite any more.
- Modeling of background in a principal component becomes feasible.
- Systems of highly correlated spectra can also be used for multicomponent analysis.

The different methods for multivariate calibration differ by the mathematical model that is based either on Beer's law, that is, the spectra are regressed on concentrations as with the K-matrix approach or on inverse models where the regression of concentrations on spectra is carried out.

Collinearity of data refers to approximate linear dependence among variables.

K-Matrix Approach The K-matrix approach is based on an extension of Eq. (6.72) to matrix form

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & ap \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{np} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & & & \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1p} \\ k_{21} & & & k_{2p} \\ \vdots & & & \\ k_{m1} & k_{m2} & \dots & k_{mp} \end{bmatrix}$$

or in matrix notation

$$\boldsymbol{A} = \boldsymbol{CK} \tag{6.78}$$

where $\boldsymbol{A}$ is the $n \times p$ matrix of absorbances, $\boldsymbol{C}$ is the $n \times m$ matrix of concentrations of constituents, $\boldsymbol{K}$ is the $m \times p$ matrix of absorptivities, n is the number of samples, p is the number of wavelengths, and m is the number of components.

In the present notation, it is assumed that the absorbance data are centered and that, therefore, there is no intercept at the absorbance axis. If uncentered data are used, the first column in the concentration matrix should consist of 1's, and in the K matrix, the intercept coefficients would have to be introduced as the first row.

Calibration is based on a set of n samples of known concentrations for which the spectra are measured. By means of the calibration sample set, estimation of absorptivities is possible by solving for the matrix $\boldsymbol{K}$ according to the general least squares solution:

$$\boldsymbol{K} = (\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{A} \tag{6.79}$$

The analysis is then based on the spectrum a_0 $(1 \times p)$ of the unknown sample by

$$\boldsymbol{c}_0 = \boldsymbol{a}_0\boldsymbol{K}^{\mathrm{T}}(\boldsymbol{K}\boldsymbol{K}^{\mathrm{T}})^{-1} \tag{6.80}$$

where $\boldsymbol{c}_0$ is the $(1 \times m)$ vector of sought-for concentrations.

A great advantage of the K-matrix approach is the fact that the elements of the $\boldsymbol{K}$ matrix represent genuine absorptivities with reference to the spectra of the individual constituents. Also, the general assumption in least squares regression analysis is valid, such that only the dependent variable, here the absorbance, is error prone.

In the K-matrix approach, all absorbing constituents of a sample must be explicitly known to be included into the calibration procedure. As we will see in the following, with more soft modeling techniques, it will also be possible to account for unknown constituents without their explicit calibration.

Another disadvantage of the K-matrix approach results from the fact that calibration *and* analysis are connected to the inversion of a matrix. Although this is not a problem from the point

of view of computational time, it might become a problem if ill-conditioned (less selective) systems are applied, where the spectra of the constituents are very similar. Then in the analysis step (Eq. (6.80)), a badly conditioned matrix of absorptivities has to be inverted that might be almost singular, that is, all singular values or eigenvalues are zero. To overcome this difficulty, powerful algorithms for solving linear equations, such as SVD, should be used in connection with reduction of the dimensionality of the problem.

An alternative to the K-matrix approach is to calibrate the concentrations directly on the spectra. These methods are called *inverse calibration methods*.

P-Matrix Approach The P-matrix approach is based on the following model:

$$\begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & & & \\ c_{n1} & c_{n2} & \cdots & c_{nm} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & & p_{2m} \\ \vdots & & & \\ p_{p1} & p_{p2} & \cdots & p_{pm} \end{bmatrix}$$

Calibration of concentrations on signals instead of signals on concentrations is termed an *inverse calibration*.

and in matrix notation:

$$\boldsymbol{C} = \boldsymbol{AP} \tag{6.81}$$

The calibration coefficients are now the elements of the $\boldsymbol{P}$ matrix that are estimated by the generalized least squares solution (OLS) according to

$$\boldsymbol{P} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{C} \tag{6.82}$$

Analysis is carried out by direct multiplication of the measured sample spectrum $\boldsymbol{a}_0$ by the $\boldsymbol{P}$ matrix:

$$\boldsymbol{c}_0 = \boldsymbol{a}_0\boldsymbol{P} \tag{6.83}$$

A disadvantage of this calibration method is the fact that the calibration coefficients (elements of the $\boldsymbol{P}$ matrix) have no physical meaning, since they do not reflect the spectra of the individual components. The usual assumptions about errorless independent variables (here, the absorbances) and error-prone dependent variables (here, concentrations) are not valid. Therefore, if this method of inverse calibration is used in connection with OLS for estimating the $\boldsymbol{P}$ coefficients, there is only a slight advantage over the classical K-matrix approach, due to the fact that a second matrix inversion is avoided. However, in connection with more soft modeling methods, such as PCR or PLS, the inverse calibration approach is one of the most frequently used calibration tools.

Soft Modeling

The methods of soft modeling are based on the inverse calibration model where concentrations are regressed on spectral data:

$$\boldsymbol{C} = \boldsymbol{AB} \tag{6.84}$$

where $\boldsymbol{C}$, $\boldsymbol{A}$ are again the $n \times m$ concentration and $n \times p$ absorbance matrix, respectively, and $\boldsymbol{B}$ is the $p \times m$ matrix of regression or B coefficients.

PCR Approach The method of PCR was outlined earlier on the basis of SVD. For simultaneous spectroscopic multicomponent analysis, the decomposition of the absorbance matrix $\boldsymbol{A}$ can be written as (cf. Eq. (5.23)):

$$\boldsymbol{A} = \boldsymbol{UWV}^{\mathrm{T}} \tag{6.85}$$

Estimation of the matrix of regression coefficients $\boldsymbol{B}$ is performed columnwise by

$$\boldsymbol{b} = \boldsymbol{A}^{+}\boldsymbol{c} \tag{6.86}$$

with $\boldsymbol{A}^{+}$ being the pseudo-inverse of the absorbance matrix $\boldsymbol{A}$ (cf. Eq. (6.53)).

The main advantages of PCR calibration are as follows:

- Decomposition of the absorbance matrix into smaller orthogonal matrices enables reduction of the dimensionality of the problem in the case of ill-conditioned systems. So, if highly correlated spectra are to be investigated, one will always obtain the best solution, even in the case of nearly singular matrices.
- Additional unknown components or background components can be automatically modeled as principal components if the concentrations of those components vary within the different calibration samples.

Problems may occur if small principal components are eliminated in the process of reducing the number of significant principal components/singular values, because it might happen that one of the eliminated singular values is important for the prediction of a certain constituent concentration. The decomposition of the absorbance matrix $\boldsymbol{A}$ does not consider relationships between the concentrations and the absorbances. Therefore, the decomposition might be not optimal with respect to further use of the calibration model for prediction of concentrations in unknown samples.

A method that accounts for the concentration–spectra relationships during decomposition is the PLS approach.

PLS Approach Details of the PLS method were given earlier in this chapter. In multicomponent analysis, we obtain the following equations for the decomposition of the absorbance matrix $\boldsymbol{A}$ (the former $\boldsymbol{X}$ matrix) and the concentration matrix $\boldsymbol{C}$ (formerly the $\boldsymbol{Y}$ matrix) according to the inverse calibration model in Eq. (6.87):

$$\boldsymbol{C} = \boldsymbol{AB} \tag{6.87}$$

$$\boldsymbol{A} = \boldsymbol{TP}^{\mathrm{T}} + \boldsymbol{E} \tag{6.88}$$

$$\boldsymbol{C} = \boldsymbol{TQ}^{\mathrm{T}} + \boldsymbol{F} \tag{6.89}$$

$$\boldsymbol{B} = \boldsymbol{W}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W})^{-1}\boldsymbol{Q}^{\mathrm{T}} \tag{6.90}$$

The meaning of the additional matrices is the same as in Eqs. (6.54)–(6.56).

The main advantage of the PLS method is based on the interrelated decomposition of the concentration matrix $\boldsymbol{C}$ and the absorbance matrix $\boldsymbol{A}$, so that it is with this algorithm that the most robust calibrations can presently be obtained.

Regression Diagnostics

Leading vendors of software for multicomponent analysis now provide a great variety of tools for diagnosing the suitability of the calibration model, for detecting outliers and influential samples or for estimating realistic prediction errors. It is not unusual that for a calibration set consisting of 30 standard samples, about 5000 different diagnostic plots could be generated.

Visual inspection should be possible from plots of predicted versus measured concentrations, from principal component plots of loadings and scores in the case of soft modeling techniques, and by plotting the standard error of calibration (SEC) or the standard error of prediction ($\mathrm{SEP}_{\mathrm{CV}}$, Eq. (6.68)) from cross-validation in dependence on the number of eigenvalues or of principal components.

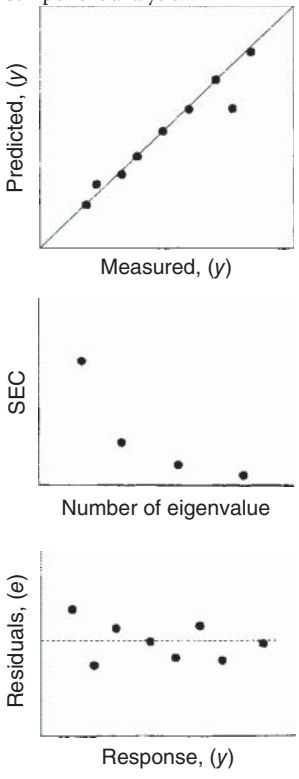
The study of the residuals is very important in diagnostic statistics. Let us return to the general least squares model given in Eq. (6.41). We rewrite it here for a single y variable as follows:

$$\boldsymbol{y} = \boldsymbol{Xb} + \boldsymbol{e} \tag{6.91}$$

where $\boldsymbol{e}$ is the vector of residuals of y values, that is, the difference between the measured y value, $\boldsymbol{y}$, and the y value estimated by the model, $\hat{\boldsymbol{y}}$; for a single y value j $e_j = y_j - \hat{y}_j$.

In the case of an inverse calibration model, one can interpret the model in Eq. (6.88) as regressing the concentrations of a single

Diagnostic plots in multicomponent analysis:



component, $\boldsymbol{y}$, on the spectra of the calibration samples collected in the matrix $\boldsymbol{X}$ with n rows (samples) and p columns (wavelength) according to Eq. (6.78). The regression coefficients $\boldsymbol{b}$ are then in a $p \times 1$ vector.

The relationship between the estimated and measured y values can be described by a fundamental matrix, the *hat matrix*, $\boldsymbol{H}$. As explained in "Ordinary Least Squares Regression" Section, the regression parameters are estimated by the general inverse as

$$\boldsymbol{b} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{6.92}$$

The fitted model has the form

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{b}} \tag{6.93}$$

Substitution of Eq. (6.89) into Eq. (6.90) reveals

$$\hat{\boldsymbol{y}} = \boldsymbol{X}[(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}] = \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{H}\boldsymbol{y} \tag{6.94}$$

The projection matrix $\boldsymbol{H}$ is called the *hat matrix* because it puts the "hat" on $\boldsymbol{y}$.

where $\boldsymbol{H}$ is the $n \times n$ hat matrix defined by

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}} \tag{6.95}$$

The hat matrix transforms the vector of measured y values into the vector of fitted $\hat{\boldsymbol{y}}$ values. The element of $\boldsymbol{H}$, denoted by h_{ij}, is computed by

$$h_{ij} = \boldsymbol{x}_i^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{x}_j \tag{6.96}$$

Many special relationships can be found with the h_{ij}. For example, the *rank* of the matrix $\boldsymbol{X}$ is easily found from the diagonal elements of the hat matrix by the formula

$$\mathrm{rank}(\boldsymbol{X}) = \sum_{i=1}^{n} h_{ii} \tag{6.97}$$

Further relationships will be learnt subsequently.

Residuals and Prediction Error

In a principal component model, the leverage value is replaced by the *squared* Mahalanobis distance. For the component score matrix $\boldsymbol{T}$, this distance is computed by $\boldsymbol{T}(\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T})^{-1}\boldsymbol{T}^{\mathrm{T}}$.

The relationship of the hat matrix with the residuals can be understood from the following equations:

$$\hat{\boldsymbol{e}} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = [\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}] = [\boldsymbol{I} - \boldsymbol{H}]\boldsymbol{y} \tag{6.98}$$

The residuals depend directly on the product formed by the vector of the measured y values and the difference between the identity and the hat matrix.

The elements of the hat matrix are also important for estimating the standard error of prediction. In general, the prediction error

is calculated as the *predictive residual sums of squares* (PRESS) by

$$\text{PRESS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{6.99}$$

The influence of observations on a regression model can be assessed by its *leverage.* If the prediction error is to be estimated on the basis of the calibration sample set, k samples are left out to calculate new prediction residual errors, $\hat{e}_{(k)}$. By using the elements of the hat matrix, sometimes also called *leverages* of case k, a good approximation of the estimated prediction error can be obtained according to

$$e_{\text{cv}} = \frac{\hat{e}_{(k)}}{1 - h_{kk}} \tag{6.100}$$

For the new *PRESS*, we obtain

$$\text{PRESS}_{\text{CV}} = \sum e_{\text{cv}}^2 \tag{6.101}$$

Based on Eq. (6.98), it is possible to estimate a realistic prediction error without analyzing additional standard samples.

Outliers

Low prediction errors go along with good models. Samples that do not follow the same model as the rest of the data are called *outliers*. Testing for outliers can also be based on the leverage values, h_{ij}. Suppose that the kth sample is an outlier, then a new model is calculated after deleting the kth sample from the data set. Based on the new estimation for the regression parameters $\hat{\boldsymbol{b}}_{(k)}$, new residual values, $\hat{e}_{(k)}$, are obtained where the kth sample was not used. To test for the significance of the outlier, a Student's t-test can be applied. If $\hat{y}_k$ is not an outlier, the null hypothesis can be assumed, such that there is no difference in predicting $\hat{y}_k$ with the full model or with the model estimated without the potential outlier k. If $\hat{y}_k$ is an outlier, then the t value should exceed the critical value at a certain risk level. The *t value* can be approximated by the leverage value by formula

The externally studentized residual is also termed *jackknifed residual*.

$$t_k = \frac{\hat{e}_k}{s_{(k)}\sqrt{1 - h_{kk}}} \tag{6.102}$$

where t_k is called the *externally studentized residual* since case k is not used in computing $s_{(k)}$; it has $n - p - 1$ degrees of freedom; $s_{(k)}$ is derived from the residual mean square, that is,

$$s_{(k)}^2 = \frac{\sum_{i=1, i \neq k}^{n} e_i^2}{n - p - 1}$$

The externally studentized residual is scale invariant. Commonly, an outlier is detected if $t_k > 2$.

Influential Observations

Outliers should not be confused with influential observations. Until now, we have used the residuals in order to find problems with a model. If we want to study the robustness of a model to perturbations, we do an influence analysis. This kind of study is done as though the model were correct. Influential observations cannot be detected by large residuals. Their removal, however, may cause major changes in subsequent use of the model. The difference can be understood from Figure 6.7. A straight-line model that includes the influential observation will give a different slope if that observation is deleted. On the other hand, if the obvious outlier is included in the model, we will estimate larger residuals for all of the cases.

To measure the change of the influential observation, the model has to be built by including or deleting it. From the two models, we obtain different estimations for the y values that can be used to compute a measure, the so-called *Cook's distance*, D_k:

$$D_k = \frac{(\hat{\boldsymbol{y}}_{(k)} - \hat{\boldsymbol{y}})^{\mathrm{T}}(\hat{\boldsymbol{y}}_{(k)} - \hat{\boldsymbol{y}})}{ps^2} \tag{6.103}$$

where $\hat{\boldsymbol{y}}_{(k)}$ is the vector of y values estimated from the model without case k, $\hat{\boldsymbol{y}}$ is the vector of y values estimated with the full model, and s^2 is the variance computed similar as in Eq. (6.103) for the full model, that is,

$$s^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p}$$

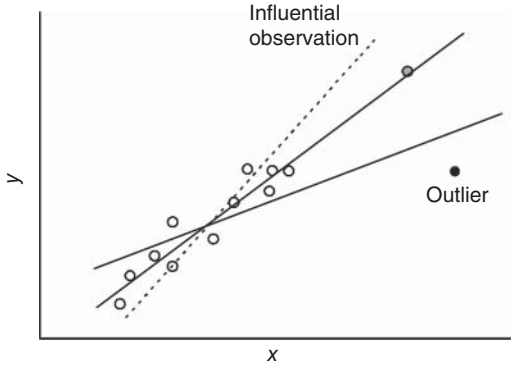


Figure 6.7 Straight modeling in the presence of an outlier and an influential observation.

Again the leverage value can be used to estimate the D value as follows:

$$D_k = \frac{1}{p} r_k^2 \left(\frac{h_{kk}}{1 - h_{kk}} \right) \quad \text{with } r_k = \frac{\hat{e}_k}{s\sqrt{1 - h_{kk}}} \tag{6.104}$$

r_k is called the *internally studentized residual* because s is estimated by including all of the data.

Large D_k values reflect the substantial influence of case k. Therefore, samples or cases with the largest D_k values will be of interest. In practice, those samples should be deleted and the model should be recomputed in order to understand which changes will happen.

An additional possibility for standardization of residuals is the standardized residual computed by

$$r_i = \hat{e}_i (s\sqrt{1 - h_{ii}})$$

Example 6.10 *NIR – Multivariate Analysis of Protein in Wheat*

As an example of spectroscopic multicomponent analysis, the protein content in whole wheat is calibrated on the basis of NIR spectra. The model is then used for analysis of an unknown sample.

In total, $n = 30$ calibration samples are available. The NIR spectrum of one sample is given in Figure 6.8. We will use the inverse calibration method (cf. Eq. (6.81)), that is, according to the general equation of multiple regression (Eq. (6.41)), the protein content is arranged in the y vector and the matrix $\boldsymbol{X}$ contains the NIR spectra. In order to keep the calibration model small, only the five most

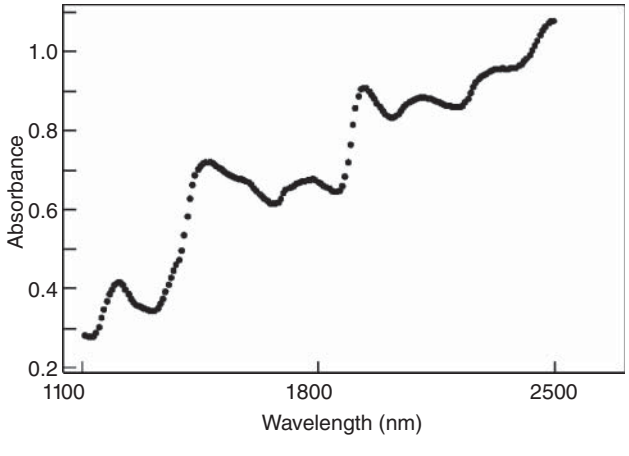


Figure 6.8 NIR spectrum of wheat.

important wavelengths are evaluated. The regression model on the basis of OLS reads as follows:

$$y = 6.23 + 670.9x_1 - 4154x_2 + 3682x_3 - 176.8x_4 - 9.371x_5 \quad (6.105)$$

Table 6.4 provides the results for the ANOVA. The sum of squares corrected for the mean is explained by 95.41% (coefficient of determination Eq. (6.17)) due to the factors, here the wavelength. The F test for goodness-of-fit is also given in Table 6.4. Based on a significance level of $\alpha = 0.05$, the goodness-of-fit test is significant, since the p level is smaller than 0.05. This means that the parameters of the linear calibration model are significantly different from zero.

Figure 6.9 illustrates the recovery function for the analysis of protein contents in wheat. The predictions correspond here to

Table 6.4 ANOVA table for OLS calibration of NIR spectra in the analysis of protein in wheat.

Source of variation	SS	*df*	MSS	*F* value	*p*-level
Corrected for the mean, SS_{corr}	69.80	29	2.41	—	—
Factors, SS_{fact}	66.60	5	13.32	99.81	0.000
Residuals, SS_R	3.203	24	0.1334	—	—

SS – sum of squares, MSS – mean sum of squares, and *df* – degrees of freedom.

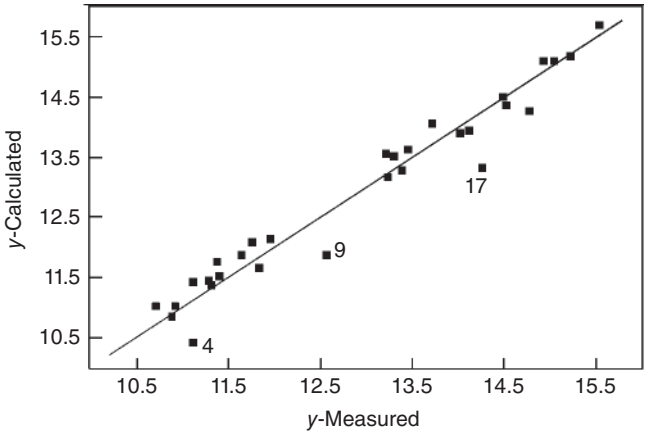


Figure 6.9 Recovery function for resubstitution of samples in case of inverse OLS calibration of protein (mass%) by means of NIR spectra.

resubstitution of the calibration samples into the model in Eq. (6.105). For the mean *SEC*, one obtains from the *PRESS* value (Eq. (6.99))

$$\text{PRESS} = \sum_{i=1}^{n} \hat{e}_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = 3.203$$

$$\text{SEC} = \sqrt{\frac{\text{PRESS}}{n}} = \sqrt{\frac{3.203}{30}} = 0.327 \qquad (6.106)$$

To estimate the error of unknown (independent) samples, the standard error of prediction of cross-validation is to be applied (SEP_{CV}, Eq. (6.68)). It is expected that this error is greater than that for resubstitution, and it amounts to

$$\text{SEP}_{\text{CV}} = \sqrt{\frac{\text{PRESS}_{\text{CV}}}{n}} = \sqrt{\frac{4.592}{30}} = 0.391$$

Related to the mean protein content of 13 mass%, this corresponds to a relative error of (0.391/13)100% = 3.0%.

Regression diagnostics are carried out by using Cook's distance (cf. Figure 6.10 and Eq. (6.104)) and the analysis of residuals. The latter are given as *common residuals* according to Eq. (6.28) and as *jackknifed residuals* after Eq. (6.102) in dependence on the diagonal elements of the hat matrix. The plot of Cook's distances (Figure 6.10a) reveals sample 4 as a potential influential observation. However, this sample also has a high residual error, so that the model has to be recalculated after elimination of the suspicious sample.

A typical outlier is sample number 17. The sample has a very large residual, but cannot be identified as an influential observation.

For computation of the content of the unknown sample, the absorbances are inserted into the calibration model (Eq. (6.105)). The following protein content in percentage mass results:

$$y = 6.23 + 670.9 \cdot 0.4569 - 4154 \cdot 0.4178 + 3682 \cdot 0.4134 - 176.8 \cdot 0.4348 - 9.371 \cdot 0.9816 = 13.29$$

Finally, calibration is performed on the basis of the complete spectrum using 176 wavelengths. The OLS method cannot be used for this, since only 30 samples are available, and therefore, $n < p$, that is, the matrix is rank deficient.

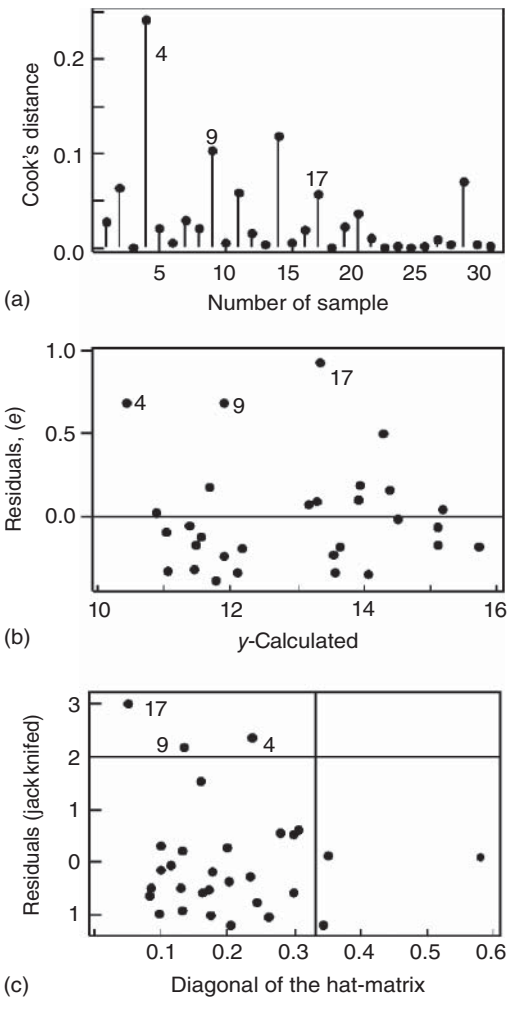


Figure 6.10 Regression diagnostics for influential observations and outliers. (a) Cook's distance for recognition of influential observations. (b) Residual plot in dependence on the calculated *y* values. (c) Jackknifed residuals according to Eq. (6.102).

The results based on biased parameter estimates, PCR and PLS, are represented by their *PRESS* values in Figure 6.11.

As expected, the fit of calibration spectra is improved with increasing component number. This effect is here still more pronounced for the PLS method than for the PCR method. Prediction of unknown samples, however, should not be based on the maximum number of principal components. For

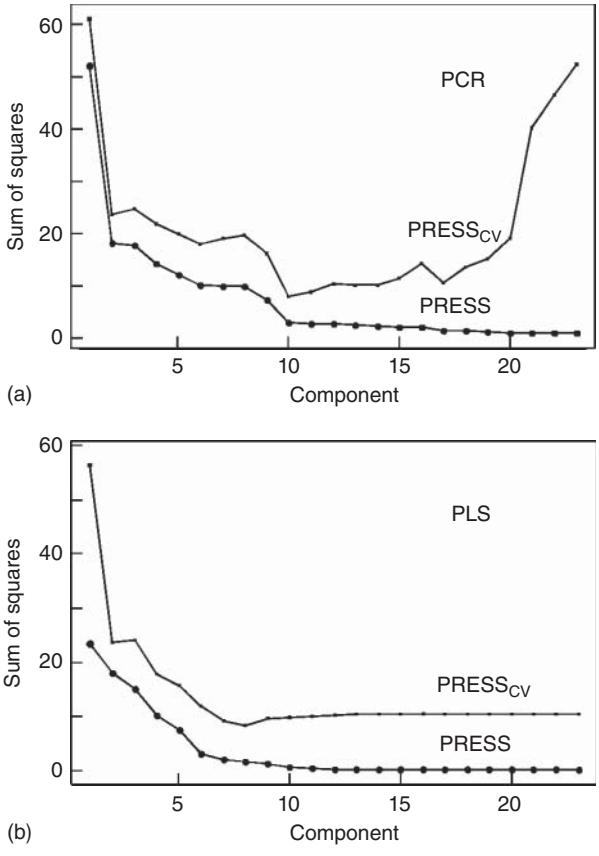


Figure 6.11 Error for calibration of 30 wheat samples by means of 176 wavelengths if principal component regression (*PCR*) (a) and partial least squares (*PLS*) regression (b) are used for data evaluation. The predictive residual sums of squares (*PRESS*) values are computed by Eq. (6.99). *PRESS* corresponds to the error due to resubstitution (Eq. (6.99)) and $PRESS_{CV}$ to the error estimated by cross-validation (Eq. (6.101)).

estimation of the optimal component number with respect to robust prediction, the error due to cross-validation, $PRESS_{CV}$, according to Eq. (6.101), is computed on the basis of the *leverage* value. For both calibration methods, we observe a minimum in the dependence on the number of components. It lies at 10 components for PCR and at 8 for PLS.

If the prediction error based on the standard error is compared to the computation with only five wavelengths, for both methods, PCR ($SEP_{CV} = 0.519$) and PLS ($SEP_{CV} = 0.526$), worse results are obtained. This is explained by the fact that

the five optimal wavelengths were selected. The judicious choice of features, here the wavelength, is therefore an important aspect in applying the methods of multicomponent analysis.

Multiway Regression (Modeling)

Up to this point, regression has been restricted to two blocks of two-way data $\boldsymbol{Y}$ and $\boldsymbol{X}$. In chemical analysis, however, a growing number of problems can be cast as three-way regression analysis. Consider the calibration of chemical constituents on the basis of their fluorescence excitation/emission spectrum or of gas chromatography/mass spectrometry (GC/MS) data. For each sample, two-dimensional measurements are available that constitute a three-way data array, $\underline{\boldsymbol{X}}$. This data array has to be related to sample concentrations of one, vector $\boldsymbol{y}$, or several analytes, matrix $\boldsymbol{Y}$. Cases can be imagined where even the matrix $\boldsymbol{Y}$ constitutes a three-way data array.

Methods for *simultaneous* N-way regression can be based on the decomposition of the $\underline{\boldsymbol{X}}$ array by multiway methods introduced in Section 5.2 (parallel factor analysis (PARAFAC) or Tucker models) and regressing the dependent variable on the components of those models. A drawback with this approach is that the separately estimated components are not necessarily predictive for $\boldsymbol{Y}$. This caused the development of improved algorithms for multiway regression analysis of that kind.

A frequently used *sequential* N-way method that predicts $\boldsymbol{y}$ and decomposes $\underline{\boldsymbol{X}}$ in a PARAFAC-like mode is multilinear or N-way partial least squares (N-PLS). Denoting the component vector $\boldsymbol{t}$ and the weighing vectors for mode K, $\boldsymbol{w}^K$, and for mode J, $\boldsymbol{w}^J$, the following criterion has to be maximized (cf. Figure 6.12):

$$\max_{w^J \boldsymbol{w}^K} \left[cov\left(\boldsymbol{t}, y\right) \middle| \min \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(x_{ijk} - t_i w_j^J w_k^K \right)^2 \right) \right] \quad (6.107)$$

Introducing a summation vector $\boldsymbol{z}$ computed from $\boldsymbol{y}$ and $\boldsymbol{x}$ implies

$$\begin{aligned} &\max_{w^J \boldsymbol{w}^K} \left[\sum_{i=1}^{I} t_i y_i \middle| t_i = \sum_{j=1}^{J} \sum_{k=1}^{K} x_{ijk} w_j^J w_k^K \right] \\ &= \max_{w^J \boldsymbol{w}^K} \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} y_i x_{ijk} w_j^J w_k^K \right) = \max_{w^J \boldsymbol{w}^K} \sum_{j=1}^{J} \sum_{k=1}^{K} z_{jk} w_j^J w_k^K \end{aligned} \quad (6.108)$$

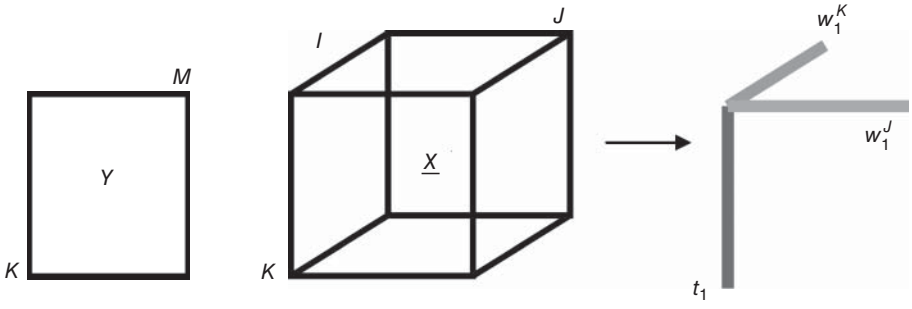


Figure 6.12 Scheme of *N*-way PLS regression of responses, $\boldsymbol{Y}$ ($K \times M$), on a three-way data array, $\underline{\boldsymbol{X}}$, after decomposition to one component $\boldsymbol{t}$ with its weighing vectors $\boldsymbol{w}$.

In matrix form, the criterion to be maximized becomes

$$\max_{w^J w^K}[(\boldsymbol{w}^J)^{\mathrm{T}} \boldsymbol{Z} \boldsymbol{w}^K] = \mathrm{SVD}(\boldsymbol{Z}) \tag{6.109}$$

where SVD is again the singular value decomposition (cf. Eq. (6.51)).

Generalization of *N*-PLS to higher order arrays $\underline{\boldsymbol{X}}$ and $\underline{\boldsymbol{Y}}$ is feasible. The principal algorithm for regressing a $\boldsymbol{y}$ vector on a three-way array $\underline{\boldsymbol{X}}$ is given in Example 6.11 and can be compared to the original PLS1 algorithm in Example 6.7.

Example 6.11 *Algorithm for N-PLS* [7]

Center $\boldsymbol{X}$ and $\boldsymbol{y}$

$a = 1$

1. Calculate matrix $\boldsymbol{Z}$.
2. Determine $\boldsymbol{w}^J$ and $\boldsymbol{w}^K$ by SVD.
3. Calculate $\boldsymbol{t}$.
4. $\boldsymbol{b} = (\boldsymbol{T}^{\mathrm{T}}\boldsymbol{T})^{-1}\boldsymbol{T}^{\mathrm{T}}\boldsymbol{y}$.
5. Each sample $\boldsymbol{X}_i$ is replaced by $\boldsymbol{X}_i - \boldsymbol{t}_i \boldsymbol{w}^J (\boldsymbol{w}^K)^{\mathrm{T}}$ and $\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{T}\boldsymbol{b}$.
6. $a = a + 1$. Continue from step 1 until proper description of $\boldsymbol{y}$.

Prediction: $\boldsymbol{y}_0 = \boldsymbol{T}\boldsymbol{b}_{\mathrm{A}} = \boldsymbol{X}_0\, \boldsymbol{b}_{\mathrm{PLS}}$

Important applications of *N*-PLS are in the area of multivariate calibrations for excitation/emission fluorescence spectrometry, for hyphenated analytical methods, such as HPLC/diode array detection and GC/MS, or for multidimensional separation techniques with or without coupling to spectroscopy.

6.3 Nonlinear Methods

In analytics, nonlinear relationships can be frequently modeled without the application of nonlinear methods. This is feasible by means of *transformations* of variables, such as signals or concentrations. Remember Beer's law in the form

$$I = I_0 e^{-\varepsilon_\lambda dc} \tag{6.110}$$

where I is the transmitted light intensity; I_0 is the incident intensity of light, and ε_λ, d, c have the meanings as given Eq. (6.72).

Parameters that cannot be transferred by any operation into linear parameters are denoted as being *intrinsically nonlinear.*

The signal intensity is related to the concentration in a nonlinear way. Logarithmic transformation, however, leads to a linear relationship ($A_\lambda = \varepsilon_\lambda dc$, Eq. (6.72)), so that the linear methods discussed can be used. The transformation in this example is based on a physical law, that is, it is of a *mechanistic* nature.

Another possibility consists of an *empirical* transformation on the basis of polynomials of higher order. In this context, we have already used quadratic polynomials for response surface methods in Section 4.2.

In this section, on the one hand, methods that are used to estimate intrinsically nonlinear parameters by means of nonlinear regression (NLR) analysis will be introduced. On the other hand, we will learn about methods that are based on nonparametric, nonlinear modeling. Among those are nonlinear partial least squares (NPLS), the method of alternating conditional expectations (ACE), and multivariate adaptive regression splines (MARS).

Nonlinear modeling on the basis of neuronal networks is discussed in Section 8.2.

Nonlinear Regression Analysis

Nonlinear parameters can be estimated by the methods of NLR. Chemical equilibria represent typical nonlinear models. For example, the retention behavior in HPLC is described in dependence on the pH value or the hydrogen ion concentration by a set of parameters of distribution coefficients and acid protolysis constants:

$$k' = \frac{k_0 + k_1 \frac{[H^+]}{K_{A_1}} + k_{-1} \frac{K_{A_2}}{[H^+]}}{1 + \frac{[H^+]}{K_{A_1}} + \frac{K_{A_2}}{[H^+]}} \tag{6.111}$$

where k' is the capacity factor as a relative measure for the retention time; $[H^+]$ is the molar concentration of hydrogen ions; k_0,

k_1, and k_{-1} are the distribution coefficients for the different forms of the analyte molecules HS, HS^+, and S^-; and K_{A_1} and K_{A_2} are the acid constants for HS and S^-, respectively.

In mathematical terms, the nonlinear model is expressed by the dependent variable *y,* the independent variable *x*, and the parameter b_i in Eq. (6.111):

$$y = \frac{b_0 + b_1 \frac{x}{b_3} + b_2 \frac{b_4}{x}}{1 + \frac{x}{b_3} + \frac{b_4}{x}} \tag{6.112}$$

In this equation, we note two types of parameters: first, the linear parameters b_0, b_1, and b_2, and then, the intrinsically nonlinear parameters b_3 and b_4. By appropriate reshaping of the equation, the linear parameters can be determined by the methods of linear algebra. In fact, there exist algorithms for separation of linear and nonlinear parameters in a given model.

The nonlinear parameters can be estimated by means of NLR. There, however, no closed solution is feasible, but the parameters are iteratively approximated on the basis of initial values for them. The following approximation methods can be distinguished:

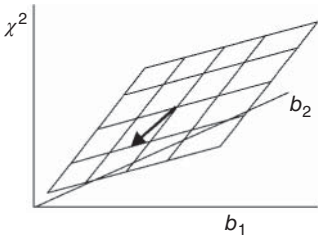
- The method of steepest descent
- Linearization methods
- Search methods.

In all cases, an objective function, χ^2, is minimized that represents the deviations of the *n* estimated *y* values from the observed ones:

$$\chi^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \Rightarrow \text{Minimum} \tag{6.113}$$

Method of Steepest Descent

With this method, the direction of steepest descent is searched on a plane or hyperplane of the objective function in dependence on the parameters of the model. The basis is a design, for example, 2^m, in the *m* parameters where the objective function, χ^2 (Eq. (6.110)), is approximated by means of a linear model in the parameters ***a:***



$$\chi^2 = a_0 + \sum_{j=1}^{m} a_j \frac{[b_j - \overline{b}_j]}{s_j} + e \tag{6.114}$$

where $\overline{b}_j$ is the center of the design and s_j is the standard deviation for the parameters as a normalization factor.

The direction of steepest descent is given by the values $-1 \cdot a_j$, that is,

$$\frac{[b_j - \overline{b}_j]}{s_j} = \lambda(-a_j) \tag{6.115}$$

or

$$b_j = \overline{b}_j - \lambda a_j s_j \tag{6.116}$$

where λ is the descent parameter.

The method is repeated with new parameter values b_j as long as the minimum is found.

Linearization Methods

In contrast to linear parameters, nonlinear parameters cannot be represented simply by the product between the matrix of independent variables, $\boldsymbol{X}$, and the parameter vector, $\boldsymbol{b}$ (cf. Eq. (6.13)). If, however, approximate initial values, $\boldsymbol{b}_0$, are known for the parameter vector, $\boldsymbol{b}$, then the function $y = f(x)$ can be rewritten such that a linear function emerges in dependence on the difference of the parameter to be estimated and its starting value, that is, $\Delta \boldsymbol{b} = \boldsymbol{b} - \boldsymbol{b}_0$. This means that the method can be traced back to the solution of a least squares problem.

The basis for linearization is the extension of the function into a Taylor series and truncation after the first-order term under the prerequisite that the difference $\Delta \boldsymbol{b}$ is small enough:

$$y_i = f(x_i, b_1, b_2, \ldots, b_m) + e_i \tag{6.117}$$

$$\begin{aligned} y_i = f(x_i, \boldsymbol{b} = \boldsymbol{b}_0) &+ \left.\frac{\delta f_i}{\delta b_1}\right|_{\mathbf{b}=\mathbf{b}_0} \Delta b_1 \\ &+ \left.\frac{\delta f_i}{\delta b_2}\right|_{\boldsymbol{b}=\boldsymbol{b}_0} \Delta b_2 + \cdots + \left.\frac{\delta f_i}{\delta b_m}\right|_{\boldsymbol{b}=\boldsymbol{b}_0} \Delta b_m + e_i \end{aligned} \tag{6.118}$$

where $\left.\frac{\delta f_i}{\delta b_j}\right|_{\boldsymbol{b}=\boldsymbol{b}_0}$ is the first derivative of the parameters b_j at point i at the initial value $\boldsymbol{b}_0$. Equation (6.115) is then linear in the parameters Δb_1, Δb_2, …, Δb_m and similar to the general equation for linear regression (Eq. (6.13)). The matrix of the independent variables, $\boldsymbol{X}$, now contains, however, the partial first derivatives of the function to the parameters in the form

$$\begin{pmatrix} \frac{\delta f_1}{\delta b_1} & \frac{\delta f_1}{\delta b_2} & \cdots & \frac{\delta f_1}{\delta b_m} \\ \frac{\delta f_2}{\delta b_1} & \frac{\delta f_2}{\delta b_2} & \cdots & \frac{\delta f_2}{\delta b_m} \\ \vdots & & & \\ \frac{\delta f_n}{\delta b_1} & \frac{\delta f_n}{\delta b_2} & \cdots & \frac{\delta f_n}{\delta b_m} \end{pmatrix} = \boldsymbol{X} = \boldsymbol{J} \tag{6.119}$$

The matrix is termed the *Jacobian matrix*, J. The linearized model reads then

$$\Delta \boldsymbol{y} = \boldsymbol{J}\Delta \boldsymbol{b} + \boldsymbol{e} \tag{6.120}$$

where the vector $\Delta \boldsymbol{y}$ represents the difference between the measured y values and the predicted value at position $\boldsymbol{b}_0$, that is,

$$\Delta \boldsymbol{y} = \begin{pmatrix} y_1 - \widehat{y}_{1,0} \\ y_2 - \widehat{y}_{2,0} \\ \vdots \\ y_n - \widehat{y}_{n,0} \end{pmatrix} \tag{6.121}$$

Estimation of the parameter vector $\Delta \boldsymbol{b}$ is carried out in analogy to Eq. (6.14) by

$$\Delta \boldsymbol{b} = (\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J})^{-1}\boldsymbol{J}\Delta \boldsymbol{y} \tag{6.122}$$

The vector of the nonlinear parameters is finally obtained by

$$\boldsymbol{b} = \boldsymbol{b}_0 + \Delta \boldsymbol{b} \tag{6.123}$$

The accuracy of parameter estimations depends first on good starting values. Bad initial values may lead to slow convergence or oscillations.

Second, the quality of parameter estimates is also determined by the appropriateness of series extension without considering higher order terms. If the first order is not sufficient, divergences cannot be ruled out. In principle, Taylor expansion can also be performed by inclusion of the second derivative (*Hessian matrix*).

The derivation of a mechanistic model can be carried out analytically; this is in contrast to empirical models, which require numerical derivatives.

Search Methods

A particular search method has already been introduced in the form of the *simplex method* in Section 4.3. Although it was explained in connection with the optimization of experimental observations, it can be transferred analogously to the optimization of the objective function in Eq. (6.113).

A very simple method is the *grid search,* where each point at the given grid is evaluated and in that way the minimum is found. If necessary, the grid is reduced to estimate the parameters locally more precisely.

Extrapolations outside the areas that were used for building the model are more critical for nonlinear methods than for linear ones.

In the case of the *Monte Carlo method*, the optimum is searched by random change of the parameters to be estimated. The best set of parameters is maintained in subsequent computations.

Marquardt Algorithm

In practice, a combination of the method of steepest descent and the linearization procedure is preferred. The algorithm is based on a proposition by Levenberg, which was further developed by Marquardt. As long as the parameter estimations are still poor, the algorithm operates on the basis of the steepest descent method. In the course of optimization, the linearization method is progressively included.

Regression Diagnostics

A particular cost criterion of nonlinear modeling is the minimization criterion, γ^2, by itself. If this criterion becomes zero, a perfect fit is obtained. Similarly to linear regression analysis, the residuals can be graphically inspected and the recovery function can be evaluated.

More difficult is the estimation of errors for the nonlinear parameters, since no variance–covariance matrix exists. Frequently, the error estimations are restricted to a locally linear range. In the linearization range, the confidence bands for the parameters are then calculated as in the linear case (Eqs. (6.25)–(6.27)). An alternative consists in error estimations on the basis of Monte Carlo simulations or bootstrapping methods (cf. Section 8.2).

Nonparametric Methods

Alternating Conditional Expectations

At the beginning of Section 6.3, the possibilities of transformations of variables for modeling nonlinear relationships were discussed. In the ACE method, these transformations need not be predefined, but are found by the algorithm. To understand the ACE method, we start from the linear multivariate model for a single dependent variable y and p independent variables x_j (cf. Eq. (6.13)):

$$y = b_0 + \sum_{j=1}^{p} b_j x_j + e \tag{6.124}$$

where b_j is the regression parameter.

The model of the ACE method is an analogy [8]:

$$g(y) = \sum_{j=1}^{p} f_j(x_j) + e \tag{6.125}$$

here $g(y)$ is a transformation of the y variable and the f_j are transformations of the variables x_j. The transformation functions are smooth, but unconstrained functions of the variables y and x.

The ACE model is stored in the form of p pairs of $[y_j, g(y_j)]$ and $[x_{ij}, f(x_{ij})]$. The transformation functions are not in closed or analytical form in contrast to parametric models. The transformation is obtained on the basis of an optimality criterion, according to which the variance of the error e in Eq. (6.122) is minimized with respect to the variance of the transformed variable y.

The notion of ACE derives from the algorithm, which serves the estimation of the optimum transformations. Based on initial estimates of the transformations (g^0 and f^0), the algorithm improves the function g^k and subsequently f^k. If, for example, f^k is given, then an improved version for g is obtained by computation of

$$g^{k+1}(y) = \frac{E\left[\sum_{j=1}^{p} f_j^k\left(x_j\right) \middle| y\right]}{\left\| E\left[\sum_{j=1}^{p} f_j^k\left(x_j\right) \middle| y\right] \right\|} \tag{6.126}$$

E denotes here the expectation value and $\|\cdot\| = [E(\cdot)]^{1/2}$.

In analogy, an improved estimation of f is computed if the actual estimation g^k for g is

$$f_j^{k+1}(x_j) = E\left[g^k\left(y\right) - \sum_{i \neq j}^{p} f_j^k(x_i) \middle| x_j\right] \tag{6.127}$$

The algorithm uses conditioned expectation operators in alternating sequence. It converges to the optimum transformations $g^k \to g$ and $f^k \to f$.

In the case of evaluating real data sets, the algorithm in Eqs. (6.126) and (6.127) only serves the purpose of providing the mathematical basis of how the algorithm should perform in principle. The operator of conditional expectations is replaced by an appropriate *smoothing algorithm*.

The actual transformations can be graphically evaluated. This is shown in Figure 6.13 for the data from Example 6.12. Linear relationships are also reflected by linear transformations.

Prediction of y values is carried out in two steps. First, the transformations $f(x)$ on the basis of the x values are evaluated in a look-up table. Usually, interpolation between two points will be necessary. After that, the p functional values are inserted into Eq. (6.125) and the y value is calculated by adding up the individual functions.

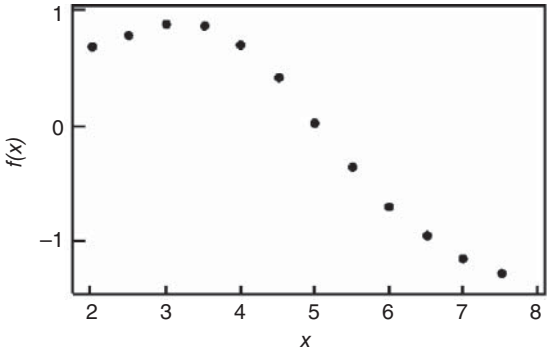


Figure 6.13 Plot of the transformation of an *x* variable in the alternating conditional expectations (ACE) method. The plot is based on the data in Example 6.11, that is, the *x* variable corresponds here to a pH value.

For predictions, the ACE model can be used until this stage only if the y values are not transformed, that is, if $g(y) = y$.

The main advantage of the ACE algorithm is the diversity of possible transformations. For collinear data, a previous reduction of dimensionality of the $\boldsymbol{X}$ matrix is to be recommended, for example, by means of principal component analysis.

Nonlinear PLS

The linear PLS method has been introduced in Section 6.2 and also in connection with multicomponent analysis. The NPLS method fits an NLR model to the data. This model is based on a nonlinear inner relationship of the PLS algorithm, which is described by means of a smoothing function [9]. The inner relationship denotes regression of the scores of the $\boldsymbol{Y}$ matrix on the scores of the X matrix. According to the algorithm in Example 6.7 (step 7), after rearranging for a given dimension, we obtain the *inner relationship:*

$$\boldsymbol{u} = f(\boldsymbol{t}) + e \tag{6.128}$$

where $\boldsymbol{u}$ represents the score vector of the $\boldsymbol{X}$ matrix, $\boldsymbol{t}$ is the score vector of the $\boldsymbol{Y}$ matrix, e are the residuals, and f is the smoothing function. In the first step, the latent variables are computed as in the linear PLS. After that, instead of linear approximation of $\boldsymbol{u}$ on $\boldsymbol{t}$, the nonlinear relationship is constructed by means of the smoothing function. The smoothing function is the same as in the ACE method. The nonlinear relationship can be analyzed graphically by plotting the latent variables. In Figure 6.14, this is

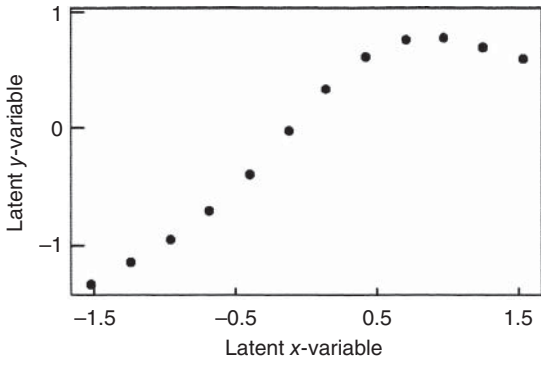


Figure 6.14 Demonstration of the nonlinear relationship between latent variables for the univariate data in Example 6.10.

demonstrated for univariate data. For a multivariate data set, one obtains for each latent x variable an individual plot.

Multivariate Adaptive Regression Splines

MARS represents one of the most complex nonlinear methods [10]. This method is based on the idea that only few variables significantly influence the dependent variable in subspaces of the multidimensional space. If these subspaces can be identified, and therefore the corresponding variables assigned, it should be possible to approximate linear or cubic splines to the observations. The fundamentals of *spline functions* were discussed in Section 3.1.

The model for a *multivariate regression spline* is based on linear combinations of univariate spline functions S_{kj} of the general form [10]

$$y_i = b_0 + \sum_{k=1}^{K} b_k \prod_{j=1}^{J} [S_{kj}(x_{ij} - t_{kj})] + e_i \tag{6.129}$$

where b_0 and b_k are the parameters of the linear combinations, t_k is the knot at point k, K is the number of knots of the spline, and J is the number of basis functions.

The nonlinearity of the method results from the fact that both the spline coefficients and also the linear combinations are optimized with respect to the problem at hand. *Adaptive* means here that the subranges are fit to the concrete data set. The knots are therefore not fixed as in conventional splines. The search for the most appropriate positions of the knots may be very time intensive.

Table 6.5 pH dependence of the retention (k') of anthranilic acid in HPLC.

pH	k'
2.0	10.60
2.5	14.28
3.0	16.63
3.5	17.60
4.0	15.64
4.5	13.31
5.0	8.09
5.5	4.48
6.0	2.60
6.5	1.92
7.0	1.56
7.5	1.45

Example 6.12 *Nonlinear Methods*

The retention behavior of anthranilic acid in dependence on the pH value is to be modeled. To solve this problem, first, the method of nonlinear regression is exploited to estimate the parameters according to a physicochemical model given in Eq. (6.111) and formalized in Eq. (6.112). In a second step, the nonparametric methods NPLS and ACE are applied.

The measured k' values are given in Table 6.5 and plotted in Figure 6.15. Regression by means of NLR provides the parameters summarized in Table 6.6, together with their standard errors.

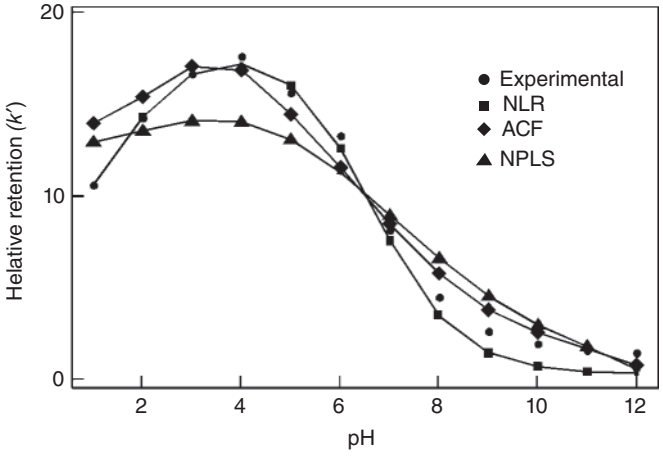


Figure 6.15 Modeling of the retention (k') of anthranilic acid in HPLC in dependence on the pH value.

Table 6.6 Parameter for the model of retention (k') of anthranilic acid in HPLC according to Eq. (6.112).

Parameter	Estimation	Standard error
b_0	18.55	0.31
b_1	4.979	1.912
b_2	0.300	0.116
b_3	7.06×10^{-3}	2.27×10^{-3}
b_4	1.52×10^{-5}	0.11×10^{-5}

The *standard error* of the parameters b_i is calculated from the diagonal elements of the variance–covariance matrix in Eq. (6.22):

$$\sqrt{s^2_{b_{ii}}}$$

The parameters can be interpreted in a physicochemical sense. For example, the acid protolysis constants can be compared with those determined by potentiometry. The total error for modeling is given in Table 6.7 by the *PRESS* value (Eq. (6.99)) and the standard error (in analogy to Eq. (6.106)).

Table 6.7 Error for modeling the pH dependence of the retention (k') of anthranilic acid in HPLC.

Method	PRESS	Standard error
NLR	7.647	0.798
NPLS	46.72	1.973
ACE	22.06	1.356

Table 6.7 also demonstrates the results of modeling with the nonparametric methods NPLS and ACE. The computed retention values are plotted for all three methods in Figure 6.14. Both plots demonstrate the superiority of the parametric NLR method. It should be noted that only by regression on the basis of the mechanistic model in Eq. (6.112) can the independent variable be maintained in its antilogarithmic form, that is, as hydrogen ion concentration. For the nonparametric computations, an acceptable fit of the k' values is only feasible if pH values are applied. This is explained by the fact that the nonparametric methods can only be used in the sense of curve fitting, but not in the sense of model approximation.

Scientifically, a mechanistic model is always to be preferred if it can be constructed. In practice, however, empirical modeling is frequently sufficient, so that the nonparametric nonlinear methods are also important tools.

Regression Trees

Another nonparametric regression method is CART (classification and regression trees). The basic concepts were outlined in "Discriminant Analysis" Section about tree-based classification. We remember from that chapter that CART is a recursive binary partition method based on a simple model constant for each region. If the residual sums of squares of responses is minimized,

it can be shown that the best constant, c_m, for regression trees is just the average of the responses, y_i, in region R_m:

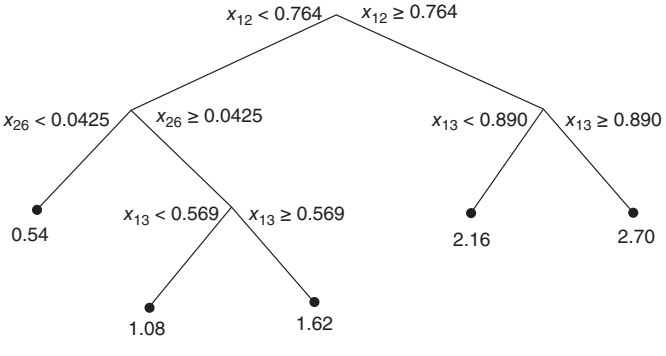
$$\hat{y} = \sum_m c_m I(x \in R_m) \tag{6.130}$$

The unit step function yields for $I(x \in R_m) = 1$ and for $I(x \notin R_m) = 0$.

In Example 6.13, a regression tree model for calibration and prediction of an organic pollutant from ultraviolet spectra and the model improvement by bagging are described.

Example 6.13 *Regression Trees*

A calibration model for prediction of benzanthracene on the basis of UV spectra was constructed by regressing concentrations of benzanthracene in 25 sample solutions on UV absorbances at 27 wavelengths by CART. The regression tree has the following form:



Resubstitution of the benzanthracene concentrations reveals perfect fit, that is, the calibration mean squared error (Eq. (6.106)) is almost zero. Estimation of predictions from a single tree by leave-one-out cross-validation reveals a mean squared prediction error of 0.428. A plot of the recovery function for the individual predictions is given in Figure 6.16a. Further improvement of the model is feasible if again ensemble methods (cf. "Tree-Based Classification" Section) are applied. Figure 6.16b shows the recovery function for a *bagged* model with a smaller prediction error of 0.337.

Ensemble methods are not restricted to tree-based methods but can also be applied for *k*-NN or discriminant analyses.

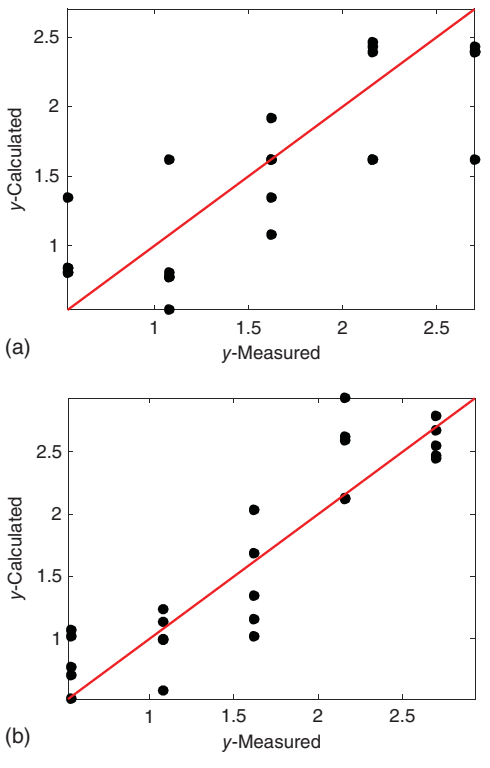


Figure 6.16 Recovery functions for cross-validated predictions of benzanthracene from UV-spectra based on a single tree (a) and a bagged tree model (b).

General Reading

1. Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning*, Springer, New York.
2. Draper, N. and Smith, H. (1981) *Applied Regression Analysis*, 2nd edn, John Wiley & Sons, Inc., New York.
3. Box, G.E.P. and Draper, N.R. (1987) *Empirical Model. Building and Response Surfaces*, John Wiley & Sons, Inc., New York.
4. Weisberg, S. (1985) *Applied Linear Regression, 2nd edn* edn, John Wiley & Sons, Inc., New York.
5. Deming, S.N. and Morgan, S.L. (1987) *Experimental Design: a Chemometric Approach*, Elsevier, Amsterdam.
6. Trygg, J. and Wold, S. (2002) *J. Chemom.*, **16**, 119.
7. Bro, R. (1996) *J. Chemom.*, **10**, 47.
8. Breiman, L. and Friedman, J.H. (1985) *J. Am. Stat. Assoc.*, **80**, 580.
9. Frank, I.E. (1990) *Chemometrics Int. Lab. Syst.*, **8**, 109.

10. Friedman, J.H., Stuetzle, W., and Grosse, E. (August 1988) Multidimensional Additive Spline Approximation. Technical Report No. 101, Stanford University, California.
11. de Boor, C. (1978) *A Practical Guide to Splines*, Springer, Berlin.
12. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. **2**/**3**, Elsevier, Amsterdam.
13. Varmuza, K. and Filzmoser, P. (2009) *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, Boca Raton, FL, Berlin.

Questions and Problems

1. Decide which of the following models contains intrinsically nonlinear parameters, which cannot be estimated by linear regression analysis:

$y = b_0 + b_1 x + b_{111} x^3$

$y = b_0 + e^{-b_1 x}$

$y = b_0 + b_1 \frac{1}{x^2}$

$y = b_1 x + \frac{1}{x + b_2}$

The following data were obtained for liquid chromatograms of standard solutions of atrazine. The concentrations are given in mmol l^{-1} and the signals as relative peak area.

Concentration	Area
0.2	0.85
0.2	0.83
0.5	1.34
1.0	2.15
2.0	3.70
3.0	5.15
4.0	6.50
5.0	7.75
6.0	8.90
7.0	9.95

(a) Plot the data and select an appropriate calibration model.

(b) Estimate the parameters of the model by linear repression analysis.

(c) Calculate the confidence intervals of the parameters.

(d) Investigate the residuals of the model and perform a complete ANOVA in order to test the significance of the calibration parameters.

2. Which variations describe the sums of squares in the F tests for goodness-of-fit and lack-of-fit of a model?

3. In which situations are biased regression methods particularly useful compared to ordinary linear regression analysis?

4. Summarize the advantages and disadvantages of direct and inverse multivariate calibration.

5. What is the difference between an outlier and an influential observation and how are they detected?
6. Explain the differences between parametric and nonparametric nonlinear regression methods and give examples of their applications.

7
Analytical Databases

Learning Objectives

- To introduce the principles of representation, conversion, and storage of analytical data
- To code spectra and chemical structures in analytical databases and to learn about library search methods and the simulation of spectra.

Chemical databases serve different purposes, such as the search for scientific and patent-related literature or retrieval of facts about chemical compounds. In analytical chemistry, the databases that are of interest are those that contain either original measurements (spectra and chromatograms) or derived data, such as concentrations or chemical structures. These data can be retrieved online via network from a host, for example, STN International. On the other hand, databases can be stored at individual PCs or in connection with an analytical instrument.

Examples of some of the oldest analytical databases are given in Table 7.1. At present, databases that comprise several spectrometric methods, such as SDBS (Spectral database for organic compounds, AIST), exist and in SciFinder (CAS), several databases are combined to reveal more than 42 million of spectra. Apart from representation of the analytical measurements in the computer, the coding of chemical structures is an important aspect of constructing analytical databases.

Efficient retrieval of the analytical information depends on appropriate *search strategies.* To confirm a chemical structure on the basis of its spectrum, the database must contain the sought-for spectrum. Very often, however, no spectrum related to the assumed chemical structure is available, so then methods for *simulation of spectra* from a chemical structure are needed.

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

Table 7.1 Some analytical databases.

Method	Database /supplier	Data	Compounds	Remarks
NMR	SpecInfo (CC)	85 000 spectra	150 000	On-line
	Bruker	19 000 spectra	—	Spectrometer /PC
MS	NIST/EPA /MSDC	210 000 spectra	200 000	On-line/PC
	John Wiley & Sons	125 000 spectra	110 000	PC
IR	Sadtler	80 000 spectra	—	PC
	Aldrich-Nicolet	>10 000 spectra	10 600	Spectrometer /PC
Atomic emission	Plasma 2000 (PE)	50 000 atomic lines	60 elements	PC
GC	Sadtler	Retention indices	—	PC

7.1 Representation of Analytical Information

Type of Information

Different sorts of information for analytical databases exist:

- Numerical
- Alphanumerical, for example, text
- Topological
- Graphical.

Numerical data are in question, if spectral, chromatographic, or electroanalytical data have been measured and, if concentrations, errors, or analysis costs are to be stored. Typical alphanumeric data concern descriptions of sample identity or analytical procedures. Chemical structures are represented topologically from electron microprobe analysis.

Structure of Databases

The demands for storing and processing information dictate the format of a database. Usually, the content of the database is acquired in different steps.

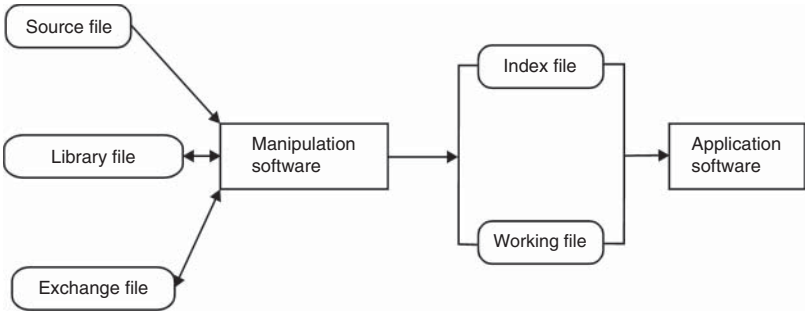


Figure 7.1 Structure of analytical databases.

Source and Library Files

The raw analytical data are stored in the source file (cf. Figure 7.1). Elimination of unimportant data, filtering, transformations, or compression of data leads to a library file, which is archived. The library files of an analytical database consist of a header and a collection of data blocks. The header contains information about the file organization, as well as control parameters. Stored in the data block are different sets of data that contain information about the analytical data, such as spectra, and about chemical structures and additional remarks.

Exchange Files

To transfer data between different bulk storage media, files of fixed exchange formats are created. The most important exchange format in spectroscopy is that elaborated by the JCAMP/DX format, which was elaborated by the Joint Committee on Atomic and Molecular Data with the following objectives:

- Different sorts of spectra should be describable, for example, Fourier transformation/infrared (FT/IR), Raman, UV/VIS spectrometry, X-ray diffraction spectrometry, nuclear magnetic resonance (NMR), or MS.
- The text of the file should be readable by computer, humans, and telecommunication systems alike. Therefore, only ASCII characters are allowed.
- Descriptive information about the sample should be compact.
- The format must be flexible enough to guarantee later extensions.

Table 7.2 demonstrates an example of a JCAMP/DX data file for storing the IR spectrum of epichlorohydrin vapor. Here, the minimum information is given. Further items concern information about the compound, for example, the molecular mass or the

Table 7.2 Important information of a JCAMP/DX exchange file for an IR spectrum.

1.	##Title = Epichlorohydrin vapor
2.	##JCAMP-DX = 4.24
3.	##DATA TYPE = INFRARED SPECTRUM
4.	##ORIGIN = Sadtler Research Laboratories
5.	##OWNER = EPA/Public Domain
⋮	
	Rows 6–24 are optional
⋮	
25.	##XUNITS = 1/CM
26.	##YUNITS = ABSORBANCE
27.	##XFACTOR = 1.0
28.	##YFACTOR = 0.001
29.	##FIRSTX = 450.
30.	##LASTX = 4000.
31.	##NPOINTS = 1842
32.	##FIRSTY = 0.058
33.	##XYDATA = (X++(..Y))
34.	450 58 44 34 39 26 24 22 21 21 19 16 15 15 17
35.	etc.
36.	3998 15 15 14
37.	##END=

Chemical Abstract number, about sample preparation, the instrument used, or about measuring conditions and data processing methods, such as smoothing or derivatives.

The format is general enough that it can be exploited for similar purposes. Thus, a convention exists for describing chemical structures (JCAMP/CS format where CS stands for chemical structure).

Coding of Spectra

For a long time, the limited storage capacity of bulk storage media hindered the complete storage of spectra and chromatograms in an analytical database. Therefore, many spectroscopic databases contain only features of the spectra. For example, UV spectra are based on maximum absorbances or IR spectra are represented as peak lists (cf. Figure 7.2).

Now, one attempts to store the complete analytical information, that is, full spectra, complete chromatograms, or even spectrochromatograms that are generated by hyphenated systems, such as HPLC with diode array detection. Usually, the digitization rate is determined by the measuring conditions. Table 7.3 exemplifies the representation of signal position and

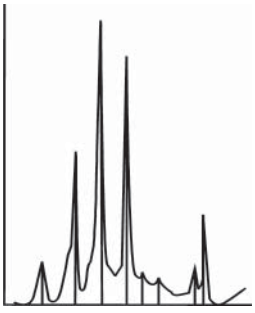


Figure 7.2 Evaluation of the peak list of a spectrum.

Table 7.3 Coding of spectra in the molecular database of the Organic Institute of ETH Zurich (OCETH).

Method	Signal position	Intensity	Number of data points
^{13}C-NMR	0.1 ppm	0.1%	100
MS	0.1 mass number	0.1%	500
IR	1 cm^{-1} (200–2000 cm^{-1}) 4 cm^{-1} (2000–4000 cm^{-1})	%transmission at 63 levels	2300
UV/VIS	1 nm	0.01 in lg ε	400

intensity of molecular spectra as stored in the beginning of the 1980s in a database of the ETH Zurich.

As a rule, the databases contain additional information, for example, about the starting point of a measurement, resolution, or multiplets in NMR spectroscopy.

Apart from the storage of full spectra at certain digitalization levels, several algorithms are known to represent the original data in a compact, and for further processing, efficient way. As an extreme case, the measured data or chemical structures are encoded as binary vectors of constant lengths.

As a rule, the user does not need to know about the details of data compression. In addition, because of the increasing performance of computers, the necessity for compressing the original data decreases, at least for analytical data banks.

Recent developments in spectroscopy and in analytical separation science, in the hyphenation of those methods as well as in multidimensional separations, have led to a dramatic increase in the amount of data emerging from a single analytical measurement. For example, a gas chromatography–mass spectrometry (GC-MS) experiment with a capillary column and a

high-resolution mass spectrometer might produce a data file in the order of 2 or even 20 GB. Storage of those data in ASCII files is prohibitive. Therefore, storage as binary data or new formats are exploited. Such a new format is Base 64 used in storage of, for example, protein mass spectra by mzXML (m/z Extensible Markup Language). As an example, an ASCII file of the size of 816 MB could be stored in a binary file in a size of 234 MB. Using the Base 64 format compression is possible to 312 MB and, after elimination of all zero intensities, even further to 13 MB, that is, data reduction from the ASCII file to Base 64 is achieved by more than 60-fold.

Binary data storage is common in the separation sciences. Here, the Network Common Data Format (Netcdf) is used with the following characteristics:

- Data arrays
- Binary data
- Self-explanatory
- Portable (independent of computer platform and storage media)
- Scalable (effective access to subsets of big data sets)
- Extendable (without new definitions of data structures)
- Simultaneously accessible
- Archivable.

Coding of Chemical Structures

Fragmentation Codes An easy possibility for converting a chemical structure into a computer-readable format is based on fragmentation codes. Typical fragmentations are aromatic rings, structural skeletons, the OH group, and the azo group (–N=N–). The fragmentations are numbered and stored in a fragmentation list. More generally, a fragmentation is formed on the basis of a freely eligible center of the molecule and is described by its first to fourth spheres by means of hierarchically ordered symbols.

Consider a conventional encoding of chemical structures in ^{13}C-NMR spectroscopy as introduced with the so-called HOSE code (hierarchically ordered spherical description of environment). Table 7.4 contains some symbol descriptions of this code.

The molecule given in Figure 7.3 is to be coded by the HOSE code.

The carbon atoms around the center of the molecule (bold face C atom) in the first sphere are described according to Table 7.3 by

∗ C ∗ CC(

Table 7.4 Symbols for the HOSE substructure code ordered by priority.

Symbol	Meaning
R	Ring
%	Triple bound
=	Double bond
*	Aromatic bond
C	C
O	O
N	N
S	S
X	Cl
Y	Br
&	Ring closure
,	Separator
(//)	Sphere separator

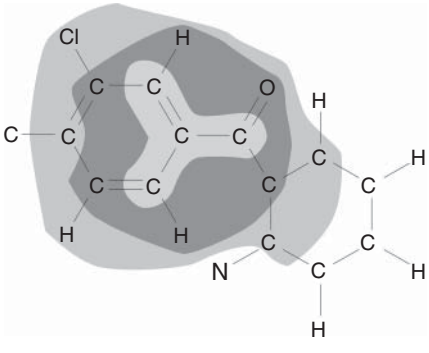


Figure 7.3 Spheres around a carbon atom (*bold face*) as the basis for encoding the structure by the hierarchically ordered spherical description of environment (HOSE) code.

Brackets symbolize the end of the first sphere. The second sphere reads then as

$$*\,C, *\,C, = OC/$$

Here the end of the sphere is characterized by the symbol /. Analogously, the third and fourth spheres are obtained and the HOSE code for all of the four spheres of the molecule in Figure 7.3 is

$$*\,C *\,CC(*\,C, *\,C, = OC/ *\,CX, *\,\varsigma, , CC/ *\,\varsigma C, , CN, C)$$

Although with this kind of code, a fragmentation can be represented up to the fourth sphere, in many applications, the

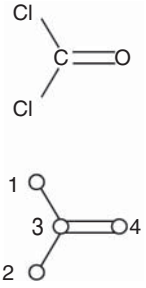


Figure 7.4 Chemical structure of phosgene represented as an undirected graph.

description of the first sphere is a good approximation. Rings are characterized by special fragmentation codes, for example, the hierarchically ordered ring description (HORD code) .

The HOSE code represents a *linear notation* of a chemical structure. Other codes are ROSDAL, SYBYL, and SMILES. The latter code is currently the most popular linear notation.

In principle, the assignment of fragmentations is easy and unique. Thus, fragmentation codes are also used in structure elucidation in IR spectroscopy, although the vibrations of a molecule are coupled and a decomposition of the molecule into individual fragmentations is sometimes misleading for the interpretation process.

Generation of a molecule from its fragments is carried out by a *structure generator.*

The number of theoretically possible chemical structures for a given molecular formula can be very large. For example, a structure generator could generate more than 151 million isomers for a molecule with the molecular formula $C_6H_6N_4O$ (mass = 150 Da). In contrast, the BEILSTEIN database of organic compounds contains only 273 structures for this molecular formula [4].

The easy assignment of fragmentations does not provide the structure of the unknown compound. In the next step, the fragmentations have to be connected and the hypothetical structure has to be compared with candidate structures, atom by atom. Of course, only those molecules that contain all of the found fragmentations have to be considered as candidates.

Matrix Representation of Chemical Structures A mathematical description of a chemical structure can be derived by means of *graph theory.* In Figure 7.4, the graph of the molecule phosgene is given as an example.

The atoms of the molecule form the nodes of the graph and the bonds the edges (cf. Table 7.5). The coding of the graph is performed by the so-called *adjacency matrix A.* For every element, a_{ij}, of this matrix, the following is valid: if the node K_i is connected with another node K_j, then its value is 1; otherwise, it is 0.

Table 7.5 Representation of a chemical structure based on undirected graphs.

Chemical term	Graph theoretical term	Symbol
Molecular formula	Molecular graph	G
Atom	Node	K
Covalent bond	Edge	G
Free electrons	Loop	n
Topological map	Adjacency matrix	A

Example 7.1 *Adjacency Matrix*

For the phosgene molecule, we obtain:

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \tag{7.1}$$

The additional numbers above and at the side of the brackets correspond to the nodes in the graph as numbered in Figure 7.4.

In the adjacency matrix, no bonds are considered. This would be possible by the analogous representation of a molecule based on the *bond electron matrix*, or BE matrix for short. In the latter, the g-fold connection of two nodes as well as the number of n free electrons of an atom is accounted for.

Example 7.2 *BE matrix*

For the phosgene molecule, the BE matrix B is given by

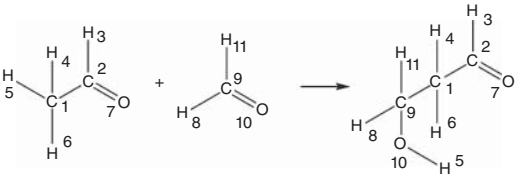
$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 6 & 0 & 1 & 0 \\ 0 & 6 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 0 & 0 & 2 & 4 \end{pmatrix} \end{array} \tag{7.2}$$

The advantage of matrix representation consists of the fact that all of the matrix operations can be applied to the encoded chemical structures. Simple subtraction of two *BE* matrices provides,

for example, a reaction matrix, ***R,*** which consists of broken (minus sign) and made (plus sign) bonds. This reaction matrix is obtained as the difference of the *BE* matrices from the beginning, ***B,*** and to the end, ***E,*** of a reaction.

Example 7.3

BE matrices for the reaction of acetaldehyde and formaldehyde to form 3-hydroxypropanal:



$$\boldsymbol{B} = \begin{bmatrix} 0&1&0&1&1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&2&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0 \\ 0&2&0&0&0&0&4&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0&2&1 \\ 0&0&0&0&0&0&0&0&2&4&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix} \boldsymbol{E} = \begin{bmatrix} 0&1&0&1&0&1&0&0&1&0&0 \\ 1&0&1&0&0&0&2&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0 \\ 1&0&0&0&0&0&0&0&0&0&0 \\ 0&2&0&0&0&0&4&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&1&0&1&1 \\ 0&0&0&0&1&0&0&0&1&4&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$$

$$\boldsymbol{R} = \boldsymbol{E} - \boldsymbol{B} = \begin{bmatrix} 0&0&0&0&-1&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ -1&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&-1&0 \\ 0&0&0&0&1&0&0&0&-1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

As can be deduced from Eqs. 7.1 and 7.2, the quadratic matrices are symmetric around the main diagonal. However, even in the case where the matrices are stored as triangular matrices, a

disadvantage arises from the need for high storage capacity that accompanies representation of a chemical structure in this way. In addition, the graphs are sparsely connected. Therefore, many matrix elements are equal to zero and many parts of the matrix are redundant.

Connection Tables In practice, the node-oriented connection table is applied. This table can be derived from the connection matrix of atoms (cf. Table 7.6) and contains only the numbered chemical elements, the bonds connected to the atoms, and the type of the actual bond. In Table 7.7, the connection table for the phosgene molecule is written down. A less redundant connection table representation is given in Table 7.8.

Table 7.6 Connection matrix for the molecule phosgene (cf. Figure 7.4).

	1	2	3	4
1	Cl	0	1	0
2	0	Cl	1	0
3	1	1	C	2
4	0	0	2	O

The numbering of atoms in the considered molecule is arbitrary. Therefore, many connection tables are feasible for a given molecule. Mathematically the number of connection tables for a molecule consisting of n atoms is calculated from factorials. A molecule of 3 atoms, e.g., hypochloric acid, Cl–O–H, yields 3! = 6 connection tables. For a 12-atom molecule 12! = 479.001.600 connection tables are theoretically possible.

Table 7.7 Bond atoms and bonds in a connection table of phosgene (cf. Figure 7.4).

Atom number	Atom symbol	Atom1	Bond	Atom2	Bond	Atom3	Bond
1	Cl	3	1	—	—	—	—
2	Cl	3	1	—	—	—	—
3	C	1	1	2	1	4	2
4	O	3	2	—	—	—	—

1 Single bond and 2 double bond.

Table 7.8 A nonredundant connection table for phosgene (cf. Figure 7.4).

Node no.	Atom	Connected to	Bond
1	Cl		
2	Cl	3	1
3	C	1	1
4	O	3	2

1 Single bond and 2 double bond.

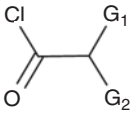


Figure 7.5 Example of a Markush structure with the general groups G_1 of phenyl, naphthenyl, *N*-pyridinyl, and G_2 of cycloalkyl.

As a general rule, in the representation of chemical structures, the hydrogen atoms are not coded at all. If necessary, for example, for graphical representation of a molecule, hydrogen atoms can be added by a suitable algorithm automatically.

Connection tables can be easily extended in their rows, for example, by having information about alternating bonds, cyclic and noncyclic bonds, stereochemistry or by description of variable positions or generic groups in a molecule. Generic groups are represented by Markush structures.

Markush Structures Searching for chemical structures is often related to searching for a whole family of structures rather than for a single compound. The description of general, so-called generic classes is feasible by means of Markush structures (cf. Figure 7.5). Such a structure consists of a core with well-defined atoms and bonds and at least one general group. The general groups are additionally specified. Substructure search is then performed for a whole compound class. This problem is especially important in the field of patent literature where the most general claim for a compound is envisaged. Eugene A. Markush was the first to apply for a patent in the United States in 1923 that claimed for the generic class of a chemical compound.

Canonization Important for the representation of a chemical structure as matrix or table is the unique assignment (canonization) of atoms in the structure. This can be calculated, for example, by Morgan's algorithm.

Working and Index File

In Figure 7.1, two additional files are mentioned, that is, the working and index files. The working file contains the data that are needed in an actual application. Judicious organization of this file guarantees fast access to the data. The organization is determined by the type of information, that is, whether there are numerical, alphanumerical, topological, or graphical data. The searching algorithm is also responsible for the data representation.

Access to other data or files is organized via the index file. For processing databank information, additional software will be necessary.

Laboratory Information and Management Systems (LIMS)

Among analytical databases, there are also systems for organizing laboratory work, for exchanging information, and for communicating within a company. They are termed *laboratory information and management systems* (*LIMS*).

Typical performance characteristics of a LIMS are as follows:

- Sample identification
- Design of analytical procedures
- Compilation of analytical reports
- Release of analytical results
- Acquisition of raw data and data reduction
- Archiving of analytical results
- Database functions for chemicals, reference materials, suppliers, specifications, personal, and bibliographies.

Organization of data in a LIMS can be carried out by different models of data bank theory:

- Entity relationship model
- Network data model
- Hierarchical data model
- Relational data model.

Today, the relational data model is the dominant one. In contrast to a telephone directory, this model enables related lists to be represented. It is based on combinations of keys. The individual list can be kept short and the data structure can be extended at any time if required. Thus, in the development of a LIMS, not all options of the applicant have to be known in advance.

Table 7.9 key serves here the sample identification. The origin and matrix of the sample form the attributes of the relation. Every row represents a realization of the relation.

Manipulation of entities in the database is performed either by relation-oriented operations, such as projection, connection, and

Table 7.9 Example relation for characterization of an analytical sample.

Sample identification	Origin	Matrix
P1	Final control	Alloy
P2	Plant 1	Steel
P3	Plant 2	Fly ash
P4	Supplier 4	Ore
P5	Customer 007	Sewage

selection, or by set-oriented operations, that is, union, intersection, and negation (cf. Section 8.3).

7.2 Library Search

Spectra and chemical structure searches are based on distance and similarity measures as introduced in Section 5.2. Different strategies are known: sequential search, search based on inverted lists, and hierarchical search trees. The strategies are explained for search of spectra.

Search Strategies

Sequential Search

This kind of search is based on comparing the measured spectrum with candidate spectra of the library, bit by bit. Sequential search is only useful if a small data set is to be treated or if it is obligatory to retrieve every individual data set. A more efficient way is to sort the entities in a database by deriving appropriate keys.

Inverted Lists

With this method, selected data keys are defined and the data are arranged in new files that contain the information on the individual data sets. Consider an inverted list for a spectral library in the IR range (Figure 7.6). The key consists here of the numbered spectral features, that is, in this example, the wavenumbers representing absorption maxima.

Every feature appears in the list of keys together with the identity numbers (IDs) of all spectra that contain the actual feature. After collection of all features of the unknown spectrum, a rather short file can be generated on the basis of the keys that consist of all the candidate spectra.

Problems may arise if the lengths of the inverted lists differ. This may be because certain wavenumbers are more typical than others or certain chemical structures appear more often, for example, –C–C– is more frequent than –C=C–. The solution to this problem is the application of Hash coding algorithms: the key is coded by a random number that is then stored in a random access file.

Hierarchical Search Trees

Hierarchical arrangements of spectra or chemical structures are based on grouping of data by means of some similarity measure.

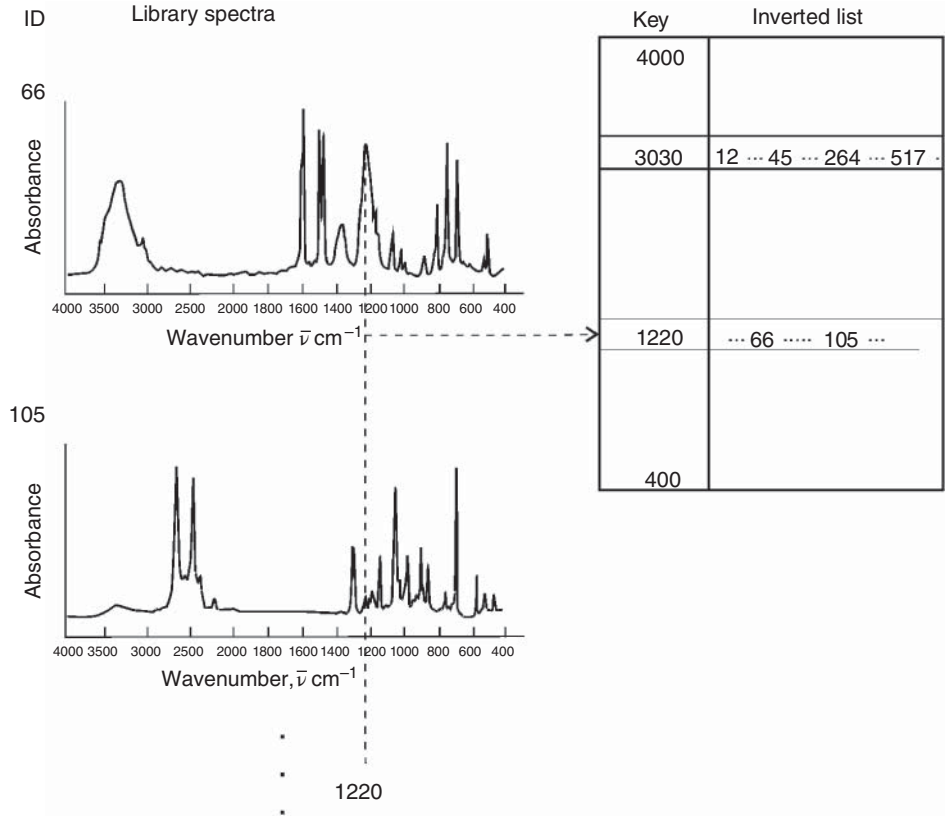


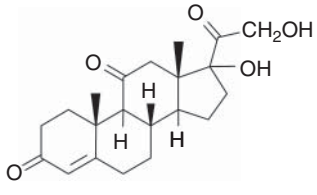
Figure 7.6 Inverted list for an IR spectral library. *ID* Identity number of spectrum.

The fundamentals have been introduced with cluster analysis in Section 5.2. The main problem in library search is in deciding the metric to be chosen in order to describe the similarity of spectra or chemical structures. In addition, clustering of large amounts of data may still be limited by the computer resources available at present.

Similarity Measures for Spectra

Comparison of a measured spectrum with a candidate library spectrum is feasible by these different principles:

- Correlation of spectra
- Similarity and distance measures
- Logical operations.



Cortisone

Table 7.10 Hit list for comparison of UV spectra of corticoids in methanol/water (70/30%) eluent based on the correlation coefficient (Eq. (5.12) in Section 5.1).

Compound	Correlation coefficient
Cortisone	0.999
Dexamethasone	0.965
Betamethasone	0.962
Prednisolone	0.913

Correlation Measures

Comparison of full spectra can be achieved by applying correlation or similarity measures. In the case of correlation, the coefficient of correlation between the spectra to be compared is computed (cf. Eq. (5.12)). Ranking the comparisons by the size of correlation coefficients, Table 7.10 provides a *hit list* that describes the quality of comparison between the unknown and the candidate library spectrum. In principle, the spectrum with the highest correlation coefficient is the sought-for spectrum. In order to ensure that a certain degree of similarity is reached for the top spectrum of the hit list, a threshold for assigning the library spectrum should be specified.

Typically, the correlation coefficient is used for comparison of UV spectra, for example, as is common in HPLC with diode array detection.

Similarity and Dissimilarity Measures

Comparisons of spectra with similarity or distance measures are based on the same definitions as given in Eqs. (5.87) – (5.91).

In the case of full spectra or of other analytical signal curves, both the Euclidean distance and the Manhattan distance are used as similarity measures. In the case of the Manhattan distance, the differences between the unknown and the library spectrum are summed. As a result of comparison, a hit list ranked according to distances or similarities of spectra is again obtained.

Grouping and Feature Selection

One possibility to speed up the search is preliminary sorting of the data sets. Here, the methods of unsupervised pattern recognition are used, for example, principal component and factor analysis, cluster analysis, or neural networks (cf. Sections 5.2 and 8.2). The unknown spectrum is then compared with every class separately.

To improve spectral comparisons, the selection of features will very often be necessary. For example, in mass spectrometry, the

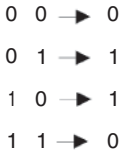


Figure 7.7 Connection of bits by exclusive OR (XOR).

original spectra are scarcely used for spectral retrievals. Instead, a collection of features is derived, such as the modulo-14-spectrum.

Logical Operations

Comparison of spectra is also possible by using logical connectives (cf. truth Table 1.2). A prerequisite for logical comparison is the conversion of spectra into a bit format. Bitwise conversion can be performed with complete spectra. More frequently, however, the bit vectors are formed from features derived from the raw spectral data. The logical operations can also be considered as distances of the derived vectors in bit space.

In the simplest case, the unknown spectrum is compared with the candidate library spectrum by an AND connective (cf. Table 1.2).

A typical dissimilarity measure is the so-called Hamming distance based on the exclusive OR (cf. Figure 7.7) calculated as follows:

$$\text{Hamming distance} = \sum_{i=1}^{p} \text{XOR}(y_i^{(\text{A})}, y_i^{(\text{B})}) \tag{7.3}$$

where $y_i^{(\text{A})}$ is the bit vector for the unknown spectrum at point i, $y_i^{(\text{B})}$ the bit vector of the candidate library spectrum, and p the number of points (wavelengths).

The distance calculation is shown in Figure 7.8. For identical spectra, the Hamming distance would be zero.

A	1	0	0	1	1	0
B	0	1	0	1	1	0
XOR	1	1	0	0	0	0

A combination of different logical operations can be found in mass spectrometry with the following dissimilarity measure S′:

$$S' = 2 + \sum_{i=1}^{p} \text{XOR}(y_i^{(\text{A})}, y_i^{(\text{B})}) - 2\sum_{i=1}^{p} \text{AND}(y_i^{(\text{A})}, y_i^{(\text{B})}) \tag{7.4}$$

In the examples discussed, the comparison is based on bit *vectors.* Another type of logical operation can be based on

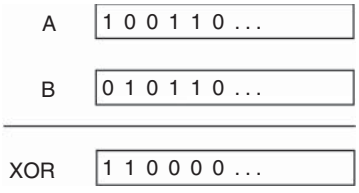


Figure 7.8 Comparison of an unknown spectrum (A) with a candidate library spectrum (B) by the exclusive OR connective (XOR).

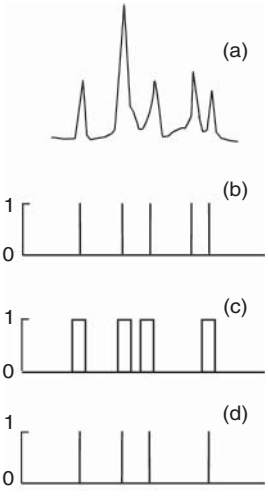


Figure 7.9 Set-oriented comparison of two spectra: (a) Unknown original spectrum. (b) Signal positions for the unknown spectrum. (c) Library spectrum with intervals for signal positions. (d) Comparison of (b) and (c) by intersection of the two sets (AND-connective).

set-oriented comparisons, which is necessary if the length of data sets differs from spectrum to spectrum. Typical examples are the peak list in IR spectrometry or capillary chromatograms. Figure 7.9 demonstrates the set-oriented comparison of two spectra.

The peak list of the library spectrum is assumed as errorless or crisp and the peak positions of the measured unknown spectrum are characterized by intervals. Comparison is performed usually by the AND connective, that is, by the intersection of both sets.

In the case of fuzzy intervals, the spectra have to be compared on the basis of fuzzy set theory (cf. Section 8.3).

Similarity Measures for Chemical Structures

If the chemical structures are encoded in *fragmentation codes,* a preselection of substructures is feasible. The comparison between the coded vectors of the unknown and library structures is possible by means of AND connection.

A comparison of structures *atom by atom* is based on the connection tables. Consider the classical example presented by E. Meyer in 1970 as given in Figure 7.10.

Example 7.4 *Substructure Search*

The task is to check whether the specified substructure is contained in the given molecule. Starting atom-wise comparison at nitrogen atom no. 1, in both structures the attached atom is carbon. The next atom, however, is oxygen in the molecule and carbon in the substructure. At this point, the search is stopped and backtracked. A new trial starting at the nitrogen atom no. 8 in the molecule will match the substructure step by step.

Molecule:

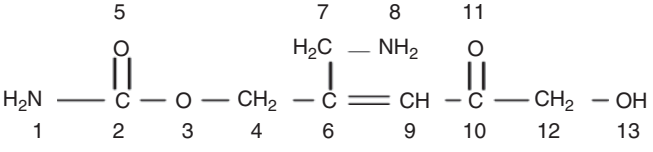
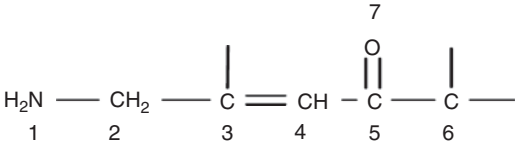


Figure 7.10 Example of substructure search based on connection tables of the chemical structures (for representation of the connection table see Table 7.8).

Connection table:

Atom no.	Symbol	Atom1	Bond	Atom2	Bond	Atom3	Bond
1	N	2	1	—	—	—	—
2	C	1	1	3	1	5	2
3	O	2	1	4	1	—	—
4	C	3	1	6	1	—	—
5	O	2	2	—	—	—	—
6	C	4	1	7	1	9	2
7	C	6	1	8	1	—	—
8	N	7	1	—	—	—	—
9	C	6	2	10	1	—	—
10	C	9	1	11	2	12	1
11	O	10	2	—	—	—	—
12	C	10	1	13	1	—	—
13	O	12	1	—	—	—	—

Sought-for substructure:



Atom no.	Symbol	Atom1	Bond	Atom2	Bond	Atom3	Bond
1	N	2	1	—	—	—	—
2	C	1	1	3	1	—	—
3	C	2	1	4	2	—	—
4	C	3	2	5	1	—	—
5	C	4	1	6	1	7	2
6	C	5	1	—	—	—	—
7	O	5	2	—	—	—	—

Match						
Substructure	1	2	3	4	5	6
Molecule structure	7	8	9	10	11	12

7.3 Simulation of Spectra

When there is no spectrum in the library that can be used to elucidate a chemical structure, interpretative methods are needed. The methods of pattern recognition and of artificial intelligence must then be used. As a result, different chemical structures will be obtained as candidates for the unknown molecule. To verify an assumed structure, simulation of spectra becomes important. In a final step, the simulated spectrum could be compared with that measured.

High-performance methods for routine simulations of IR and mass spectra are not yet available. In IR spectroscopy, the best simulations are obtained on the basis of quantum-chemical approaches.

Simulations are more successful for NMR spectroscopy (Figure 7.11). Simulation of chemical shifts, δ, are based on increments that are derived from investigations of a set of well-characterized compounds by means of multiple linear regression

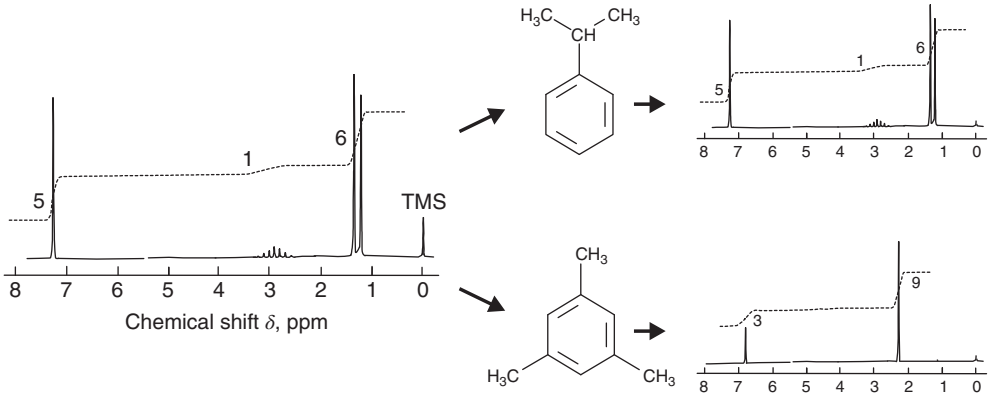


Figure 7.11 Simulation of the ^{1}H NMR spectrum for cumene (above right) and 1,3,5-trimethylbenzene (below right) to enable verification of the measured spectrum (shown left) as being that of cumene.

analysis (cf. Section 6.2):

$$\delta = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n \tag{7.5}$$

x_i	numerical parameter describing the environment of the structure
b_i	regression coefficient
n	number of descriptors.

The increments used here in the NMR simulation represent in a more general sense *molecular descriptors* that can be defined as follows [5]:

> The molecular descriptor is the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment.

With this definition, we can consider the structure representations of molecules in Section 7.2 as topological descriptors. At present, more than 5000 molecular descriptors can be computed [5]. They can be categorized by their data types (Table 7.11) or by their dimensionality (Table 7.12).

In the context of analytical databases, the simulation or prediction of spectra from molecular descriptors is of interest. Of course, the descriptors can be correlated with any kind of properties or activities a molecule might have. This is then called *QSPR*

Table 7.11 Molecular descriptors categorized by data type.

Data type	Example
Logical variable	Existence of functional groups
Integer	Number of heteroatoms
Natural	Molecular mass
Vector	Dipole moment
Tensor (3×3)	Electric polarizability
Scalar field	Electrostatic field
Vector field	Force

Table 7.12 Molecular descriptors categorized by dimensionality of data.

Data type	Examples
0D	Number of carbon atoms, double bonds, molecular mass
1D	Alcohols (aromatic/aliphatic, secondary, tertiary)
2D	Topological descriptors (adjacency matrix, *BE* matrix)
3D	Surface properties (area of potential)
4D	3D coordinates plus conformers

(Quantitative Structure – Property Relationship) or *QSAR* (Quantitative Structure – Activity Relationship).

General Reading

1. Gasteiger, J. and Engel, T. (eds) (2003) *Chemoinformatics – a Textbook*, Wiley-VCH Verlag GmbH, Weinheim.
2. Gasteiger, J. (ed.) (2003) *Handbook of Chemoinformatics: From Data to Knowledge*, Wiley-VCH Verlag GmbH, Weinheim.
3. Zupan, J. (1989) *Algorithm for Chemists*, John Wiley & Sons, Ltd., Chichester.
4. Kerber, A., Laue, R., Meringer, M., Rücker, C., and Schymanski, E. (2014) *Mathematical Chemistry and Chemoinformatics*, De Gruyter, Berlin, Boston, MA.
5. Todeschini, R. and Consonni, V. (2009) *Molecular Descriptors for Chemoinformatics*, Wiley-VCH Verlag GmbH, Weinheim.

Questions and Problems

1. Draw the chemical structure of the molecule given by the following connection table:

Node no.	Atom	Connected to	Bond
1	C	—	—
2	C	1	1
3	C	2	1
4	C	3	2
5	N	3	1

2. Derive the redundant and nonredundant connection table for the molecule acetaldehyde with atom numbering as given in Example 7.3.

3. Which substructures describe the two fragmentations based on the HOSE codes = OCC(,*C*C,//) and *C*CC ⇒ (*C,*C,C//) ?

4. Compute the SMILES code for caffeine, for example with the program ChemSpider of the Royal Society of Chemistry.

5. Build the *BE* matrices ***B*** and ***E*** for the chemical reaction of formaldehyde with hydrocyanic acid to give cyanohydrin and calculate the reaction matrix ***R*** as given in Example 7.3.

6. How does one derive a peak list from a full spectrum?

7. Which spectral features can be used as keys in sorting spectra in an inverted list?

8. Mention common similarity measures in retrieving spectra and chemical structures.

9. Which methods can be used to group spectra and structures?

10. Compare two spectra by the dissimilarity measure given in (Eq. (7.2)). What is the value of S'?

8
Knowledge Processing and Soft Computing

Learning objectives

- To introduce representation and processing of knowledge in the computer for developing analytical expert systems
- To understand the operation of artificial neural networks and their application to pattern recognition and modeling of analytical data
- To learn about the theory of fuzzy sets for handling vague and incomplete data and knowledge
- To demonstrate the genetic algorithms for solving complex optimization problems.

8.1
Artificial Intelligence and Expert Systems

Introduction to Artificial Intelligence

A definition of artificial intelligence (AI) is as follows [1]:

> *Artificial intelligence is the study of mental faculties through the use of computational methods.*

Particularly, "intelligent" human faculties, such as abstracting or reasoning, are not generally considered, but rather quite common faculties, such as vision and understanding. Another property of the faculties of interest is that humans can easily cope with them in contrast to tasks a computer can easily perform, for example, multiplication of 10-digit numbers (cf. Section 1.3). Modules of AI research are given in Figure 8.1.

The beginning of AI research is ascribed to the Dartmouth Conference in 1956, organized by the scientists John McCarthy and Marvin Minsky. Intensive AI developments were connected in the

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

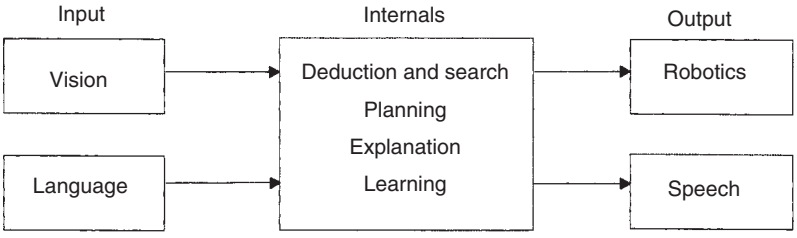


Figure 8.1 Modules of artificial intelligence (AI) research in analogy to mental faculties of human beings.

1980s to a Japanese research project on the fifth computer generation based on the symbolic programming language PROLOG (recursive programming in logic).

The methods of AI are mainly knowledge-oriented, in contrast to algorithmic data processing considered in the previous chapters. Acquisition and processing of knowledge are feasible in two ways:

- First, *symbolically* by means of programming languages, such as list processing language (LISP) or PROLOG. There, the knowledge is represented *explicitly* in the form of facts and rules. The problem is described, rather than solution strategies being implemented as is usual with conventional programming languages.
- Second, *neural networks* that may store knowledge *implicitly* and find appropriate answers after presentation of training patterns or structures (cf. Section 8.2) are exploited. Neural networks are built at present by using conventional programming languages. In the future, however, parallel operating computers or transputers will be applied.

Symbolic programming and neural nets are increasingly developing as complementary AI techniques. Neural networks are especially suited for *pattern recognition.* For example, images and language can be interpreted similarly to multidimensional data. Symbolic programming is mainly used for inferring by logical or approximate reasoning.

Application Areas in Analytics

The most important applications of the methods of AI in analytics are as follows:

- Expert systems
- Intelligent analyzers and robot systems (cf. Section 1.3).

The development of *expert systems* is of interest, for example, for interpretation of spectra. There exist expert systems for

both molecular spectroscopy (mass spectrometry, nuclear magnetic resonance (NMR), and infrared (IR) spectroscopy) and atomic spectroscopy, for example, X-ray fluorescence analysis or atomic emission spectroscopy. In addition, in chromatography, the selection and optimization of a method can be based on an expert system. Also, the selection of analytical methods for a given area is known, for example, for analyses in a steel laboratory.

Symbolic Knowledge Processing

Knowledge Representation

For internal representation of knowledge in the computer, the knowledge has to be preprocessed. A sentence is not stored in its original form, but, for example, on the basis of *predicate logic.* Consider the following sentence:

Copper has an atomic line at 324.8 nm

In predicate calculus, we write:

$$\text{atomic line(copper, 324.8 nm)} \tag{8.1}$$

The predicate is here the term "atomic line." The predicate is characterized here by the a*rguments* for the element "copper" and the wavelength "324.8 nm." In general, a predicate can be represented by

$$\text{predicate(Attribute1, Attribute2, …)} \tag{8.2}$$

The arguments are instantiated by the different terms in the following way:

- *Constant symbols,* such as "copper," "324.8 nm"
- *Variables,* for example, "element," "wavelength"
- *Functions* of several predicates.

Complementary forms of knowledge representation are based on *semantic nets* and *frames* (Figure 8.2). Often, they represent just another form to input knowledge. The internal representation is usually based on the predicate calculus. The latter can also be interpreted as a relation of objects:

$$\text{Relation(Object1, Object2, …)} \tag{8.3}$$

In our example, a relationship between the element "copper" and its absorption wavelength at "324.8 nm" is defined and denoted as "atomic line."

Encoding of knowledge by predicate logic or in another computer internal representation is the task of a knowledge engineer, who as a rule is a nonexpert for the given knowledge domain.

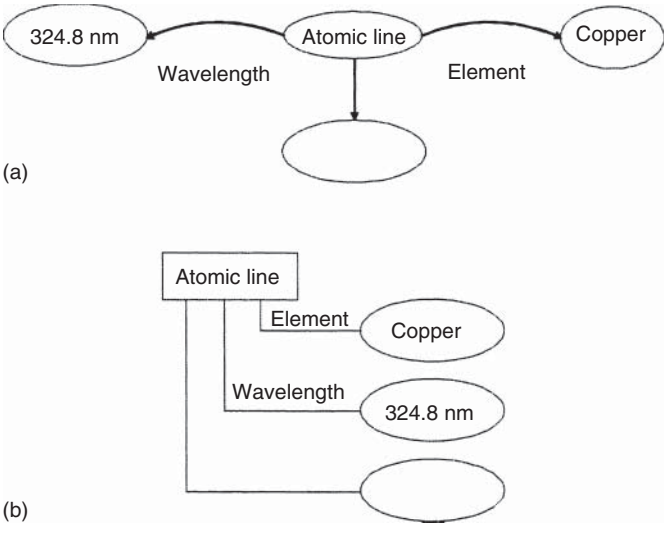


Figure 8.2 Representation of knowledge in the form of semantic nets (a) and frames (b).

Inferences

Inferences consist of *premises* (conditional part) and *consequences* (inference part).

Logical Connectives For reasoning and inferring on the basis of the internal knowledge presentation, the relationships or relations have to be described. This is feasible with logical connectives, as we know already from Section 1.1 (Table 1.2). Important connectives for compound propositions of p and q are as follows:

Conjunction (p AND q)
Alternative (p OR q)
Implication (IF p THEN q)
Negation (NOT p)
Equivalence (p IF AND ONLY IF q).

The connectives already represent a kind of a rule. A *rule* is a conclusion that is true if other conclusions and facts are true. A *fact* is the description of a true proposition, such as that in the predicate of Eq. (8.1).

The initially known facts are termed *axioms*.

Deduction By means of logical connectives, correct inferences can be derived in the sense of deductive reasoning. In general, for deduction, true axioms (postulates) are given and the conclusions drawn are again true. This is denoted a *legal inference*.

Example 8.1 *Deduction*

"An element is a metal,

IF the surface shines
AND the aggregate state is solid
AND the electrical conductivity is high.
:"

Abduction With this kind of inference, explanations are generated. Example 8.2 demonstrates this.

The best known inference rule is the *modus ponens:*

Given:
IF $p = A$ THEN $q = B$
$p = A$
Inference:
$q = B$

Example 8.2 *Abduction*

Given: "X has a high conductivity"
and "All metals have a high conductivity"
Inference: "X is a metal."

Abduction is not a legal inference. If in Example 8.2 X were, for example, a conducting polymer, then the inference will be false. Apart from this, abductive inferences are very useful. Think of a medical or instrumental diagnosis. If certain symptoms are observed, then an assumed disease or instrument state becomes plausible.

Induction Learning is usually carried out by the third kind of inference, that is, induction. The following example explains this.

Example 8.3 *Induction*

Given: "Metallic copper is solid"
and "Metallic iron is solid."
Inference: "All metals are solid."

Induction is useful, but as for abduction, does not give a legal inference. The conclusion drawn in Example 8.3 that "all metals are solid" is inadmissible, for example, for the metal mercury.

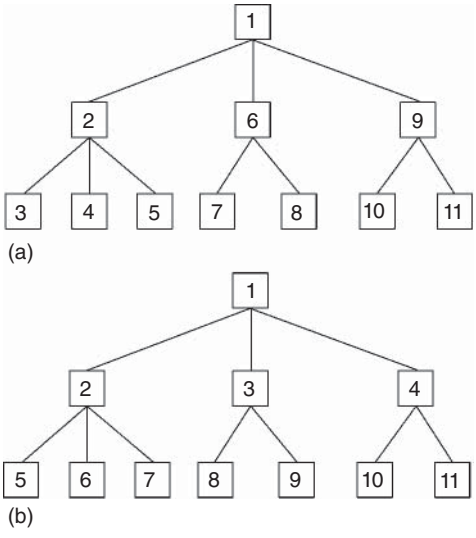


Figure 8.3 Strategies for in-depth search (a) and in-breadth search (b).

Search Strategies

In-depth search is also known as *backtracking*.

For systematic search in a knowledge base, two strategies are feasible: first, in-depth searching and second, in-breadth searching (Figure 8.3). In the case of in-depth search, new states (facts, rules) are positioned on the top of the logical chain, that is, the latest retrieved states are eliminated first (last-in/first-out) if they do not satisfy. In contrast, in the in-breadth search, the additional states are linked to the end of the chain. The initially tested rules are eliminated first (first-in/first-out).

Symbolic Programming

Algorithmic programming languages, such as BASIC, Pascal, or C, do not support logical programming. Inferences would have to be performed by means of an additional interpreter that is built externally. This is feasible in principle, but quite laborious.

LISP was developed in the United States by *John McCarthy*.

For logical inferences, appropriate symbolic languages have been developed. The mother tongue of AI is *LISP* [2]. Dialects of this language are Franz Lisp, Common Lisp, or MacLisp. The language demands quite large computer memory and a powerful computer, so that real versions are better used on a workstation rather than a PC. Facts and rules are arranged in lists (cf. Figure 8.4). The head of a list characterizes the predicate. The subsequent positions represent the arguments according to Eq. (8.2).

Figure 8.4 Program to find an element in a list of elements by means of list processing language (*LISP*) and a recursive programming in logic (*PROLOG*) procedure.

LISP	PROLOG
(member Element List)	member(Element, [Element| _]). member(Element, [_| Restlist]):- member(Element, Restlist).
Example: (member "Fe" (Co Cu Fe)) (Fe)	*Example*: (member("Fe" ("Co" "Cu" "Fe")) YES

In LISP, all inference mechanisms are implemented by the user. This guarantees that no constraints exist with respect to the type of inference. A disadvantage, however, is the relatively high workload to program the inference schemes.

A symbolic language where a particular inference mechanism is already predesigned is PROLOG as developed in the 1970s by Alain Colmerauer. In the development of that language, the aim was to describe the problem to the computer and not to formulate individual steps of the solution. Even if this does not completely work, PROLOG is at present one of the highest level symbolic programming languages. PROLOG is based on Horn clauses that form a subset of the formal system of predicate logic. A fact or rule in PROLOG corresponds, therefore, to the formal structure of Eq. (8.1). The inference mechanism is based on the *backtracking* algorithm.

Horn clauses provide *only one* consequence, for example, IF $p = A$, THEN $q = B$, but not IF $p = A$, THEN $q = B$ **AND** $r = C$.

The result of processing rules in a symbolic programming language is not a numerical value, but a truth value, that is, "true" or "false." In the case of true propositions, additional information can be output (cf. Figure 8.4).

Expert Systems

The most important application of methods of knowledge processing on the basis of logical inferences are expert systems:

Expert systems are computer programs that help to solve problems at an expert level.

They contain knowledge about a particular, limited, but originating in the real-world problem, the solution of which requires expert knowledge.

Aims

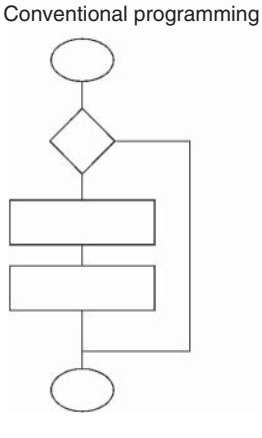
The aims for developing an expert system are manifold:

- To *copy knowledge of an expert,* in order to make it accessible to less qualified coworkers, to objectivate it, or to make it available independent of time and conditions.
- To *advise an expert* for developing or confirming a diagnosis. In this context, rare cases with unusual combinations of facts and findings are of special interest. Routine cases are usually better managed by the expert.
- *Extension* of the activity of an expert and as a result of his knowledge.
- *Training* and *education* to learn about a particular expert domain.

The performance of existing expert systems for advising or replacing the expert is rather different. Without doubt, there is benefit from developing an expert system with respect to structuring the knowledge domain considered and the complementary collection of important material. The training of less qualified personal is also very useful.

Development Tools

The aforementioned symbolic programming languages *LISP* and *PROLOG* are important tools for developing an expert system. As a rule, however, their use is reserved to the work of a science engineer. The analyst develops his expert system by means of the so-called *shells.* These systems provide predefined inference mechanisms and contain an "empty" knowledge base. The latter is to be filled in with the domain knowledge of the given field.



Structure of Expert Systems

In conventional programming, the different operations are processed in the sense of a spreadsheet. In an expert system, there is a clear division between the knowledge base, which contains the facts and rules, and the inference engine. Facts and rules can be easily added to or removed from the knowledge base.

The structure, in principle, of an expert system is given in Figure 8.5. As a rule, knowledge acquisition is carried out in

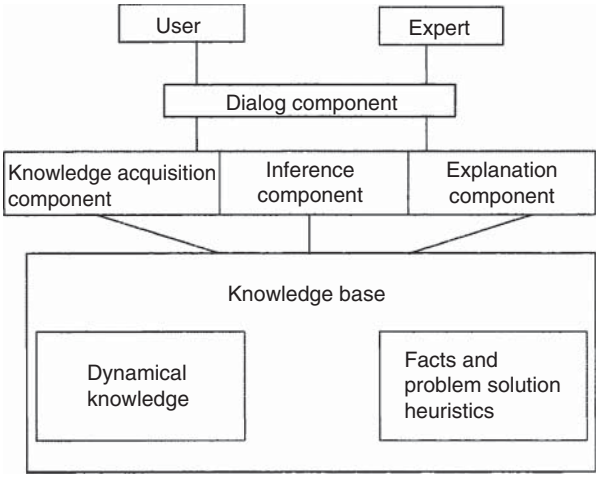
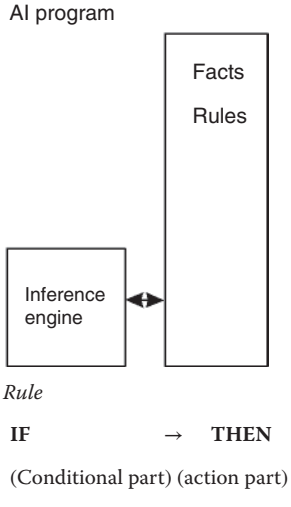


Figure 8.5 Structure of an expert system.

the dialog between the domain expert, here the analyst, and a knowledge engineer. If the knowledge engineer works in too much isolation, the problem emerges that the knowledge base might become too detailed. Redundant information, available by the in-depth knowledge of the analyst, is then maintained. Since, on the other hand, the analyst will not be the only user of the tools of AI in the future, methods for machine learning of symbolic knowledge will become increasingly important.

The *inference engine* operation is usually rule-based. Such a system exploits IF–THEN rules (implications). Production systems are termed *systems*, where a rule interpreter applies the given production rules on a working memory. A PROLOG interpreter is a typical production system.

Apart from the knowledge acquisition component, the expert system should also contain an explanation component.



Expert Systems in Analytical Chemistry

The very first expert system – *Dendral* – was developed for interpretation of organic mass spectra [3]. A dialect of LISP was used as the language: Inter-LISP. Although this system could not be developed as a routinely used expert system, it provided important experience in developing expert systems in analytics. Perhaps the most intensive studies for developing expert systems in analytics were done in the framework of the ESPRIT project of the EU [5]. Developments from this project are summarized together with other expert systems in Table 8.1.

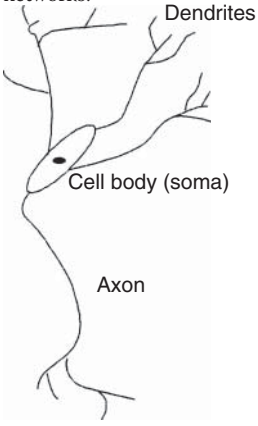
Table 8.1 Examples of expert systems in analytics.

Area	Aim	Language	References
MS	Interpretation of mass spectra	LISP	[3]
HPLC	Method selection/optimization	PROLOG, shells	[4, 5]
GC	Selection of separation system/Operating conditions	LISP	[6]
XRA	Spectra interpretation (energy dispersive)	Pascal	[7]
	Spectra interpretation (wavelength dispersive)	PROLOG	[8]
Titration	Karl Fischer titration	INSIGHT 2-Shell	[9]
Steel analysis	Sample preparation/analysis management	Kappa-PC-Shell	[10]

8.2 Neural Networks

Fundamentals

Synonymous terms for data processing with neural nets are neural computing, parallel distributed processing (PDP), or artificial neural networks.



Application of artificial neural networks for data and knowledge processing is characterized by analogy to a biological neuron. If the firing frequency of a neuron (about 1 kHz) is compared to that of a computer (greater than 100 MHz), then for the neuron, this frequency is rather low. The high performance of biological systems therefore is most probably reasoned by the 1000-fold interconnections of about 10 billions of cells in the brain.

Based on this knowledge, one tries to simulate the biologic neuron by means of an *artificial neuron*. In the biological neuron, the input signal arriving through the axon ends in the synapse. There the information is transformed and sent by passing the dendrites to the next neuron (Figure 8.6a). This signal transfer is simulated in the artificial neuron by multiplication of the input signal, x, with the synaptic weight, w, to derive the output signal y (Figure 8.6b).

What remains in comparison with the biological neuron is a much simplified neuron based on a simple product formation. Good performance of artificial neural networks is therefore to be attributed to a high degree of interconnections rather from its analogy to a biological neuron.

Neurons, Weights, and Transfer Functions

The neural networks considered here consist of an input layer, which receives the input signals and in the simplest case is connected to a second layer, the output layer (two-layer network). Between the input and output layers, additional layers may be arranged. They are termed *hidden layers* (Figure 8.7).

Consider the operation of a single neuron j (cf. Figure 8.8). It receives from n other neurons the input signals, x_i, aggregates

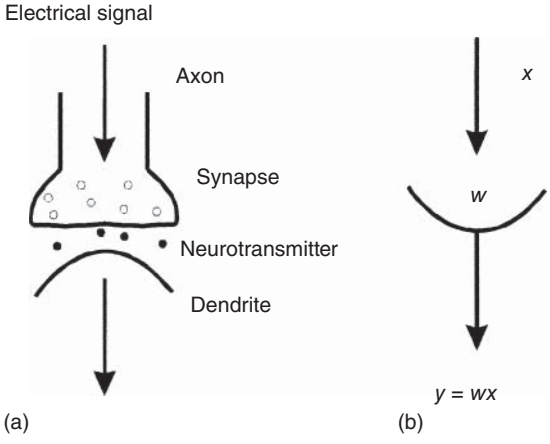


Figure 8.6 Analogy between a biological (a) and an artificial neuron (b).

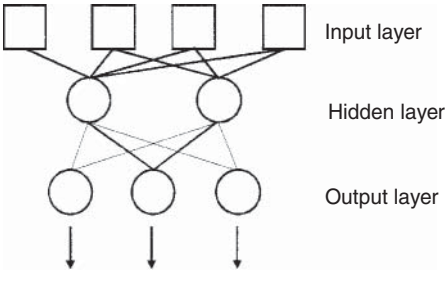


Figure 8.7 Structure of an artificial neural network.

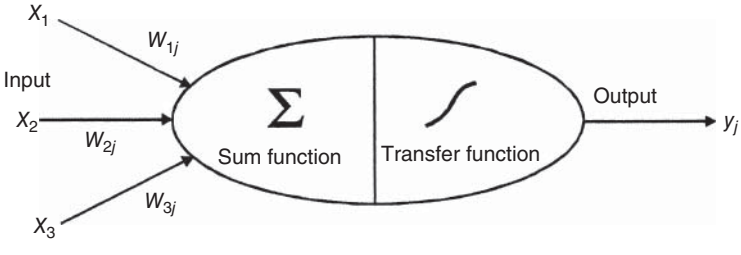


Figure 8.8 Operation of a single neuron.

them by using the weights of the synapses, and passes the result after suitable *transformation* as the output signal y_j.

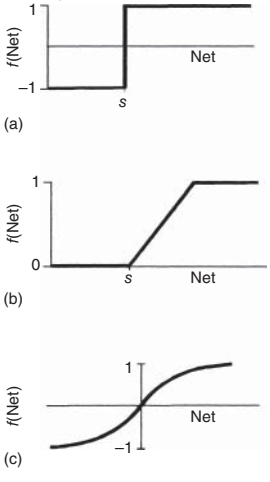
Typically, for aggregation of the input signal, their summation is applied. If the result of the aggregation in neuron j is denoted as

Table 8.2 Transfer functions for neural nets.

Transfer function	0. Derivative	1. Derivative
Threshold logic (binary)	$f(Net) = \begin{cases} 1 \text{ for } Net \geq s \\ 0 \text{ for } Net < s \end{cases}$	—
Threshold logic (bipolar)	$f(Net) = \begin{cases} 1 \text{ for } Net \geq s \\ -1 \text{ for } Net < s \end{cases}$	—
Linear transfer function	$f(Net) = c \cdot Net$	$f'(Net) = c > 0$
Linear threshold function	$f(Net) = \begin{cases} 1 \text{ for } c \cdot Net \geq s \\ 0 \text{ for } c \cdot Net < s \\ c \cdot Net \text{ else} \end{cases}$	$f'(Net) = c > 0$
Sigmoid function	$f(Net) = \frac{1}{1 + e^{-c \cdot Net}}$	$f'(Net) = cf(Net)[1 - f(Net)] > 0$
Hyperbolic tangent	$f(Net) = \frac{e^{cNet} - c^{-cNet}}{e^{cNet} + e^{-c \cdot Net}}$	$f'(Net) = c[1 - f(Net)^2] > 0$

s Threshold and c constant

Transfer functions for threshold logic (a), linear threshold function (b), and sigmoid function (c)



the net signal, Net_j, then we obtain

$$Net_j = \sum_{i=1}^{n} x_i w_{ij} \tag{8.4}$$

Other aggregation operators are possible, for example, formation of the minimum or maximum over all n signals x_i. Before the aggregated signal leaves the neuron, a transformation is performed by means of a transfer function f to get the output signal y_j:

$$y_j = f(Net_j) \tag{8.5}$$

Important transfer functions are given in Table 8.2. For learning algorithms, the derivatives of the transfer functions are needed (also given in Table 8.2). Apart from linear transfer functions, all transformations provide a nonlinear transformation of the aggregated signals. The use of neural networks is therefore especially interesting for solving nonlinear problems.

A Simple Neural Net: BAM

The bidirectional associative memory (BAM) is used here to explain the operation of a neural network in more detail.

The BAM consists of an input layer, $\boldsymbol{x}$, and an output layer, $\boldsymbol{y}$, as well as of the layer with weights, $\boldsymbol{W}$. Since the BAM passes and transforms signals in the input and output layers, the neurons are simultaneously characterized by circles and squares.

To learn m different associations of patterns ($\boldsymbol{x}_i$, $\boldsymbol{y}_j$), *correlation learning* is used. The synaptic weights w_{ij} are obtained by

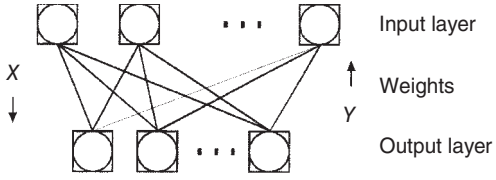


Figure 8.9 Bidirectional associative memory (BAM) consisting of n input and p output neurons.

(Figure 8.9):

$$w_{ij} = x_i y_j \tag{8.6}$$

The weights for all m associations are stored in a correlation matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = \boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{y}_1 + \boldsymbol{x}_2^{\mathrm{T}} \boldsymbol{y}_2 + \cdots + \boldsymbol{x}_m^{\mathrm{T}} \boldsymbol{y}_m \tag{8.7}$$

If an unknown data vector, $\boldsymbol{x}$, is to be recognized, then it is presented at the input of the net and a first $\boldsymbol{y}$ vector is estimated:

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{W} \tag{8.8}$$

The output vector, $\boldsymbol{y}$, is passed in backward direction by

$$\boldsymbol{x} = \boldsymbol{y}\boldsymbol{W}^{\mathrm{T}} \tag{8.9}$$

This procedure is repeated iteratively as long as a stable state of the network is reached:

$$\begin{aligned}
&\boldsymbol{x} \rightarrow \boldsymbol{W} \rightarrow \boldsymbol{y}^{(0)} \\
&\boldsymbol{x}^{(1)} \leftarrow \boldsymbol{W}^{\mathrm{T}} \leftarrow \boldsymbol{y}^{(0)} \\
&\boldsymbol{x}^{(1)} \rightarrow \boldsymbol{W} \rightarrow \boldsymbol{y}^{(1)} \\
&\boldsymbol{x}^{(2)} \leftarrow \boldsymbol{W}^{\mathrm{T}} \leftarrow \boldsymbol{y}^{(1)} \\
&\vdots \\
&\boldsymbol{x}^{(f)} \rightarrow \boldsymbol{W} \rightarrow \boldsymbol{y}^{(f)} \\
&\boldsymbol{x}^{(f)} \leftarrow \boldsymbol{W}^{\mathrm{T}} \leftarrow \boldsymbol{y}^{(f)}
\end{aligned} \tag{8.10}$$

The network takes on a resonant state. We should also mention the threshold function, which is applied at each computation of new $\boldsymbol{x}$ and $\boldsymbol{y}$ values. Consider the calculations in Example 8.4.

Example 8.4 *BAM*

For two bipolar associations $(\boldsymbol{x}_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2, \boldsymbol{y}_2)$, the weights of a BAM are to be trained:

$$\boldsymbol{x}_1 = \begin{pmatrix} 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix}, \quad \boldsymbol{y}_1 = \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} \tag{8.11}$$

$$\boldsymbol{x}_2 = \begin{pmatrix} 1 & 1 & 1 & -1 & -1 & -1 \end{pmatrix}, \boldsymbol{y}_2 = \begin{pmatrix} 1 & -1 & 1 & -1 \end{pmatrix} \tag{8.12}$$

After that, a new $\boldsymbol{x}$ vector is to be assigned consisting of the following values:

$$\boldsymbol{x} = \begin{pmatrix} -1 & 1 & 1 & -1 & -1 & -1 \end{pmatrix} \tag{8.13}$$

To code the two associations, we calculate the matrix of weights (correlation matrix) according to Eq. (8.7):

$$\boldsymbol{x}_1{}^{\mathrm{T}} \boldsymbol{y}_1 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$$

$$\boldsymbol{x}_2{}^{\mathrm{T}} \boldsymbol{y}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 \end{pmatrix}$$

$$\boldsymbol{W} = \boldsymbol{x}_1{}^{\mathrm{T}} \boldsymbol{y}_1 + \boldsymbol{x}_2{}^{\mathrm{T}} \boldsymbol{y}_2 = \begin{pmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 2 \\ -2 & 0 & 0 & 2 \end{pmatrix}$$

To assign the $\boldsymbol{y}$ vector associated with the given $\boldsymbol{x}$ vector in 8.13, multiplication by the weight matrix is carried out (cf. Eq. (8.8)):

$$\boldsymbol{y} = \boldsymbol{x}\boldsymbol{W} = (-1\ 1\ 1\ -1\ -1\ -1) \begin{bmatrix} 2 & 0 & 0 & -2 \\ 0 & -2 & 2 & 0 \\ 2 & 0 & 0 & -2 \\ -2 & 0 & 0 & 2 \\ 0 & 2 & -2 & 2 \\ -2 & 0 & 0 & 2 \end{bmatrix}$$

$$= (4 - 4\ 4 - 4)$$

Threshold formation according to the bipolar threshold logic leads to

$$\boldsymbol{y} = \begin{pmatrix} 1 & -1 & 1 & -1 \end{pmatrix}$$

The sought vector $\boldsymbol{y}$ is found in the first iteration cycle, so that no additional iterations are necessary. The found vector

corresponds to $\boldsymbol{y}_2$, that is, to association given in Eq. (8.12). Note that the net is error tolerant. The tested vector $\boldsymbol{x}$ is not identical to vector $\boldsymbol{x}_2$ of the found association, but differs in the first element of that vector.

Learning Paradigms

Neural networks can be used for supervised and unsupervised learning as we know from the statistical methods of pattern recognition (Chapter 5).

In the case of *unsupervised learning,* only input vectors are presented as known from factorial methods and cluster analysis (cf. Section 5.2). The objects are grouped on the basis of their features. Unknown objects can then be automatically recognized. We will learn about the Kohonen network as a typical neural net of that kind.

For *supervised learning,* the network receives additional information about the associated patterns or classifiers in the learning phase, which helps to adjust the weights. Unknown patterns and classes can subsequently be classified.

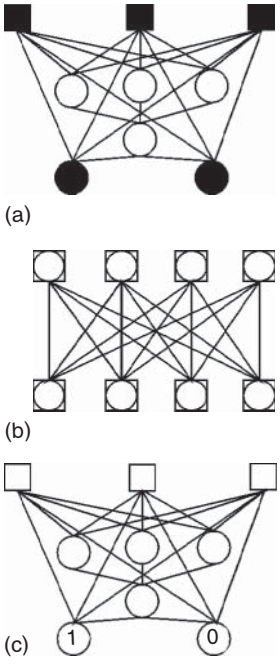
With respect to *associations of patterns,* we distinguish between auto- and heteroassociations. In the first case, the number of input and output neurons is equal. Heteroassociative networks have a different number of neurons in the input and output layers. Pattern associations can be used, for example, to learn about character or image combinations or spectra–structure relationships.

For *classification* of data vectors, the output of the net is coded by the class that corresponds with the input vector. At the end of the network training, the unknown data vectors, for example, spectra, can be assigned on the basis of the value of the output neurons. Without particular constraints, however, a pattern or class is always assigned. We already know of this disadvantage from the statistical methods of pattern recognition (Section 5.2).

Finally, neural networks can also be applied for the purpose of *modeling.* Model parameters are in a general sense the weights, $\boldsymbol{w}$, of the network. Consider the following linear model for a dependent variable y and three independent variables, x_i:

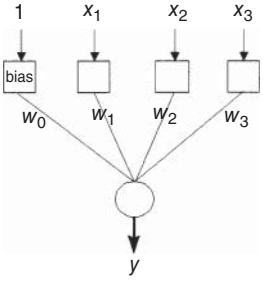
$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \tag{8.14}$$

This function is mapped in the simplest case by a net consisting of a single layer of weights, w_0, w_1, w_2, and w_3. The shift along the ordinate is accounted for by presentation of ones at one neuron, so that the intercept w_0 can be estimated (cf. Eq. (6.1)). This particular neuron is termed *bias neuron*.



Neuronal networks for learning of *heteroassociative* (a) and *autoassociative* (b) *patterns* as well as for *classification* (c)

Neural net for *parameter estimation* in Eq. (8.14)



Learning Laws

Learning in neural networks happens by associative or competitive learning laws. In this context, learning means the following:

- Linking new or elimination of existing synaptic connections.
- Change of the weights of existing connections by minimizing a predefined objective function.

Associative Learning Laws

The simplest learning law of that kind is the Hebb law. We have already used this rule in the BAM network (Eq. (8.6)). According to the Hebb law, a weight is strengthened if the corresponding neurons x_i and y_j are simultaneously activated:

$$w_{ij}(t+1) = w_{ij}(t) + \eta_t x_i y_j \tag{8.15}$$

$$\Delta w_{ij} = \eta_t x_i y_j \tag{8.16}$$

Time t denotes *cycles* here, since the network training is carried out in discrete iterations.

where η_t is the learning coefficient as a function of time t, for example,

$$\eta_t = 0.1\left(1 - \frac{t}{5000}\right)$$

Delta Rule This rule changes the weights in relation to a target pattern $\boldsymbol{t}$, which is presented at the input (t_i) or output (t_j) for comparison:

$$\Delta w_{ij} = \eta_t(t_i - x_i)y_j \tag{8.17}$$

$$\Delta w_{ij} = \eta_t x_i(t_j - y_j) \tag{8.18}$$

A direct application of Hebb's law is given in simple input–output networks. We will learn about a more general application in the backpropagation network, in the form of the generalized delta rule.

Competitive Learning Laws

In the case of competitive learning, the distance between the input vector, $\boldsymbol{x}$, is compared to the weight vector, $\boldsymbol{w}$, by using an appropriate distance measure. Usually, the Euclidian distance is applied (cf. Eq. (5.87)). In detail, the following steps are followed:

- Find the winning neuron j. It is that neuron that possesses the smallest distance to the input vector among all neurons $i = 1, n$:

$$\|\boldsymbol{w}_j(t) - \boldsymbol{x}(t)\| = \min_i \|\boldsymbol{w}_i(t) - \boldsymbol{x}(t)\| \tag{8.19}$$

  where $\| \, . \, \|$ is the Euclidian vector norm of $\boldsymbol{x}$.
- Improve $\boldsymbol{w}_j(t)$ by applying one of the following learning algorithms.

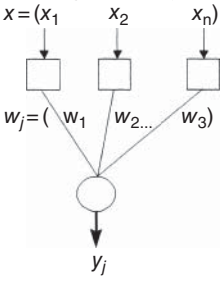
Unsupervised Competitive Learning This learning algorithm is used if no information about the class membership of the training data vectors is available. The change of the weights at iteration t is updated by

$$\boldsymbol{w}_j(t+1) = \boldsymbol{w}_j(t) + \eta_t[\boldsymbol{x}(t) - \boldsymbol{w}_j(t)] \tag{8.20}$$

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) \quad \text{for } i \neq j \tag{8.21}$$

If the difference $[\boldsymbol{x}(t) - \boldsymbol{w}_j(t)]$ is higher or lower than zero, the weight of the winning neuron $\boldsymbol{w}_j(t)$ is increased or decreased, respectively. The weights of the other neurons are not changed (Eq. (8.21)).

Competitive learning operates on the basis of a *next neighbor classifier*



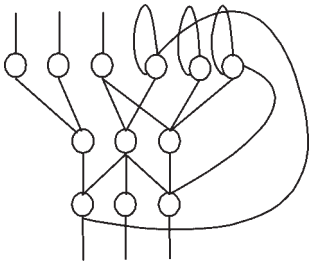
Supervised Competitive Learning If class assignments of $\boldsymbol{x}$ vectors are feasible, the weights can be trained in a supervised fashion. If we denote the class membership of the neuron j by D_j, then we have for the algorithm

$$\boldsymbol{w}_j(t+1) = \begin{cases} \boldsymbol{w}_j(t) + \eta_t[\boldsymbol{x}(t) - \boldsymbol{w}_j(t)] & \text{for} \quad x \in D_j \\ \boldsymbol{w}_j(t) - \eta_t[\boldsymbol{w}(t) - \boldsymbol{w}_j(t)] & \text{for} \quad x \notin D_j \end{cases} \tag{8.22}$$

The weight vector $\boldsymbol{w}_j$ learns positive, if $\boldsymbol{x}$ is correctly classified. It learns negative or forgets in the case of misclassification. This means the synaptic changes are carried out supervised or "reinforced."

Network Architecture

For optimization of the architecture of a neural network, different possibilities exist:



- Variation of the number of input and output neurons
- Change of the number of hidden layers
- Addition or elimination of neurons in a particular layer
- Modification of the connections of neurons within a layer and between layers. As a consequence, the number of weights is modified
- Selection of those neurons that receive a correction signal
- Definition of the information flow, which can be directed forward, backward, or recurrent.

The variety of network architectures complicates optimization of a neural network. In addition, further parameters, such as the learning coefficient, have to be adjusted. The danger of designing overdimensioned networks, which are redundant with respect to several parameters, is therefore large.

Neural Network Models

Perceptron

The simplest neural network is the perceptron. It was introduced by E. Rosenblatt (1950) and served the purpose of optical pattern recognition, that is, it represents a very simple model of the retina of the eye.

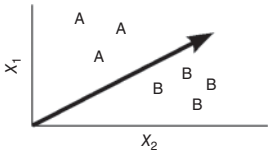
Formally, it can be categorized as a one-layer net with two inputs, x_1 and x_2, and one output, y, in the sense of a classifier for linearly separable classes. For this classifier, it is valid at time or cycle t that

$$y(t) = f(Net) = f\left(\sum_{i=1}^{n} w_i(t)\, x_i(t) + s\right) \tag{8.23}$$

$$y(t) = \begin{cases} 1 & \text{for } Net \geq s \\ 0 & \text{for } Net < s \end{cases} \tag{8.24}$$

here, s is again a threshold.

The learning procedure corresponds to a supervised learning process according to an associative learning law, that is,

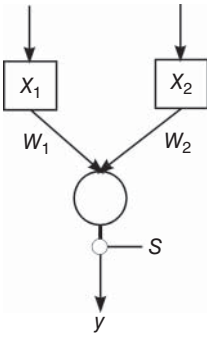


$$w_i(t+1) = w_i(t) + \eta_t[D(t) - y(t)]x_i(t) \tag{8.25}$$

$D(t)$ represents the class membership:

$$D(t) = \begin{cases} 1 & \text{for class A} \\ 0 & \text{for class B} \end{cases} \tag{8.26}$$

The perceptron can be compared with the linear learning machine (Section 5.3). As demonstrated by Minsky and Papert (1969), certain problems cannot be solved by using a simple perceptron. As an example, there is the exclusive OR connection as given in Table 8.3 (cf. Figure 7.8).



In the case of two input neurons, x_1 and x_2, for the following model for the separation plane

$$y = f(Net) = \begin{cases} 1 & \text{for } w_1x_2 + w_2x_2 \geq s \\ 0 & \text{for } w_1x_2 + w_2x_2 < s \end{cases} \tag{8.27}$$

Table 8.3 Exclusive OR (XOR problem).

x_1	x_2	y	Class
0	0	0	A
0	1	1	B
1	0	1	B
1	1	0	A

A graphical interpretation of the XOR problem is given in Examples 8.5 and 8.6.

Example 8.5 *XOR problem*

The problem of the exclusive OR is illustrated in Figure 8.10. The problem is to find a plane that separates the classes A and B from each other. As the separation plane drawn in Figure 8.10 demonstrates, this is not possible. Also no "better" separation planes could be found.

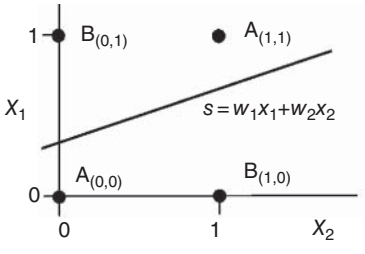
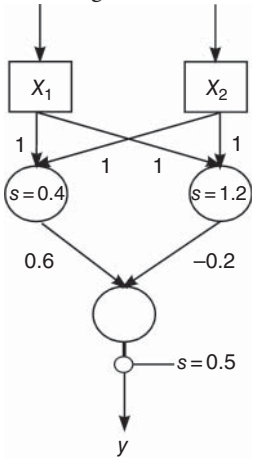


Figure 8.10 Insolubility of the XOR problem by using a network based on a *single* separation plane.

A solution exists if a network is constructed that contains at least two neurons in the hidden layer. Example 8.6 gives one possibility to solve the XOR problem.

Example 8.6 *Solution of the XOR problem*

To solve the XOR problem in Table 8.3, we construct the following neural net with two neurons in the hidden layer:



If the weights are adjusted such as given in the figure, then the following initial values are obtained for the two input vectors in (a) and (b):

a) $x_1 = 1$ and $x_2 = 1$;

$$y = s_{0.5}(0.6 \cdot (s_{0.4}(1 \cdot 1 + 1 \cdot 1)) - 0.2 \cdot (s_{1.2}(1 \cdot 1 + 1 \cdot 1)))$$
$$= s_{0.5}(0.6 \cdot (s_{0.4}(2)) - 0.2 \cdot (s_{1.2}(2))) = s_{0.5}(0.6 - 0.2) = 0$$

b) $x_1 = 0$ and $x_2 = 1$;

$$y = s_{0.5}(0.6 \cdot (s_{0.4}(1 \cdot 0 + 1 \cdot 1)) - 0.2 \cdot (s_{1.2}(1 \cdot 0 + 1 \cdot 1)))$$
$$= s_{0.5}(0.6 \cdot (s_{0.4}(1)) - 0.2 \cdot (s_{1.2}(1))) = s_{0.5}(0.6 - 0) = 1$$

Graphical interpretation of the net provides Figure 8.11.

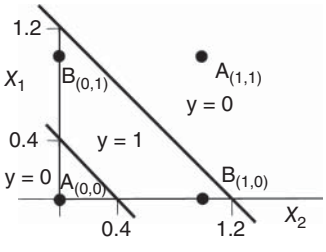
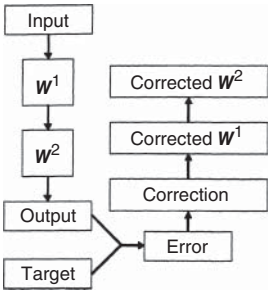


Figure 8.11 Solution of the XOR problem with a network that contains two neurons in the hidden layer.

Backpropagation Networks

The notion *backpropagation* characterizes the kind of error correction done backward from the output layer:



Multilayer perceptrons are used nowadays in connection with the backpropagation algorithm. In analytics, more than 90% of applications are based on this learning algorithm. A first approach for this forward directed learning algorithm was made by Werbos (1974). This was further developed by McClelland and Rumelhart [11] (Figure 8.12).

Learning Algorithm The basis of the backpropagation algorithm is the generalized delta rule (cf. Eq. (8.18)). The signal vector produced at the output of the net, ***y,*** is compared to a target vector, ***t.*** Mathematically, the path of steepest descend is traced (cf. Section 6.3). The individual steps of the algorithm are as follows:

$$\Delta w_{ij}^{l}(t+1) = \eta_t \delta_j^{l-1} y_i^{l-1} + \alpha \Delta w_{ij}^{l}(t) \tag{8.28}$$

where l is the index of the actual layer, t is time, η_t the learning coefficient, δ the error, and α the momentum.

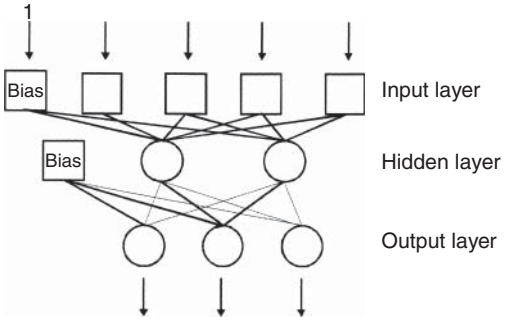


Figure 8.12 Multilayer perceptron as basis for a backpropagation network.

The momentum is applied to increase the convergence rate for learning. By means of a momentum parameter, α, during computation, part of the hitherto performed weight changes is maintained.

The error signal in the output layer is obtained by

$$\delta_j^{\text{last}} = (t_j - y_j) f'(y_j) \tag{8.29}$$

For the neurons in the hidden layer, the error signal is calculated by

$$\delta_j^l = \sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1} f'(y_j^l) \tag{8.30}$$

where r is the number of neurons in the layer $l+1$ and the derivation of the sigmoid function is

$$f'(y) = y(1-y) \tag{8.31}$$

Before training of a backpropagation network, the following settlements are required:

- Selection of the network architecture including the labeling of the bias neurons.
- Initialization of weights by random numbers in a predefined interval, for example, [−0.1, +0.1], or by least squares estimations.
- Adjusting values for the learning rate and the momentum in Eq. (8.28).
- Selection of a stopping criterion based on a maximum iteration number, a threshold for the error between the output and target value, or a threshold for the change of weights.

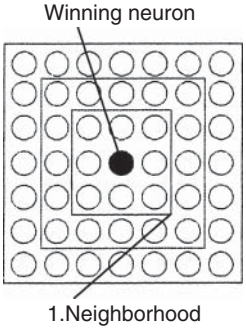
The backpropagation net is an example of a supervised learning neural network. Its applications in analytics are given in Table 8.4.

In the following, we consider models of unsupervised learning networks.

Kohonen Network

Kohonen networks are also termed *self-organizing nets* or *self-organizing feature maps*.

Neural networks for unsupervised learning are based on a competitive layer of weights arranged linearly or in a plane (Figure 8.13). If arranged in a plane, the nets are termed a *Kohonen network*. The peculiarity of this network type is the maintenance of topology or, more general, the pattern of the data vector to be learned.



The preservation of topology results from the introduction of neighborhood relationships of neurons in the learning algorithm. These relationships are described by a neighborhood function with the distance measure d as the independent variable. This function characterizes the distance to the winning neuron:

$$nf = nf(d) \tag{8.32}$$

Typical distance functions are based on triangular functions or the Mexican hat function (Figure 8.14).

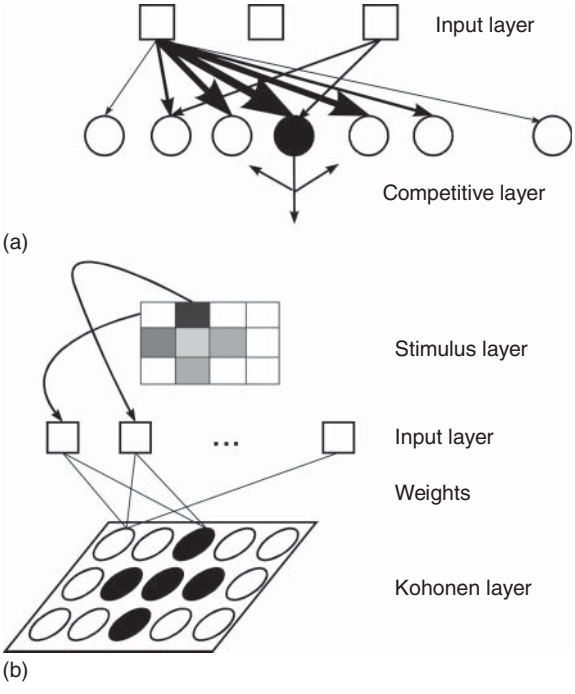


Figure 8.13 Structure of self-organizing networks in linear arrangement (a) of the competitive layer and in Kohonen representation (b).

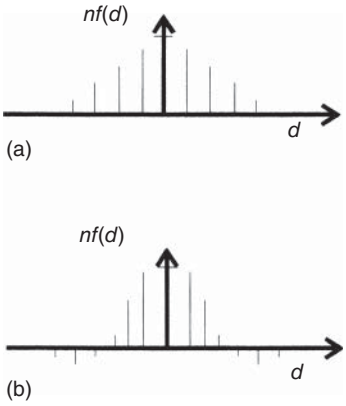


Figure 8.14 Neighborhood function in the form of a triangle (a) and a Mexican hat (b).

The learning algorithm for the Kohonen net operates in analogy to the competitive learning laws in Eqs. 8.19–8.21:

- Presentation of an input vector $\boldsymbol{x}$.
- Calculation of the distance between the input vector and each output unit i to evaluate the winning neuron j:

$$\|\boldsymbol{w}_j(t) - \boldsymbol{x}(t)\| = \min_i \|\boldsymbol{w}_i(t) - \boldsymbol{x}(t)\| \tag{8.33}$$

- Change of the weight for the winning neuron and its neighbors by using the neighborhood function is carried out according to Eq. (8.32):

$$w_{ik}(t+1) = w_{ik}(t) + \eta_t nf(d)[x_i(t) - w_{ik}(t)] \tag{8.34}$$

  where k is the neighborhood element at time t.
- Continuation of the learning process with the subsequent $\boldsymbol{x}$ pattern.

As a result of learning, the patterns are arranged in clusters, presupposed the data vectors can be grouped in a Kohonen layer. The clusters can be explored now for assigning objects to them.

The result of unsupervised Kohonen learning, however, can also be used for *classification.* For this, an additional layer is introduced. The output information trained by the Kohonen net is further trained on the known patterns or class information by means of an associative learning law. After adjustment of the additional weights, the net can be subsequently applied for classifications.

Applications of neural networks are given in Table 8.4. The applications are, in principle, similar to those discussed in the statistical Chapters 5 and 6, that is, grouping, classification, and modeling of data. In addition, the nets also serve the purpose of knowledge processing, for example, for machine learning of rules.

Table 8.4 Applications of neural networks in analytics.

Objective	Neural net	Area	Reference
Parameter estimation	Backpropagation	Quantitative analysis in NIR	[14]
Clustering	Kohonen	IR spectra grouping	[15]
Classification	Backpropagation	MS structure elucidation	[16]
Identification	Backpropagation	HPLC diode array detection	[17]

Bootstrapping is restrained to the use of the original data set to estimate confidence intervals. Part of the data (rows of the data matrix) is sorted out and used for later predictions. The missing rows in the matrix are replaced randomly by data vectors kept. The latter vectors are then used twice in a computational run (cf. "Tree-Based Classification" Section).

If used for modeling, it should be noted that estimation of confidence intervals for the weights (parameters) or for predictions cannot be performed analytically because of the nonlinearity of the networks. As with other nonlinear methods, the confidence intervals have to be estimated by means of Monte Carlo simulations or bootstrapping methods.

At present, investigations for applying neural networks in analytics are done without reference to statistical methods. Future studies will show which applications of the networks could be superior to existing multivariate methods including the nonlinear statistical methods.

In Example 8.7, the use of a multilayer perceptron for classification of data sets in two dimensions is explored.

Example 8.7 *Artificial neural network*

The performance of a neural network is explored here for classification of two-dimensional input data by means of a feedforward neural network. The simulated data are known from Example 5.12 and describe 200 cases from four classes (Figure 8.16).

The network consists of two input neurons for presentation of the two x-values as well as four output neurons, y, which represent the four classes (cf. Figure 8.15). In addition, a hidden layer was added with up to 20 neurons and the intercepts of the surfaces are modeled by bias neurons to both the hidden and output layers. The transfer function in the neurons of the hidden layer was of sigmoid type, and aggregation of the neurons in the output layer was carried out by calculating the normalized exponentials (softmax criterion).

The optimum number of neurons in the hidden layer varied with the type of training algorithm. If a *Bayesian regulation backpropagation* algorithm is used, the optimum number of

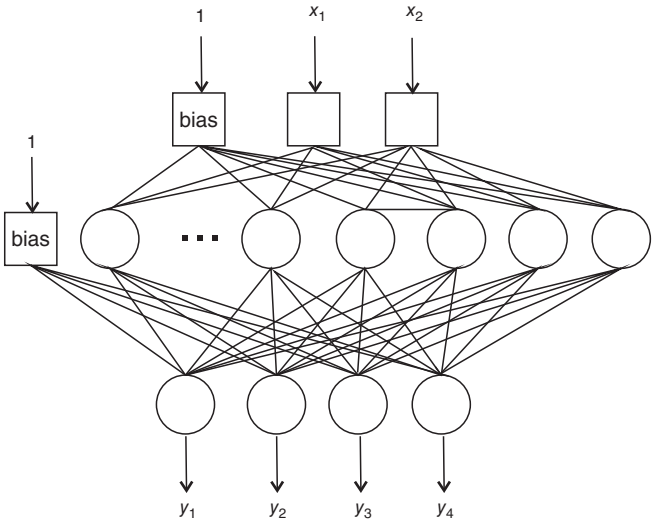


Figure 8.15 Artificial neural network for classification of four classes based on two-dimensional input data.

hidden neuros was found to be 16. The decision boundaries for this training algorithm are depicted in Figure 8.16a. Although only three wrong assignments (1.5% misclassifications) were found with this training algorithm for resubstitution of the cases, it is to be expected that predictions of new cases on the basis of these flexible decision boundaries will not be at the same misclassification level. A less flexible decision boundary is found with a *conjugate gradient backpropagation* algorithm. The number of neurons in the hidden layer then amounts to only nine, which is, however, connected with a somewhat higher fraction of misclassification of 7% for resubstitution (Figure 8.16b).

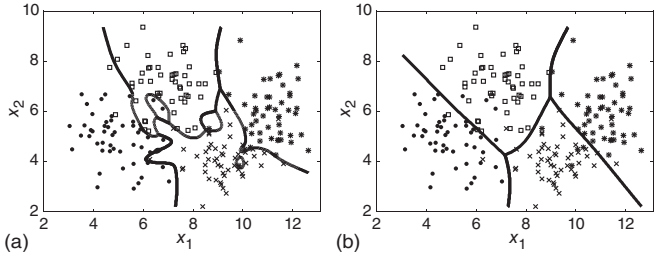
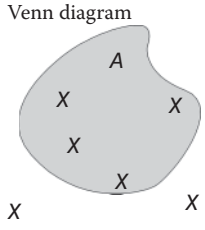


Figure 8.16 Decision boundaries of a feedforward neural network trained by a *Bayesian regulation* (a) and a *conjugate gradient* backpropagation algorithm (b).

Venn diagram



8.3 Fuzzy Theory

The theory of fuzzy sets enables representation and processing of vague propositions and uncertain information. In contrast to probability theory, fuzzy theory is a possibilistic approach, that is, it is based on possibility theory. The concept of fuzzy sets was introduced by Lotfi A. Zadeh of the University of California (USA) in 1965.

Fundamentals

Fuzzy Sets

In classical set theory, the containment of an element x to a subset A of the universe of discourse X is described by a characteristic function. It is called *membership function*, $m(x)$. A membership value of 1 is assigned an element x, that is, contained in a set A. If x is not an element of the set A, a membership value of zero results:

$$m(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases} \tag{8.35}$$

The concept of fuzzy sets extends this crisp assignment by allowing membership values *between* 0 and 1. These fuzzy sets are normalized to the interval [0,1]. They can be represented either as discrete or by means of a function (cf. Eqs. 8.36 – 8.39).

Example 8.8 *Set theory*

The set of all elements detectable by an analytical method can be expressed by a set over the atomic number, x. According to classical set theory, detection of elements with atomic numbers greater than a particular value is feasible, for example, X-ray fluorescence analysis can be used for analyses of elements from sodium with atomic number 11. The membership function is given by the solid line in Figure 8.17.

In practice, however, there is a transition between the detectability and nondetectability of an element. Thus, it is not unusual with this analytical principle, given instrumentation of reasonable quality that even the element with atomic number 9 (fluorine) can be detected. The transition between the sets of detectable and nondetectable chemical elements is better described by a membership function based on a fuzzy

set. This membership function is plotted in Figure 8.17 as broken line.

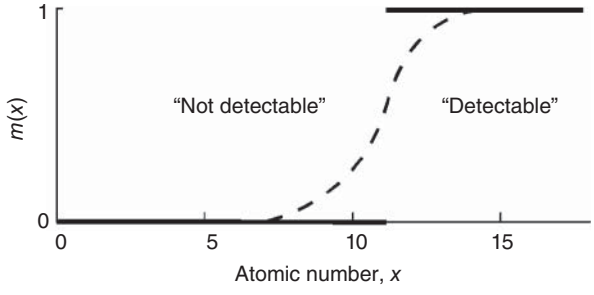


Figure 8.17 Membership function for the detectability of elements by X-ray fluorescence analysis. *Solid line* Classical (crisp) set, *broken line* fuzzy set.

Types of Membership Functions

Many possibilities exist for characterization of a fuzzy set. Depending on the objective, different types of membership functions are used.

Fuzzy Observations Consider first the description of uncertainty of experimental observations, such as the uncertain position of a spectroscopic line.

Note: membership functions are *not* probability distribution functions. A membership function can be specified by a *single* observation.

The fuzzy set is characterized, for example, by means of an exponential function of the following type:

$$m(x) = \exp\left(-\frac{|x-a|^2}{b^2}\right) \tag{8.36}$$

here a is equal to the x value with membership value $m(x) = 1$. The constant $1/b^2$ normalizes the membership function to the interval [0,1]. As an example, Figure 8.18a demonstrates exponential membership functions with $a = 9$ and $b = 3$.

For operations with exponential functions, truncation at a certain spread by setting the membership value, $m(x)$, to zero is recommended. Some membership function types provide a more "natural" truncation, for example, a function of quadratic type:

$$m(x) = \left[1 - \frac{|x-a|^2}{b^2}\right]^+ \tag{8.37}$$

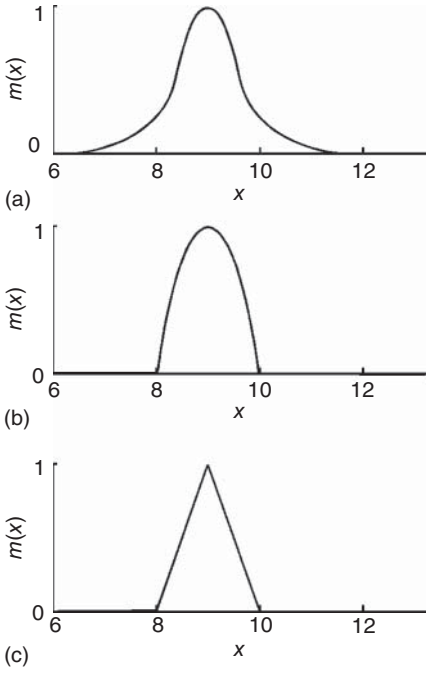


Figure 8.18 Membership functions for characterization of uncertainty of experimental observations in the form of exponential (a), quadratic (b), and linear (c) functions.

The plus sign denotes the restricted validity for positive values of the membership function. Negative values for $m(x)$ are set to zero (Figure 8.18b).

Important is the *monotonicity* of the membership function. The special form of the membership function has only a weak influence on the result of fuzzy operations. The parabola function in Eq. (8.37) can be approached, therefore, by a triangular function (Figure 8.18c):

$$m(x) = \left[1 - \frac{|x - a|}{b}\right]^{+} \tag{8.38}$$

where + again denotes truncation of the function to positive membership values.

In general, there are no restrictions for specification of membership functions. They can be based either on experimental observations or on experience. In addition, membership functions are not restricted to a single x variable. A membership function for experimental observations i in dependence on the

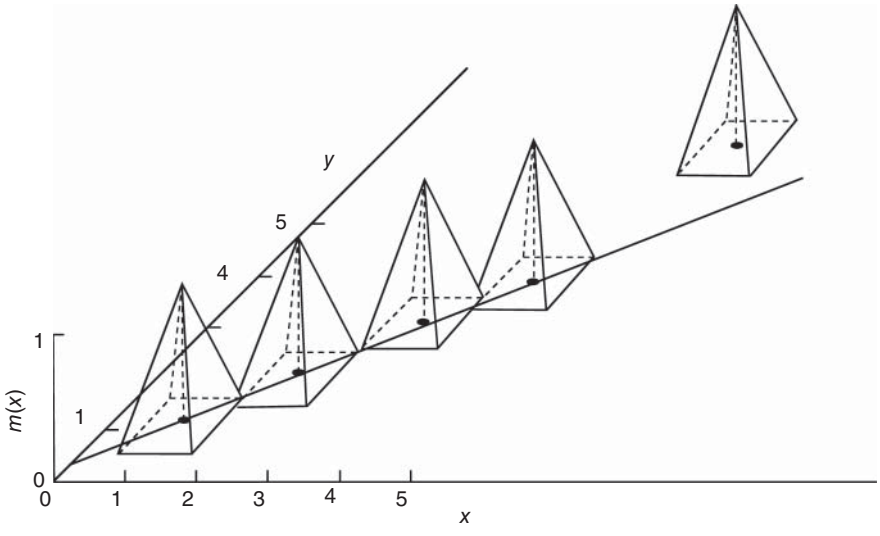


Figure 8.19 Two-dimensional membership function for fitting a straight line.

variables x and y could have, for example, the following form:

$$m(x,y) = \left\{ 1 - \left[\frac{|x - x_i|}{u_i} + \frac{|y - y_i|}{v_i} \right] \right\} \tag{8.39}$$

where u_i and v_i are constants.

The function describes a pyramid-shaped membership function (cf. Figure 8.19).

Linguistic Variables Verbal fuzzy expressions are described by fuzzy theory by means of linguistic variables; for example, the solubility of a substance in water can be characterized by the fuzzy terms "high" and "low." If the linguistic variables are further to be distinguished, the so-called *modifiers* are applied.

Figure 8.18 provides examples for linguistic modifiers, such as "very," "more or less," and "middle." The membership function "*high*" is normalized to the interval [0,1] and reads as (Figure 8.20)

$$m_h(x) = \begin{cases} 0 & \text{for } x < 0 \\ 2x^2 & \text{for } 0 \le x \le 0.5 \\ 1 - 2(x-1)^2 & \text{for } 0.5 \le x \le 1 \\ 1 & \text{for } x > 1 \end{cases} \tag{8.40}$$

Calculation of the modifiers is summarized in Table 8.5.

Truth Values For applications of multivalued logic, we need further graduation in addition to the binary truth values "true"

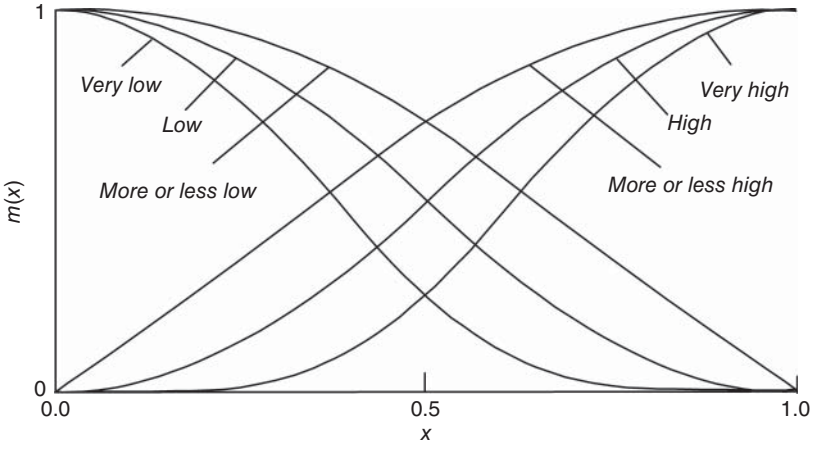
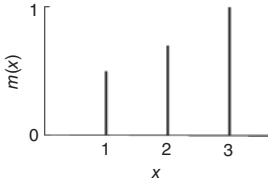


Figure 8.20 Membership functions for the linguistic variable "high" according to Eq. (8.40) together with different modifiers.

Table 8.5 Calculation of modifiers for the membership functions of the linguistic variables "high" according to Eq. (8.40).

Modifier	Formula
Very high	$m_{vh}(x) = m(x)^2$
More or less high	$m_{mlh}(x) = \sqrt{m(x)}$
Low (not high)	$m_l(x) = 1 - m(x)$
Very low	$m_{vl}(x) = [1 - m(x)]^2$
More or less low	$m_{mll}(x) = \sqrt{1 - m(x)}$
Middle	$m_m(x) = \min\{m(x), 1 - m(x)\}$

and "false." For this, we construct membership functions, which are normalized in the universe of discourse X to the interval [0,1]. The membership function is expressed, for example, by the functions introduced by Baldwin (Figure 8.21).



Discrete Membership Functions Often specification of continuous membership functions does not make sense. If, for example, the qualification of a sample state for a spectrophotometric analysis is to be described, discrete values, such as solid, gas, or liquid, are required.

Discrete membership functions in relation to the fuzzy set A are represented as pairs consisting of the variable value and the

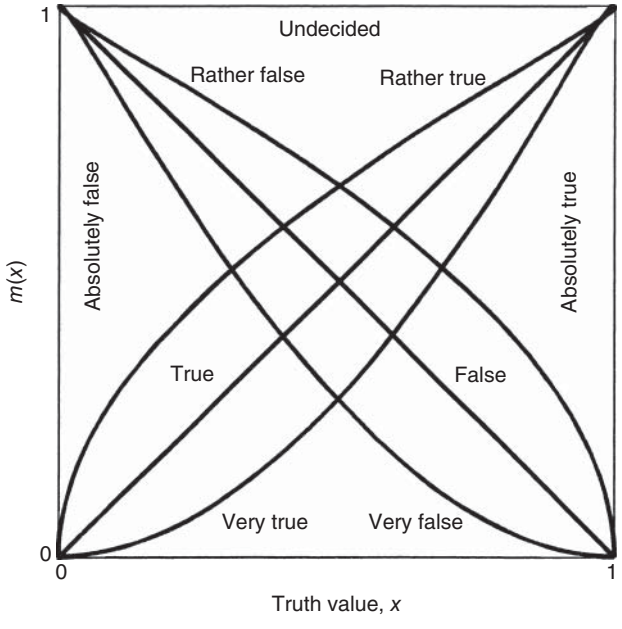


Figure 8.21 Membership functions for truth values after E. Baldwin.

corresponding membership value, for example:

$$A = |(1, 0.5), (2, 0.7)(3, 1.0)| \tag{8.41}$$

Operations with Fuzzy Sets

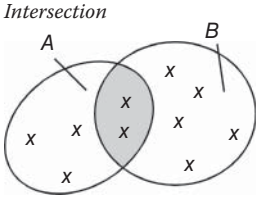
Fuzzy set operations are derived from classical set theory. In addition, there exist theories for calculating with fuzzy numbers, functions, relations, measures, or integrals.

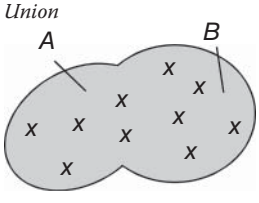
The *intersection* of two sets A and B corresponds, according to classical theory, to all elements that are simultaneously contained in both sets. For two fuzzy sets, the intersection, $A \cap B$, is derived from the minimum of both of the membership functions $m_A(x)$ and $m_B(x)$:

$$m_{A \cap B}(x) = \min\{m_A(x), m_B(x)\} \tag{8.42}$$

Two common sets are unified by aggregating all elements that belong to at least one of the two sets into one set. The *union* of fuzzy sets results from calculating the maximum of their membership functions:

$$m_{A \cup B}(x) = \max\{m_A(x), m_B(x)\} \tag{8.43}$$





The *complement* of a set A corresponds to all elements that are not contained in A. In analogy, the complement of a fuzzy set A is

$$m_{\overline{A}}(x) = 1 - m_A(x) \tag{8.44}$$

The number of elements in a set is termed *cardinality* in classical theory. The latter is obtained for fuzzy sets from summation or integration over the membership function of that set:

$$\text{card } A = \sum_i m(x_i) \text{ or card } A = \int_X m(x)dx \tag{8.45}$$

Cardinalities in the interval [0,1] are obtained by calculating their *relative cardinalities.* For this, the standard set S is taken as the basis:

$$\text{rel}_S \text{ card } A = \frac{\text{card } A}{\text{card } S} \tag{8.46}$$

Figure 8.22 illustrates the set operations.

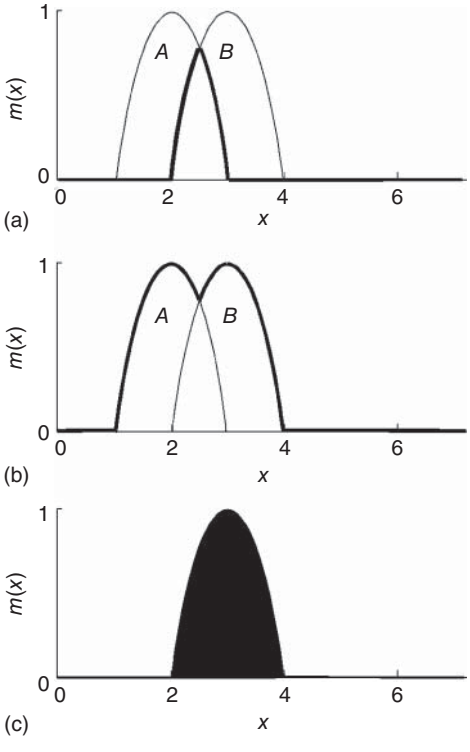


Figure 8.22 Intersection (a) and union (b) of two fuzzy sets and the cardinality of a fuzzy set (c).

Example 8.9 *Cardinality of a fuzzy set*

The relative cardinality of the fuzzy set in Figure 8.22c is to be calculated. The membership function for this set is

$$m(x) = [1 - |x - a|^2]^+ \tag{8.47}$$

Integration over the membership function in the interval $x = 2$ to $x = 4$ provides for the relative cardinality according to Eq. (8.46):

$$\text{rel}_S \text{ card } A = \frac{\int_2^4 [1 - |x - 3|^2] \mathrm{d}x}{4 - 2} = 0.666 \tag{8.48}$$

Fuzzy inference systems are also known as *fuzzy associative memories, fuzzy models, fuzzy-rule-based systems,* or *fuzzy controllers.*

Rule-Based Fuzzy Systems

We learned the fundamentals about inferences on rule-based systems in Section 8.1 on symbolic knowledge processing. One possibility to infer on the knowledge represented in the computer is based on IF–THEN rules. In fuzzy logic, the premises and consequences are described by a fuzzy relation. Consider the following rules:

$$\begin{aligned} &\text{IF } x = A_1 \text{ THEN } y = B_1 \\ &\text{IF } x = A_2 \text{ THEN } y = B_2 \\ &\vdots \\ &\text{IF } x = A_n \text{ THEN } y = B_n \end{aligned} \tag{8.49}$$

With the additional proposition

$$x = A' \tag{8.50}$$

one can reason about the unknown fuzzy set of x, that is, B', by means of one of the different inference schemes. In those inference schemes, the fuzzy relation between the fuzzy sets A and B, $A \times B$, for the IF–THEN rule, is described by their membership functions. The most used scheme is based on Zadeh's *compositional rule of inference:*

$$M_{A \times B}(x, y) = \min\{m_A(x), m_B(y)\} \tag{8.51}$$

According to the compositional rule of inference, the membership function for the consequence B' is obtained over all rules by

$$m_{B'}(y) = \max\min\{m_{A'}(x), m_{A \times B}(x, y)\} \tag{8.52}$$

Finally, the results for the B_i's have to be aggregated, for example, by calculating the minimum over all B_i's.

Tagaki and Sugeno suggested rule-based systems that involve fuzzy sets only in the premise part and the consequence part consists of a nonfuzzy function, for example,

$$\text{IF blank} = \text{high THEN detection limit} = y_B + 3s_B \tag{8.53}$$

Here, the premise is described by a membership function for the linguistic variable "high" and the function for the detection limit is the sum of the blank signal, y_B, and three times the standard deviation of the blank signal, s_B, (cf. Eq. (4.3)). Optimization of the parameters in the premise part of the rules is adaptively done by combining the fuzzy rule-based system with a neural network. Consider an adaptive neuro-fuzzy system with two inputs, x_1, and x_2, and one output, y. Then, the following fuzzy IF–THEN rules might be valid:

$$\begin{aligned} &\text{IF } x_1 = A_1 \text{ AND } x_2 = B_1 \text{ THEN } f_1 = p_1 x_1 + q_1 x_2 + r_1 \\ &\text{IF } x_1 = A_2 \text{ AND } x_2 = B_2 \text{ THEN } f_2 = p_2 x_1 + q_2 x_2 + r_2 \end{aligned} \tag{8.54}$$

A square (adaptive) node has parameters in contrast to a circle (fixed) node that has none.

with p, q, and r being parameters of the output functions f. The structure for such an *adaptive neuro-fuzzy inference system* is illustrated in Figure 8.23. In layer 1, each node is a square node that receives the input value of x and outputs the membership value from the membership function, $m_{Ai}(x)$ and $m_{Bi}(x)$. The nodes in layer 2 are represented by circle nodes that multiply the incoming signals and send the product out according to

$$w_i = m_{Ai}(x_1) \times m_{Bi}(x_2), \quad i = 1, 2 \tag{8.55}$$

The weight, w, is also called the *firing strength of a node*. In the circle node in layer 3, the firing strengths are normalized to the sum of all rules' firing strengths:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{8.56}$$

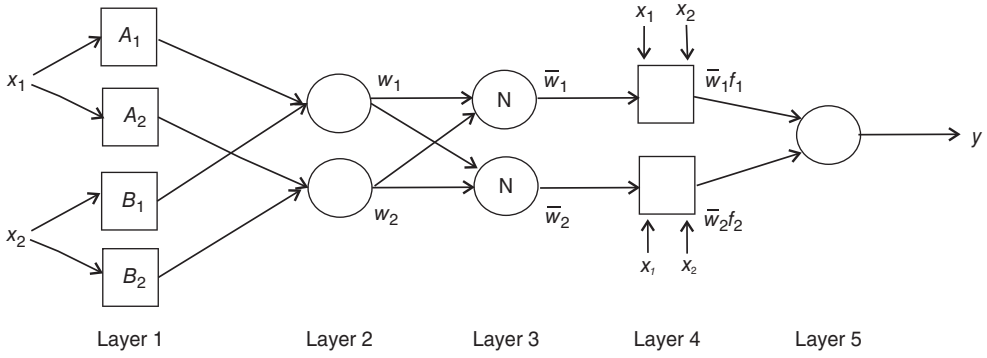


Figure 8.23 Adaptive neuro-fuzzy inference system.

The square nodes in layer 4 multiply the normalized firing strengths with the node functions as follows:

$$\overline{w}_i f_i = \overline{w}_i (p_i x_1 + q_i x_2 + r_i), \quad i = 1, 2 \tag{8.57}$$

The circle node in layer 5 aggregates the incoming signals by summation and provides as overall output:

$$y = \sum \overline{w}_i f_i = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \tag{8.58}$$

In Example 8.9, the performance of this adaptive neuro-fuzzy inference system is demonstrated for classification of two-dimensional data.

Example 8.10 *Adaptive neuro-fuzzy inference system (ANFIS)*

Here we consider again the data in two dimensions (Figure 8.24a) given for a four-class problem in Example 5.12 on k-NN classification. For validation of the neuro-fuzzy system, the 200 cases were divided into a training data set of 160 cases and a test data set of 40 cases.

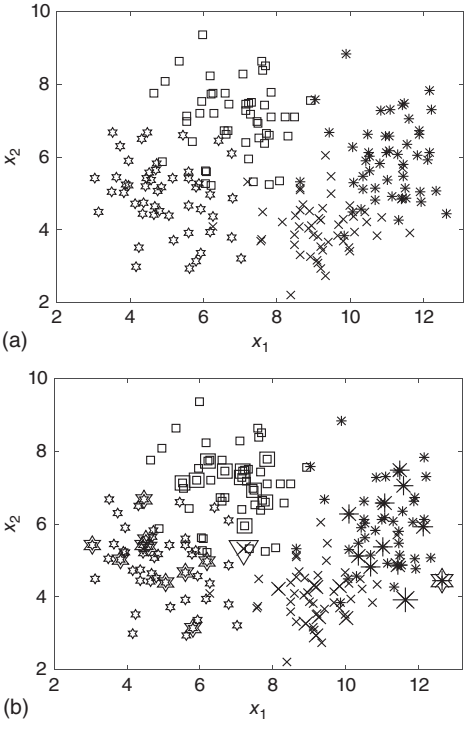


Figure 8.24 Simulated data (a) and assignment of cases in the test data set (b).

The number of bell-shaped membership functions in layer 1 (Figure 8.23) was varied between 4 and 12. In case of a high number of membership functions and up to 200 training epochs resubstitution of the training data revealed perfect classification. However, the assignment of the test cases is rather poor. Therefore, for stable classification of the test data, only 4 membership functions (cf. Figure 8.23) and 20 training epochs were found to be optimal. There were only three misclassified objects of the test data set found, that is, the fraction of misclassifications was 7.5%. The result of validation is visualized in Figure 8.24b, where the assigned 40 test cases are labeled by a greater marker size than the cases of the training set. The three wrong assignments are labeled by an even larger size of the marker. Whereas the two misclassifications around x_1-values of 12 (star and hexagram markers) are just wrong assignments to one of the four classes, it also happened that the neuro-fuzzy system predicted class assignments to additional classes. Thus, the downward-pointing triangle in the middle of the plot characterizes an assignment to a fifth class that was not trained in the model.

Applications

Fuzzy theory can be used either for data analysis or for dealing with fuzzy logic. Typical applications in analytics are [20] as follows:

- Pattern recognition based on unsupervised and supervised learning
- Multicriteria optimization
- Comparisons of spectra, chromatograms, or depth profiles on the basis of fuzzy functions
- Fuzzy modeling
- Fuzzy logic for fuzzy control and for approximate reasoning.

We have already discussed an example for *grouping* data on the basis of unsupervised learning with respect to fuzzy cluster analysis by the *c*-means algorithm (Section 5.2).

For solving problems of *supervised learning*, fuzzy methods are useful if irregularly cast data sets are to be handled. For example, in the analysis of human cell tissues by means of capillary gas chromatography (GC), the number of chromatographic peaks (components) varies by ±15 peaks per chromatogram. This is explained by the high biological variability and not due to random (analytical) error. A statistical procedure for pattern recognition

cannot be used here, since the vectors (chromatograms) to be compared would require an equal number of elements in the vector, that is, the same number of peaks in the chromatogram. By means of set theoretic approaches, differently long data vectors can be evaluated.

For *multicriteria optimization*, the individual criteria are described by means of fuzzy sets and are aggregated then to an appropriate objective function. To define the membership functions of those objective functions, heuristic knowledge can be included.

For fuzzy modeling or for comparison of fuzzy functions, the fundamentals of *fuzzy arithmetics* are needed (cf. Figure 8.25). These fundamentals are given, for example, in references [19] or [20]. Applications are known for calibration of analytical methods and for qualitative and quantitative comparison of chromatograms, spectra, or depth profiles.

Another important area of application is approximate reasoning based on *fuzzy logic*. Fuzzy inferences are applied in analytical expert systems or for controlling chemical or biotechnological reactors.

Table 8.6 comprises some applications of fuzzy theory in analytics.

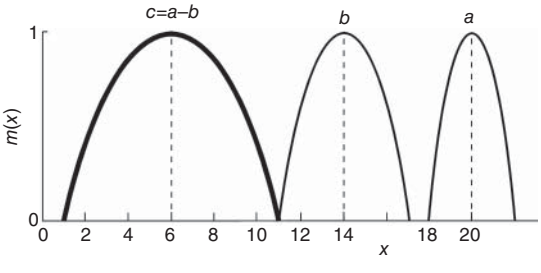


Figure 8.25 Example of a fuzzy difference of two numbers, that is, about 20 minus about 14 gives about 6.

Table 8.6 Applications of fuzzy theory in analytics [20].

Aim	Fuzzy method	Area
Unsupervised learning	Fuzzy clustering	Beer sorts
Supervised learning	Fuzzy pattern recognition	Chromatograms
Library search/identification	Set operations	IR spectra
Quality control	Fuzzy functions	Depth profiles, spectra
Modeling	Fuzzy numbers	Calibration with errors in x and y
Multicriteria optimization	Fuzzy sets	Enzymatic determination
Expert system	Fuzzy logic	X-ray fluorescence analysis [8]

8.4 Genetic Algorithms and Other Global Search Strategies

Genetic Algorithms

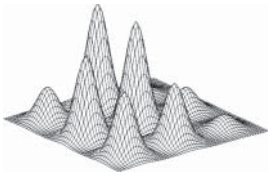
Genetic algorithms belong to methods that have a biological analog as do neural nets. These methods are based on *evolutional components,* such as population of living beings, competition between living beings, and reproduction of life. As a result, new living beings are formed that are more highly developed, or those living beings that have reached the end of their lives are to be replaced.

Natural computation denotes the following methods: simulated annealing, simulated evolution strategy, genetic algorithms, artificial immunonetworks, artificial neural nets, artificial chemical reaction systems, and artificial live (cellular atoms and fractals).

Genetic algorithms are especially useful for dealing with highly complex and highly dimensional *search problems.* This can be related to optimization of a set of variables in the sense of feature selection. A typical application is the selection of wavelength in spectrometric multicomponent analysis (cf. Section 6.2). A second application concerns optimization of parameters, for example, for multicriteria optimization or for parameter estimation in nonlinear models. Here, multipeaked objective areas are operating. Many other optimization methods may not find the global optimum in such cases.

Apart from these numerical applications, genetic algorithms can also be used to solve combinatorial problems, for example, the processing of a sample queue in a lab.

Heredity in the Computer



The starting point with genetic algorithms is a population of *living beings.* A computer-adaptable living being consists of a particular set of variables. Each variable is represented by a *chromosome* that is characterized by a set of genes. Usually, the chromosomes are binary coded (cf. Section 1.3), so that each bit corresponds to a *gene.* If an 8-bit representation is used, then the chromosome consists of eight genes and each gene can takes as an *allele* the value of either 0 or 1.

Initial Population

8-Bit chromosome

0	1	1	0	0	0	1	0

A genetic algorithm starts with an initial population of living beings, which are coded as character strings. Generation of the initial population can be performed, for example, by random coin throws providing the bits 1 and 0 (cf. Example 8.10). Subsequent populations are generated by the genetic algorithm by *heredity* in the form of selection, crossover, and mutation of the living beings.

Selection

Selection or reproduction means multiplication as adaptation to environment or survival of dangers. Formalized, the selection is based on the value of the objective function y for the living considered. In the simplest case, the objective function is a signal. However, it can also be a composed quantity (cf. Section 4.1).

The new living being should develop a maximum (minimum) output signal. Typically, the signal values are averaged. A good living being reveals a signal that is greater than the average signal. A bad living being provides one that is lower than the average. The number of living beings prone to multiplication can be derived from the signal value, which is weighted over the sum of all signals of the population.

Crossover

In case of crossover, randomly, a father and mother are selected. Also, a crossover position is fixed in the chromosome or in the bit character string, which is used to derive the aggregation of the genes of the parents. All bits left of the crossover point are transferred to the child from the father and all bits right of the point are given by the mother and vice versa.

Mutation

Mutations are changes as a result of external effects. For individual bits, it is decided on the basis of a predefined probability whether a bit switches or not.

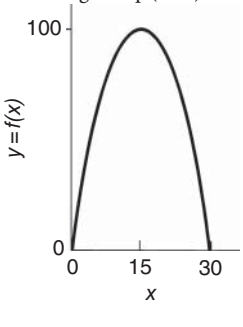
A prerequisite for a well-operating genetic algorithm is a well-performing random generator. However, one should note that genetic algorithms are not solely based on chance (randomness), as is valid for Monte Carlo simulations. In this case, randomness is only a tool to successfully proceed in the space of the objective area.

Example 8.11 *Genetic algorithm*

By means of a genetic algorithm, the optimum for a response (objective function) y in dependence on the variable x is sought. To facilitate understanding and following of the calculations, the response values are generated by the function in Eq. (8.59). In practice, the responses are experimental observations.

$$y = [100 - 0.44(x - 15)^2]^+ \tag{8.59}$$

Objective function according to Eq. (8.49):



First, variable x is to be coded as chromosome. For the x values of interest between 0 and 30, a single 5-bit string is sufficient. Random selection of the initial population reveals the following four living beings (cf. x values in Table 8.7):

1 0 0 1 1
0 0 0 0 1
1 1 0 1 0
0 0 1 0 1

Table 8.7 Selection in the genetic algorithm in Example 8.11.

No.	Initial population	x	$y = f(x)$	$\frac{y_i}{\sum y}$	Expected number, $\frac{y_i}{\bar{y}}$	Actual number
1	1 0 0 1 1	19	93	0.44	1.77	2
2	0 0 0 0 1	1	14	0.07	0.27	0
3	1 1 0 1 0	26	47	0.22	0.89	1
4	0 0 1 0 1	5	56	0.27	1.07	1
	Sum	—	210	1.00	4.00	4
	Average	—	52.5	0.25	1.00	1

For *selection,* the corresponding responses are measured or calculated according to Eq. (8.49). Then, the living beings can be judged in relation to their adaptational ability. The new population is to be formed from the surviving exemplars.

Table 8.7 summarizes the individual steps of the genetic algorithm. The four-character strings of the initial population are given with the corresponding x values and the responses y in columns 3 and 4. The weights of the y values related to the sum of all responses of the population (selection probability) show a higher value compared to the average for living being 1, but the response for living being 2 is obviously below the population average. In the sense of propagation, more frequent reproduction is expected for living being 1 than for being 2. The expected number of reproductions can be estimated from the relationship to the average of the response values (column 6 of Table 8.7). As the discrete number of reproductions, we obtain two copies for living being 1, no copy for living being 2, and for living beings 3 and 4, one copy each. The new population is then

1. 1 0 0 1 1
2. 1 0 0 1 1
3. 1 1 0 1 0
4. 0 0 1 0 1

In the next step, *crossover* of living beings is carried out. In our example, from the new population, the living beings 1 and 3 and 2 and 4 are randomly chosen as pairs for crossover. Crossover points are the positions 4 and 2. As a result, one obtains the crossovers given in Table 8.8.

Table 8.8 Crossovers in the genetic algorithm in Example 8.11.

No.	Father	Mother	Crossover point	Child	x	$y = f(x)$
1	1 0 0 1|1	1 1 0 1|0	4	1 0 0 1 0	18	96
2	1 0 0 1|1	1 1 0 1|0	4	1 1 0 1 1	21	84
3	1 0|0 1 1	0 0|1 0 1	2	1 0 1 0 1	27	37
4	1 0|0 1 1	0 0|1 0 1	2	0 0 0 1 1	3	37
	Total					254
	Average					63.5

As a result of crossover, a population emerges for which the sum of y values or their average became larger (cf. Tables 8.7 and 8.8), so we have approached the maximum of the objective function more closely.

An additional change in the population would be feasible on the basis of mutations. For a probability of, for example, 0.005 in total 20 of the transferred bits should switch, that is, $20 \times 0.005 = 0.1$. This means, practically, that no bit will mutate from 0 to 1 or vice versa.

Applications

The applications of genetic algorithms are determined by some particularities, which are as follows:

- The algorithms do not operate on the basis of the input variables themselves, but on coded variables.
- The search for the optimum starts from several points.
- The criterion for optimization is the objective function and not, for example, its derivative.
- The forward movements are not deterministically based, but probabilistically.

Traditional search methods move in the objective area from one point to the next on the basis of a given search rule. In multimodal objective areas, one would end up by this point-by-point forward movement at the closest peak, which might be far from the global optimum. Genetic algorithms start on the basis of a population of

Table 8.9 Applications of genetic algorithms.

Aim	Strategy	Area
Learning algorithm for neural nets	Optimization of weights	X-ray fluorescence analysis
Wavelength selection	Selection of subsets	Multicomponent analysis
Multicriteria optimization	Optimization	Atomic emission spectrometry
Prediction of retention data	Numerics	HPLC

points and move from those positions in parallel in the direction of the optimum.

Genetic algorithms are most frequently used for wavelength selection in multicomponent analysis. If, for example, 5 wavelengths are to be chosen out of 30 wavelengths in total, then a 30-bit chromosome could be defined. The actual wavelength was coded by 1's and all other bits are set to 0. If the wavelengths 3, 7, 13, 22, and 27 were tested, then the chromosome resembles the following:

0010001000001000000000100001000

Further applications range from multicriteria optimization up to the replacement of the backpropagation learning algorithm for training neural networks (Section 8.2) by genetic learning algorithms (Table 8.9).

Simulated Annealing

Another strategy to circumvent to be trapped in a local optimum is simulated annealing (SA). Annealing is a process in which a solid material is first melted and then allowed to cool by slowly decreasing the temperature. In contrast to conventional, local optimization methods, in SA steps are allowed in a direction that yields inferior solutions. To control these steps, a probability function that originates from Metropolis work in statistical thermodynamics is exploited.

Movement of a system to a state with higher energy, E, can be described in thermodynamics by

$$p(\delta E) = e^{-\frac{\delta E}{kT}} \tag{8.60}$$

where $p(\delta E)$ is the probability, T is the temperature, E is the energy, and k is the Boltzmann constant. After introducing a perturbation in the system, the corresponding energy change is computed. If the energy is lower, the perturbation will be accepted. If the energy is higher, the perturbation is accepted

with the probability computed by Eq. (8.60). Decreasing the temperature iteratively, the system is cooled into a frozen state where no changes yielding energy increase would be accepted.

Comparable to genetic algorithms, there are candidate solutions that are determined in SA by the different states of the system and the energy of the state as the objective function. The control parameter in SA is called *temperature*, as used in thermodynamics. The Boltzmann constant is termed a *cooling parameter*, α, which ranges between 0 and 1.

In analogy to a typical local optimization method, such as the simplex method (cf. Section 4.3), SA could be programmed as given in Example 8.10.

Example 8.12 *Simulated annealing algorithm*

1. Select a starting point, S_0.
2. Compute S_0 by evaluating $f(S_0)$.
3. Select an initial temperature value, T.
4. Select an initial cooling parameter value, α, to reduce temperature.
5. Select a neighboring solution randomly.
6. Compute $\delta = f(S) - f(S_0)$.
7. If $\delta < 0$, accept S as the new solution.
8. If $\delta > 0$, accept S with a probability, $p(\delta E)$; keep S_0 with a probability, $1 - p(\delta E)$.
9. Repeat steps 5–8 k times.
10. Update the control parameter $T = \alpha T$.
11. Repeat steps 5–10 until convergence is reached.

The probability of acceptance is compared to a random number ρ generated uniformly in the interval [0,1]. If $\rho \leq p(\delta E)$, the solution is accepted. Otherwise, another solution is randomly selected and evaluated.

The temperature and cooling parameter α have to be chosen carefully. Too high a temperature leads to acceptance of almost all inferior solutions and the search might tend toward a random search. If the temperature is too low, almost no inferior solution will be accepted and the search approaches one of the local optimization methods. Commonly, the probability for acceptance of an inferior solution should be higher than 0.5. If no information on the temperature is at hand, the algorithm can be initiated at a lower temperature that is gradually increased until an acceptable probability of inferior solutions is found. Then the cooling process

can be started in a manner comparable to heating a system until it is melted before initiating the cooling process.

The rate of cooling is influenced by the choice of cooling parameter α and the number of iterations (steps 5–8 in Example 8.12). There should be enough iteration allowed to fully explore the neighborhood of a local optimum. Often, the iteration number is allowed to increase in the cooling process.

Another approach is to combine the product of Boltzmann constant and temperature in a new constant $c = kT$. To reduce the value for control parameter, c, in the next series of movements, the interval $[0, c_0]$ could be divided into a fixed number of K subintervals and c determined by

$$c_k = \frac{K-k}{K} c_0 \quad \text{for } k = 1, \ldots, K \tag{8.61}$$

Initial applications of SA were related to combinatorial optimization problems, for example, wavelength selection in multicomponent analysis (cf. "Applications for Multicomponent Analysis" Section). However, continuous functions can also be optimized. Thus, the parameters in a nonlinear model can be found by SA if the objective function is defined in analogy to the χ^2 criterion in nonlinear regression analysis (Eq. (6.110)).

One of the drawbacks in applying SA is its inability to compete with other optimization strategies that have been distinctively formulated for unique problems. The efficiency might be limited. On the other hand, SA is very easy to implement.

Tabu Search

Another strategy to solve optimization problems with many local optima is tabu search (TS). TS searches an optimum and the location of the optimum is remembered. In future searches, this optimum is avoided.

The algorithm starts by computing solutions in the neighborhood of a selected candidate solution. A neighborhood solution is generated by flipping one bit in the best preceding bit string. If there are no improving neighbored solutions TS accepts the mildest detrimental step and allows, in addition, variables to be added and removed. To avoid an immediate revisiting of a local optimum after leaving it with a detrimental step, the complement of this move must be set tabu. All moves set tabu are listed in a tabu list. The decision on how many and which moves have to be set tabu within any iteration is carried out on the basis of different strategies.

In the *static* method, moves are set tabu as soon as their complements (inverse moves) have been selected and stay tabu for a fixed number of iterations. Thereafter, these moves are reactivated, for example, after 10 iterations. A possible algorithm is given in Example 8.13. Although this method might work well, there is still the danger that the search cycles around the same solutions in a fixed sequence.

Example 8.13 *Tabu search algorithm* [24]

Given a feasible solution x^* with the objective function value z^*, let $x = x^*$ with $z = z^*$.

Iteration

while the stopping criterion is not fulfilled **do** the following:

1. Select the best admissible move that transforms x into x^* with objective function value $z(x')$ and add its attributes to the running list.
2. Perform tabu list management: compute moves (or attributes) to be set tabu and add those to the tabu list.
3. Update solutions: $x = x'$, $z(x) = z(x')$; **if** $z(x) < z^*$ then $z^* = z(x)$, $x^* = x$ **endif**

endwhile

Result: x^* is the best of all determined solutions, with objective function value z^*.

The *dynamic* method for TS circumvents this problem by varying the tabu status according to the moves considered in a way that rigorously excludes cycling behavior. An efficient dynamic TS method is the reverse elimination method. This algorithm is based on the idea that a given solution can be used again only if it lies in the neighborhood of the actual solution. The list of already worked steps is reexamined in order to find all steps that were already set tabu and, therefore, led to a successful solution. For this purpose, a residual cancelation sequence is built up for each tracing step. In the beginning, the residual cancelation sequence is empty. In the following, only those moves are added whose complements are not in the sequence. At each tracing step, it is then known which moves have to be reversed in order to turn the actual solution back into one examined at an earlier iteration step. Example 8.14 demonstrates this reverse elimination approach.

Example 8.14 *Dynamic tabu search by the reverse elimination method*

A solution can only be revisited in the next iteration if it is a neighbor $N(x)$ of the current solution. Here we consider the selection of wavelength in a multivariate calibration problem (cf. Section 6.2):

Bit string for five wavelengths:	[1 1 0 0 1 0]
Move 1:	[0 0 0 0 1 0]
Move 6:	[0 0 0 0 1 1]

Running list: $\underline{1}\ \underline{6}\ 7\ 3\ 1\ \underline{5}\ 6\ 4\ \underline{6}\ 5$ (latest move: 5, complementary (inverse) moves are underlined)

Reverse elimination method:

Iteration	Tracing step	Residual cancelation sequence	Length	Tabu move
10	1	5	1	$\underline{5}$
9	2	$\underline{6}\ 5$	2	—
8	3	$4\ \underline{6}\ 5$	3	—
7	4	4 5	2	—
6	5	4	1	$\underline{4}$
5	6	1 4	2	—
4	7	3 1 4	3	—
3	8	7 3 1 4	4	—
2	9	$\underline{6}\ 7\ 3\ 1\ 4$	5	—
1	10	$\underline{6}\ 7\ 3\ 4$	4	—

Tracing back the running list, there exists a residual cancelation sequence with a single element (move 4), that is, the corresponding complement move $\underline{4}$ will be set tabu and the running of the algorithm is continued.

First applications of TS within Chemometrics are known for optimization of quantitative structure–activity relationship models and for multivariate spectrometric calibration.

General Reading

Section 8.1 Artificial Intelligence and Expert Systems

1. Cherniak, E. and McDermott, D. (1985) *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, MA.
2. Winston, P.H. and Horn, B.K.P. (1984) *Lisp*, *2nd edn* edn, Addison-Wesley, Reading, MA.

3. Lindsay, R.K. *et al.* (1980) *Applications of Artificial Intelligence for Organic Chemistry – the Dendral Project*, McGraw-Hill, New York.
4. Gunasingham, H., Srinivasan, B., and Ananda, A.L. (1986) *Anal. Chim. Acta*, **182**, 193.
5. Buydens, L.M.C. and Schoenmakers, P.J. (eds) (1993) *Intelligent Software for Chemical Analysis*, Elsevier, Amsterdam.
6. Peichang, L. and Hongxin, H. (1988) *J. Chromatogr.*, **452**, 175.
7. Janssens, K. and Van Espen, P. (1986) *Anal. Chim. Acta*, **184**, 117.
8. Arnold, T., Otto, M., and Wegscheider, W. (1994) *Talanta*, **41**, 1169.
9. Wünsch, G. and Gansen, M. (1989) *Anal. Chim. Acta*, **333**, 607.
10. Wagner, A., Flock, J., and Otto, M. (1995) *Stahl Eisen*, **115**, 57.

Section 8.2 Neural Networks

11. Rumelhart, D.E. and McClelland, J.L. (1986) *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, Foundations, Vol. 2, Psychological and Biological Models, vol. 1, MIT Press, Cambridge, MA.
12. Kohonen, T. (1984) *Self-Organization and Associative Memory*, Springer, Berlin.
13. Kosko, B. (1992) *Neural Networks and Fuzzy Systems*, Prentice–Hall, London.
14. Borggaard, C. and Thodberg, H.H. (1992) *Anal. Chem.*, **64**, 545.
15. Zupan, J. and Gasteiger, J. (1999) *Neural Networks for Chemists*, 2nd edn, Wiley-VCH Verlag GmbH, Weinheim.
16. Curry, B. and Rumelhart, D.E. (1990) *Tetrahedron Comput. Method.*, **3**, 213.
17. Mittermayr, C.R., Drouen, A.C.J.H., Otto, M., and Grasserbauer, M. (1994) *Anal. Chim. Acta*, **294**, 227.
18. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Comprehensive Chemometrics – Chemical and Biochemical Data Analysis*, vol. **3**, Chapter 3.14, Elsevier, Amsterdam.

Section 8.3 Fuzzy Theory

19. Bandemer, H. and Näther, W. (1992) *Fuzzy Data Analysis, Theory and Decision Library, Series B: Mathematical and Statistical Methods*, Kluwer, Dordrecht.
20. Otto, M. (1986) *Chemom. Intell. Lab. Syst.*, **1**, 71.
21. Bezdek, J.C. (1982) *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

Section 8.4 Genetic Algorithms and Other Global Search Strategies

22. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
23. Kalivas, J.H. (ed.) (1995) *Adaption of Simulated Annealing to Chemical Optimization Problems*, Elsevier, Amsterdam.
24. Glover, F. (1989) *Tabu search – Part I. ORSA J. Comput.*, **1**, 190; *Tabu search – Part II*, *ORSA J. Comput.* (1989) **2** 4.
25. Brown, S.D., Tauler, R., and Walczak, B. (eds) (2009) *Compre hensive Chemometrics – Chemical and Biochemical Data Analysis*, Vol. **1**, Chapter 1.20 and Vol. **3**, Chapter 3.22, Elsevier, Amsterdam.

Questions and Problems

1. What are the differences between PROLOG, LISP, and an expert system shell?
2. Discuss the difference between search by the simplex method and that by an in-depth strategy.
3. Summarize typical signal transfer functions used with artificial neural networks.
4. How would we arrange the neurons in a multilayer perceptron to classify gas chromatograms consisting of 20 peaks and belonging to four different classes?
5. What makes the difference between an associative and competitive learning law?
6. A bidirectional associative memory (BAM) is to be trained for the following three bipolar associations:

$$\begin{aligned} \boldsymbol{x}_1 &= (1\ -1\ 1\ 1), & \boldsymbol{y}_1 &= (1\ -1\ 1) \\ \boldsymbol{x}_2 &= (-1\ -1\ 1\ -1), & \boldsymbol{y}_1 &= (-1\ -1\ 1) \\ \boldsymbol{x}_3 &= (1\ 1\ -1\ 1), & \boldsymbol{y}_1 &= (1\ 1\ -1) \end{aligned}$$

   Calculate the weight matrix according to Eq. (8.7) for those associations. What is the value for $\boldsymbol{y}$, if a new $\boldsymbol{x}$-vector, $\boldsymbol{x} = (-1\ 1\ -1\ 1)$, is presented to the BAM and bipolar thresholding is operative?
7. Which neural network preserves the topological structure of data?
8. What is the difference between probability and possibility theory, and which methods represent the two particular approaches?
9. Repeat the set-theoretic operations for union, intersection, complement, and cardinality for the crisp sets A and B as given: $A = (2\ 1\ 5\ 3\ 7)$, $B = (1\ 4\ 6\ 2)$.
10. Compare symbolic knowledge processing in case of crisp and fuzzy-rule-based systems.
11. How are genetic algorithms used for feature selection in the framework of pattern recognition?
12. Calculate the y-value according to Eq. (8.49) for a binary coded chromosome [0 1 1 0 1 1 1]. What value would result if the vigesimal system was used?
13. Compare the optimization strategies of genetic algorithms, simulated annealing, and tabu search with those of local optimization methods.

9
Quality Assurance and Good Laboratory Practice

Learning Objectives

- To introduce the tools available to ensure the quality of analytical–chemical measurement
- To learn about regulatory and legal aspects of quality assurance and quality control.

Analytical data need to be comparable between different laboratories on an international scale. This presupposes that the quality of the data is assured. In addition, planning, performance, reporting, and archiving of tests should be regulated.

A central point of comparability of analytical data is an appropriate system of quality assurance. According to DIN 55350, *quality* is defined by

> *The total of properties and features of a product or an operation to fulfill predefined requirements.*

It therefore becomes necessary to quantify quality in the frame of *quality assurance* tests. Quality assurance comprises all activities that lead to fulfillment of the defined requirements. They include the totality of operations in quality management, quality planning, quality directing, and quality tests.

If the requirement cannot be fulfilled, then an *error* is made according to the DIN directive. If the error restricts the applicability, it is termed a *lack*. Typical in analysis are random and systematic errors, false-positive or false-negative detection results, or the complete failure of an analytical determination.

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

To control errors in an analysis, the individual steps of the procedure must be fixed in detail. In addition, the analytical procedure has to be tested for its performance, that is, it must be *validated.* For the use of procedures in routine analysis, additional checks are necessary that will be explained as follows.

9.1 Validation and Quality Control

Validation criteria are as follows:
Trueness
Precision
Dynamic range
Selectivity
Limit of detection
Limit of determination
Robustness.

Validation in Analytics

Validation describes in general the assurance that an analytical procedure provides reproducible and secure results that are required for the application intended.

The first step in developing a relative analytical method is *calibration.* It is based on the use of standard solutions or of a solid standard. The calibration function (Eq. (4.1)) is constructed by means of linear regression analysis as discussed in Section 6.1.

The limit of detection is that analyte concentration that corresponds to the averaged signal of the blank plus three times the standard deviation.

The precision of the procedure can be characterized by the *standard deviation of the procedure.* As additional performance characteristics, the *limit of detection* (Eq. (4.3)) and *working range* are reported.

To test for systematic deviations, explained by influences due to different steps of an analytical procedure or by the sample matrix, the recovery rate is calculated (cf. Eq. (4.6)). To investigate an analytical procedure for systematic deviations, the recovery function is applied.

Recovery Function

The *limit of determination* is the lowest analyte concentration that can be determined with an acceptable accuracy.

The recovery function describes the relationship between the found, x_{found}, and true, x_{true}, concentrations by means of a straight-line model:

$$x_{found} = a_0 + a_1 x_{true} \tag{9.1}$$

The *working range* of an analytical method denotes the range between the lower and upper concentration, for which accurate determinations are feasible.

where a_0 and a_1 are the regression parameters. In an ideal case, the recovery function should pass through the origin of the coordinate system and have a slope of 1, that is,

$$a_0 = 0 \text{ and } a_1 = 1.0 \tag{9.2}$$

In practice, these conditions will only be approximately valid. The test for significance of the deviations can be carried out by

means of the confidence intervals for the parameter a_0 and a_1 on the basis of a t test (cf. Eq. (6.47)). The confidence intervals for the parameters are as follows:

$$\Delta a_0 = a_0 \pm t(1-\alpha/2;f)s_{a_0} \tag{9.3}$$

$$\Delta a_1 = a_1 \pm t(1-\alpha/2;f)s_{a_1} \tag{9.4}$$

Accuracy stands for both trueness and precision.

where t is the quantile of the Student distribution (Table A.3), α is the significance level, and f is the degree of freedom.

The standard deviations for the parameters, s_{a_0} and s_{a_1}, are calculated according to Eqs. (6.7) and (6.8).

A constant systematic deviation is given at a significance level, α, if the confidence interval, Δa_0, does not include the value $a_0 = 0$. In the case that the confidence interval Δa_1 does not contain the value $a_1 = 1$, a proportional systematic deviation holds.

Finally, the *robustness* of analytical procedures must be investigated, to give assurance that the quality of data is independent of small variations during the performance of the procedure.

Robustness can be checked by round-robin tests, the performance of which will be discussed later in this section. Within the single laboratory, robustness can be checked by variation of the experimental variables within tolerable limits.

The elaborated analytical procedure is the basis for quality assurance in routine analysis. It is summarized in a *standard operating procedure* (SOP). The SOP defines the range of application and the quality goals.

The standard deviation of a procedure is calculated by Eq. (6.9):

$$s_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}}$$

Based on the concentration, x, we obtain

$$s_x = \frac{s_y}{b_1}$$

Internal Quality Assurance

Quality control of a procedure in routine analysis is based primarily on the evaluation of quantities that characterize precision and reliability, such as the mean, standard deviation, dynamic range, recovery rate, and reliability ranges (dispersion and confidence interval).

For internal quality assurance, *control samples* are applied as

- standard solutions
- blank samples
- real samples
- synthetic samples
- certified standard reference materials.

The control samples should be analyzed in each series of analyses at least once or twice, to control the accuracy of measurements.

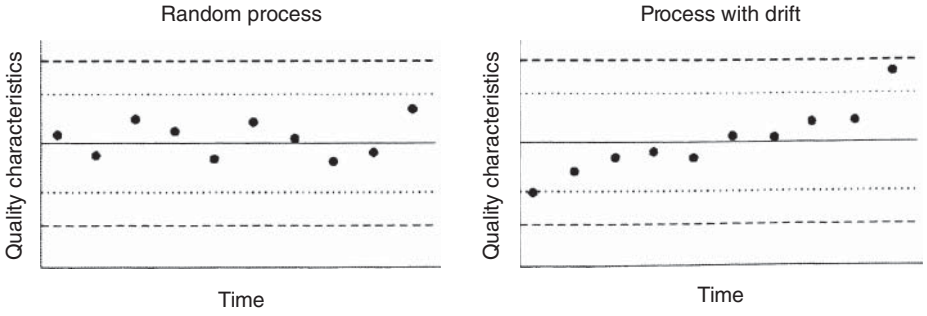


Figure 9.1 Sequence of objective quantities in a control chart. The *dotted lines* characterize the lower and upper warning limits. The *broken lines* describe the lower and upper control limits.

Control Charts

To monitor test procedures or to ensure the quality of products and processes by using analytical methods, quality control charts have proven their success. There, a quality characteristic is recorded at given spacing in a chart enabling ready recognition of typical situations (Figure 9.1).

As quality characteristics, the reference value and the control limits are used. We distinguish according to the kind of the quality characteristic individual value charts, mean charts, such as the $\overline{x}$ chart, median or blank charts, dispersion charts, such as standard deviation and range charts, and the recovery rate chart. Calculation of the objective criterion plotted as a central line and its thresholds is given in Table 9.1 for important control charts. The limits for the $\overline{x}$ chart are evaluated on the basis of a t distribution. In a standard deviation chart, the thresholds are defined by applying the χ^2 distribution. The χ^2 quantiles can be taken from Table A.5. The range charts are based on the ranges between the largest and smallest observations within a subgroup i ($x_{i,\max}$ and $x_{i,\min}$). The upper and lower limits result from multiplication with the D factors, which are given for 95% and 99% of statistical certainty in Table 9.2.

The significance level of 5% serves as a warning limit. A single crossing of the warning limit requires only an increased attention for controlling the process. The control limit is fixed at a significance limit of 1%. If a value exceeds the control limit, immediate action will be necessary. This is termed the *outer-control situation*.

The five line control charts have been introduced by Walter *Shewhart* and are also termed *Shewhart means* and *Shewhart range charts*.

Description of a quality assurance system is given in a *quality assurance handbook,* wherein all structures, responsibilities, SOPs, and tools for realization of the quality assurance are summarized.

Table 9.1 Control quantities in quality control chart.

Objective criterion	Calculation of objective criterion	Lower limit	Upper limit
Mean, $\overline{x}$	$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	$\overline{x} - t(P,f)\frac{s}{\sqrt{n}}$	$\overline{x} + t(P,f)\frac{s}{\sqrt{n}}$
Mean standard deviation, s_m	$s_m = \sqrt{\frac{\sum_{i=1}^{N}(n_i - 1)s_i^2}{\sum_{i=1}^{N}(n_i - 1)}}$	$s_m\sqrt{\frac{1}{n-1}\chi^2\left(n-1;\frac{\alpha}{2}\right)}$	$s_m\sqrt{\frac{1}{n-1}\chi^2\left(n-1;1-\frac{\alpha}{2}\right)}$
Range, $\overline{R}$	$\overline{R} = \frac{\sum_{i=1}^{N}(x_{i,\max} - x_{i,\min})}{N}$	$D_{\text{lower}}\overline{R}$	$D_{u\text{pper}}\overline{V}$

N Number of subgroups, n number of repetitive measurements per subgroup (n_i), x_i observation, s_i standard deviation from n_i parallel determinations, t Student factor, P probability, f degrees of freedom.

Table 9.2 *D* factors for calculation of the limits of the range chart for the probabilities *P* of 95% and 99%.

	$P = 95\%$ or $\alpha = 5\%$		$P = 99\%$ or $\alpha = 1\%$	
n	D_{lower}	D_{upper}	D_{lower}	D_{upper}
2	0.039	2.809	0.008	3.518
3	0.179	2.176	0.080	3.518
4	0.289	1.935	0.166	2.614
5	0.365	1.804	0.239	2.280
6	0.421	1.721	0.296	2.100
7	0.462	1.662	0.341	1.986
8	0.495	1.617	0.378	1.906
9	0.522	1.583	0.408	1.846
10	0.544	1.555	0.434	1.798

External Quality Assurance

Laboratory Intercomparison Studies

To ensure the comparability of analytical results, round-robin tests are performed. The aims are as follows:

- Standardization of analytical procedures
- Control of the analyses of a laboratory
- Preparation of certified reference material.

According to DIN 38402, at least 8 laboratories should participate in a study or still better more than 15 laboratories. These laboratories typically perform four parallel determinations each. The precision of the individual laboratories can be evaluated from the parallel determinations on the basis of the standard deviations. To judge the trueness, the recovery is determined according to Eq. (4.6). The laboratory means are then related to a true value or to the total mean.

Traceability

The results of chemical analyses should be comparable among each other, just as it is feasible with physical quantities: the length of an object is exactly given by meters or the mass by kilograms.

The basis for comparability of chemical analyses is the relation to standard reference materials of which the contents or concentrations are exactly known. The elements and compounds contained in a sample are preferably traced back to the mol. Problems in traceability of chemical analyses frequently result from the limited selectivity of analytical methods. Determination of the mass

of a pure substance can be done in principle without fundamental errors, but this is not *a priori* true for determination of the concentration of a single substance in a complex matrix, such as blood. Validation of analytical results is therefore often more difficult than evaluating physical quantities.

9.2 Accreditation and Good Laboratory Practice

Accreditation of Laboratories

An analytical laboratory proves its effective quality assurance system by *accreditation.* Accreditation attests to the competence of a laboratory for performing given analytical methods versus an independent customer. In different countries, different accreditation systems have emerged. For example, in Germany, there is no central accreditation system, but a system that is decentralized to different sectors.

The Commission of the European Union has harmonized the national accreditation systems. Unique criteria were developed for the operation of the test laboratories and for their accreditation and certification. The result of those harmonization policies is the series of directives Euronorm EN 45000 (Table 9.3). At present, all accreditations are performed on the basis of EN 45000.

Basics of Good Laboratory Practice

What is to be understood by the system of good laboratory practice (GLP), which has existed for more than 20 years? GLP can be traced back to irregularities detected in the beginning of the 1970s by the American Food and Drug Administration (FDA) during revision of toxicological studies. The results were regulations for GLP for testing in toxicology. They had to be adhered

Table 9.3 General criteria in the directive series EN 45000.

Euronorm	Topic
EN 45001	Running test laboratories
EN 45002	Evaluation of test laboratories
EN 45003	Organization of accreditation offices
EN 45011	Certification of products
EN 45012	Certification of personnel

to not only by the companies in the United States, but also by all countries exporting to the United States. The Organization For Economic Cooperation And Development (OECD) took over the internationalization of those standards.

GLP is mainly concerned with assuring the repeatability of investigations. The assurance of quality is guaranteed in the system by an additional quality assurance unit that is controlled by continuous inspections to maintain the principles of GLP.

The workload caused by the formalism and archiving is much larger compared than that attributable to the EN 45000 directives. The GLP system is applied in areas covered by legislation, for example, in the case of the introduction of new chemicals, in toxicology, or in health systems.

General Reading

1. Burgess, C. (2004) Analytical quality management, in *Analytical Chemistry*, 2nd edn (eds R. Kellner, J.-M. Mermet, M. Otto, M. Valcárcel, and H.M. Widmer), Wiley-VCH Verlag GmbH, Weinheim.
2. Günzler, H. (1996) *Accreditation and Quality Assurance in Analytical Chemistry*, Springer, Berlin.
3. DeBièvre, P. and Günzler, H. (eds) (2003) *Traceability in Chemical Measurement*, Springer, Berlin.
4. Hibbert, D.B. (2007) *Quality Assurance for the Analytical Chemistry Laboratory*, Oxford University Press, New York.

Questions and Problems

1. Define the analytical performance characteristics precision, trueness, accuracy, selectivity, dynamic range, working range, recovery, robustness, detection limit, and limit of determination.
2. What are reference materials used for?
3. What are control charts used for and which measures are common?
4. What is the meaning of "traceability to the mol?"
5. What quality assurance system is applied in GLP?

Appendix

Statistical Distributions

Tables A.1 – A.6.

Digital Filters

Tables A.7 and A.8.

Experimental Designs

Tables A.9 – A.12.

Matrix Algebra

A point in n-dimensional space R^n is represented by a *vector,* that is,

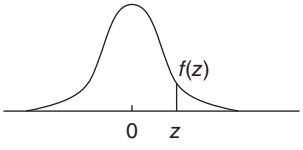
$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ or in transposed form } \quad \boldsymbol{x}^{\mathrm{T}} = (x_1, x_2, \ldots, x_n)$$

The sum of two vectors $\boldsymbol{x}, \boldsymbol{y} \in R^n$ reveals

$$\boldsymbol{x} + \boldsymbol{y} = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix} \quad \text{example}: \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 4 \\ 7 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \\ 10 \end{pmatrix}$$

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.

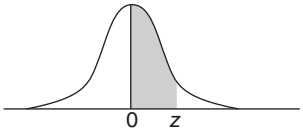
Table A.1 Probability density function (ordinate values) of the standardized normal distribution.



z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.3989	0.3989	0.3989	0.3988	0.3986	0.3984	0.3982	0.3980	0.3977	0.3973
0.1	0.3970	0.3965	0.3961	0.3956	0.3951	0.3945	0.3939	0.3932	0.3925	0.3918
0.2	0.3910	0.3902	0.3894	0.3885	0.3876	0.3867	0.3857	0.3847	0.3836	0.3825
0.3	0.3814	0.3802	0.3790	0.3778	0.3765	0.3752	0.3739	0.3726	0.3712	0.3697
0.4	0.3683	0.3668	0.3653	0.3637	0.3621	0.3605	0.3589	0.3572	0.3555	0.3538
0.5	0.3521	0.3503	0.3485	0.3467	0.3448	0.3429	0.3411	0.3391	0.3372	0.3352
0.6	0.3332	0.3312	0.3292	0.3271	0.3251	0.3230	0.3209	0.3187	0.3166	0.3144
0.7	0.3132	0.3101	0.3079	0.3056	0.3034	0.3011	0.2989	0.2966	0.2943	0.2920
0.8	0.2897	0.2874	0.2850	0.2827	0.2803	0.2779	0.2756	0.2732	0.2709	0.2685
0.9	0.2661	0.2637	0.2613	0.2589	0.2565	0.2541	0.2516	0.2492	0.2468	0.2444
1.0	0.2420	0.2396	0.2371	0.2347	0.2323	0.2299	0.2275	0.2251	0.2227	0.2203
1.1	0.2179	0.2155	0.2131	0.2107	0.2083	0.2059	0.2036	0.2012	0.1989	0.1965
1.2	0.1942	0.1919	0.1895	0.1872	0.1849	0.1827	0.1804	0.1781	0.1759	0.1736
1.3	0.1714	0.1691	0.1669	0.1647	0.1626	0.1604	0.1582	0.1561	0.1539	0.1518
1.4	0.1497	0.1476	0.1456	0.1435	0.1415	0.1394	0.1374	0.1354	0.1334	0.1315
1.5	0.1295	0.1276	0.1257	0.1238	0.1219	0.1205	0.1182	0.1163	0.1445	0.1127
1.6	0.1109	0.1092	0.1074	0.1057	0.1040	0.1223	0.1006	0.09892	0.08728	0.09566
1.7	0.09405	0.09246	0.09089	0.08933	0.08780	0.08628	0.08478	0.08329	0.08183	0.08038
1.8	0.07895	0.07754	0.07614	0.07477	0.07341	0.0721	0.07074	0.06943	0.06814	0.06687
1.9	0.06562	0.06438	0.06316	0.06195	0.06077	0.0596	0.05844	0.05730	0.05618	0.05508
2.0	0.05399	0.05292	0.05186	0.05082	0.04980	0.0488	0.04780	0.04632	0.04586	0.04491
2.1	0.04398	0.04307	0.04217	0.04128	0.04041	0.03955	0.03871	0.03788	0.03706	0.03626
2.2	0.03547	0.03470	0.03394	0.03319	0.03246	0.03174	0*l*.03103	0.03034	0.02965	0.02898
2.3	0.02833	0.02768	0.02705	0.02643	0.02582	0.02522	0.02463	0.02406	0.02349	0.02294
2.4	0.02239	0.02186	0.02134	0.02083	0.02033	0.01984	0.01936	0.01889	0.01842	0.01797
2.5	0.01753	0.01709	0.01667	0.01625	0.01585	0.01545	0.01506	0.01468	0.01431	0.01394
2.6	0.01358	0.01323	0.01289	0.01256	0.01223	0.01191	0.01160	0.01130	0.01100	0.01071
2.7	0.01042	0.1014	0.009871	0.009606	0.009347	0.00909	0.00885	0.00861	0.00837	0.00814
2.8	0.00792	0.00770	0.007483	0.007274	0.007071	0.00687	0.00668	0.00649	0.00631	0.00613
2.9	0.00595	0.00578	0.005616	0.005454	0.005296	0.00514	0.00499	0.00485	0.00471	0.00457
3.0	0.00443									

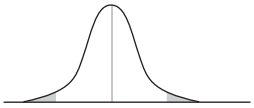
Example: $f(0.42) = 0.3653$.

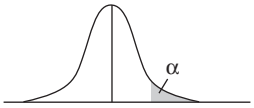
Table A.2 Areas for the standard normal variate z (Eq. (2.28)) of the normal distribution.



z	**0.00**	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**
0.0	0.0000	0.0040	0.080	0.0120	0.0160	0.0199	0.0239	0.0279	0.0319	0.0359
0.1	0.0398	0.0438	0.0478	0.0517	0.0557	0.0596	0.0636	0.0675	0.0714	0.0753
0.2	0.0793	0.0832	0.0871	0.0910	0.0948	0.0987	0.1026	0.1064	0.1103	0.1141
0.3	0.1179	0.1217	0.1255	0.1293	0.1331	0.1368	0.1406	0.1443	0.1480	0.1517
0.4	0.1554	0.1591	0.1628	0.1664	0.1700	0.1736	0.1772	0.1808	0.1844	0.1879
0.5	0.1915	0.1950	0.1985	0.2019	0.2054	0.2088	0.2123	0.2157	0.2190	0.2224
0.6	0.2257	0.2291	0.2324	0.2357	0.2389	0.2422	0.2454	0.2486	0.2518	0.2549
0.7	0.2580	0.2612	0.2642	0.2673	0.2704	0.2734	0.2764	0.2794	0.2823	0.2852
0.8	0.2881	0.2910	0.2939	0.2967	0.2995	0.3023	0.3051	0.3078	0.3106	0.3133
0.9	0.3159	0.3186	0.3212	0.3238	0.3264	0.3289	0.3315	0.3340	0.3365	0.3389
1.0	0.3413	0.3438	0.3461	0.3485	0.3508	0.3531	0.3554	0.3577	0.3599	0.3621
1.1	0.3643	0.3665	0.3686	0.3708	0.3729	0.3749	0.3770	0.3790	0.3810	0.3830
1.2	0.3849	0.3869	0.3888	0.3907	0.3925	0.3944	0.3962	0.3980	0.3997	0.4015
1.3	0.4032	0.4049	0.4066	0.4082	0.4099	0.4115	0.4131	0.4147	0.4162	0.4177
1.4	0.4192	0.4207	0.4222	0.4236	0.4251	0.4265	0.4279	0.4292	0.4306	0.4319
1.5	0.4332	0.4345	0.4357	0.4370	0.4382	0.4394	0.4406	0.4418	0.4429	0.4441
1.6	0.4452	0.4463	0.4474	0.4484	0.4495	0.4505	0.4515	0.4525	0.4535	0.4545
1.7	0.4554	0.4564	0.4573	0.4582	0.4591	0.4599	0.4608	0.4616	0.4625	0.4633
1.8	0.4641	0.4649	0.4656	0.4664	0.4671	0.4678	0.4686	0.4693	0.4699	0.4706
1.9	0.4713	0.4719	0.4726	0.4732	0.4738	0.4744	0.4750	0.4756	0.4761	0.4767
2.0	0.4772	0.4778	0.4783	0.4788	0.4793	0.4798	0.4803	0.4808	0.4812	0.4817
2.1	0.4821	0.4826	0.4830	0.4834	0.4838	0.4842	0.4846	0.4850	0.4854	0.4857
2.2	0.4861	0.4864	0.4868	0.4871	0.4875	0.4878	0.4881	0.4884	0.4887	0.4890
2.3	0.4893	0.4896	0.4898	0.4901	0.4904	0.4906	0.4909	0.4911	0.4913	0.4916
2.4	0.4918	0.4920	0.4922	0.4925	0.4927	0.4929	0.4931	0.4932	0.4934	0.4936
2.5	0.4938	0.4940	0.4941	0.4943	0.4945	0.4946	0.4948	0.4949	0.4951	0.4952
2.6	0.4953	0.4955	0.4956	0.4957	0.4959	0.4960	0.4961	0.4962	0.4963	0.4964
2.7	0.4965	0.4966	0.4967	0.4968	0.4969	0.4970	0.4971	0.4972	0.4973	0.4974
2.8	0.4974	0.4975	0.4976	0.4977	0.4977	0.4978	0.4979	0.4979	0.4980	0.4981
2.9	0.4981	0.4982	0.4982	0.4983	0.4984	0.4984	0.4985	0.4985	0.4986	0.4986
3.0	0.49865	0.4987	0.4987	0.4988	0.4988	0.4989	0.4989	0.4989	0.4990	0.4990
4.0	0.49997									

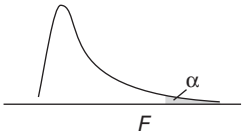
Table A.3 Two- and one-sided Student's *t*-distribution for different risk levels α and the degrees of freedom from $f = 1$ to $f = 20$.





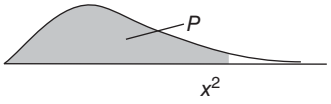
Two-sided Student's *t*-distribution			One-sided Student's *t*-distribution		
f	$\alpha = 0.05$	$\alpha = 0.01$	***f***	$\alpha = 0.05$	$\alpha = 0.025$
1	12.706	63.657	1	6.314	12.706
2	4.303	9.925	2	2.920	4.303
3	3.182	5.841	3	2.353	3.182
4	2.776	4.604	4	2.132	2.776
5	2.571	4.032	5	2.015	2.574
6	2.447	3.707	6	1.943	2.447
7	2.365	3.499	7	1.895	2.365
8	2.306	3.355	8	1.860	2.306
9	2.262	3.250	9	1.833	2.262
10	2.228	3.169	10	1.812	2.228
11	2.201	3.106	11	1.796	2.201
12	2.179	3.055	12	1.782	2.179
13	2.160	3.012	13	1.771	2.160
14	2.145	2.977	14	1.761	2.145
15	2.131	2.947	15	1.753	2.131
16	2.120	2.921	16	1.746	2.120
17	2.110	2.898	17	1.740	2.110
18	2.101	2.878	18	1.734	2.101
19	2.093	2.861	19	1.729	2.093
20	2.086	2.845	20	1.725	2.086
21	2.080	2.831	21	1.721	2.080
22	2.074	2.819	22	1.717	2.074
23	2.069	2.807	23	1.714	2.069
24	2.064	2.797	24	1.711	2.064
25	2.060	2.787	25	1.708	2.060
26	2.056	2.779	26	1.706	2.056
27	2.052	2.771	27	1.703	2.052
28	2.048	2.763	28	1.701	2.048
29	2.045	2.756	29	1.699	2.045
30	2.042	2.750	30	1.697	2.042
40	2.021	2.704	40	1.684	2.021
50	2.009	2.678	50	1.676	2.009
60	2.000	2.660	60	1.671	2.000
70	1.994	2.648	70	1.667	1.994
80	1.990	2.639	80	1.664	1.990
90	1.987	2.632	90	1.662	1.987
100	1.984	2.626	100	1.660	1.984
∞	1.960	2.576	∞	1.645	1.960

Table A.4 ***F***-distribution for the risk levels $\alpha = 0.025$ (*lightface type*) and $\alpha = 0.01$ (*boldface type*) at the degrees of freedom f_1 and f_2.



f_2	$f_1 = 1$	2	3	4	5	6	7	8	9	10	12	20	50	∝
1	647	779	864	899	922	937	948	956	963	968	976	993	1008	1018
	4052	**4999**	**5403**	**5625**	**5764**	**5859**	**5928**	**5981**	**6022**	**6056**	**6106**	**6208**	**6302**	**6366**
2	38.51	39.00	39.17	39.25	39.30	39.33	39.36	39.37	39.39	39.40	39.41	39.54	39.48	39.50
	98.49	**99.00**	**99.17**	**99.25**	**99.30**	**99.33**	**99.36**	**99.37**	**99.39**	**99.40**	**99.42**	**99.45**	**99.48**	**99.50**
3	17.44	16.04	15.44	15.10	14.88	14.73	14.62	14.54	14.47	14.42	14.34	14.17	14.01	13.90
	34.12	**30.82**	**29.46**	**28.71**	**28.24**	**27.91**	**27.67**	**27.49**	**27.34**	**27.23**	**27.05**	**26.65**	**26.35**	**26.12**
4	12.22	10.65	9.98	9.60	9.36	9.20	9.07	8.98	8.90	8.84	8.75	8.56	8.38	8.26
	21.20	**18.00**	**16.69**	**15.98**	**15.52**	**15.21**	**14.98**	**14.80**	**14.66**	**14.54**	**14.37**	**14.02**	**13.69**	**13.46**
5	10.01	8.43	7.76	7.39	7.15	6.98	6.85	6.76	6.68	6.62	6.52	6.33	6.14	6.02
	16.26	**13.27**	**12.06**	**11.39**	**10.97**	**10.67**	**10.45**	**10.29**	**10.15**	**10.05**	**9.89**	**9.55**	**9.24**	**9.02**
6	8.81	7.26	6.60	6.23	5.99	5.82	5.70	5.60	5.52	4.06	5.46	5.37	4.98	4.85
	13.74	**10.92**	**9.78**	**9.15**	**8.75**	**8.47**	**8.26**	**8.10**	**7.98**	**7.87**	**7.72**	**7.40**	**7.10**	**6.88**
7	8.07	6.54	5.89	5.52	5.29	5.12	4.99	4.90	4.82	4.76	4.67	4.47	4.28	4.14
	12.25	**9.55**	**8.45**	**7.85**	**7.46**	**7.19**	**7.00**	**6.84**	**6.71**	**6.62**	**6.47**	**6.16**	**5.87**	**5.65**
8	7.57	6.06	5.42	5.05	4.82	4.65	4.53	4.43	4.36	4.29	4.20	4.00	3.81	3.67
	11.26	**8.65**	**7.59**	**7.01**	**6.63**	**6.37**	**6.19**	**6.03**	**5.91**	**5.82**	**5.67**	**5.36**	**5.08**	**4.86**
9	7.21	5.71	5.08	4.72	4.84	4.32	4.20	4.10	4.03	3.96	3.87	3.67	3.47	3.33
	10.56	**8.02**	**6.99**	**6.42**	**6.06**	**5.80**	**5.62**	**5.47**	**5.35**	**5.26**	**5.11**	**4.81**	**4.53**	**4.31**
10	6.94	5.46	4.83	4.47	4.24	4.07	3.95	3.85	3.78	3.72	3.62	3.42	3.22	3.08
	10.04	**7.56**	**6.55**	**5.99**	**5.64**	**5.39**	**5.21**	**5.06**	**4.95**	**4.85**	**4.71**	**4.41**	**4.13**	**3.91**
12	6.55	5.10	4.47	4.12	3.89	3.73	3.61	3.51	3.44	3.37	3.28	3.07	2.87	2.72
	9.33	**6.93**	**5.95**	**5.41**	**5.06**	**4.82**	**4.65**	**4.50**	**4.39**	**4.30**	**4.16**	**3.86**	**3.58**	**3.36**
20	5.87	4.46	3.86	3.51	3.29	3.13	3.01	2.91	2.84	2.77	2.68	2.46	2.25	1.84
	8.10	**5.85**	**4.94**	**4.43**	**4.10**	**3.87**	**3.70**	**3.56**	**3.46**	**3.37**	**3.23**	**2.94**	**2.65**	**2.42**
50	5.34	3.98	3.39	3.05	2.83	2.67	2.55	2.46	2.38	2.32	2.22	1.99	1.75	1.56
	7.20	**5.08**	**4.22**	**3.74**	**3.45**	**3.21**	**3.04**	**2.91**	**2.81**	**2.72**	**2.58**	**2.29**	**1.98**	**1.70**
∝	5.02	3.69	3.12	2.79	2.57	2.41	2.29	2.19	2.11	2.05	1.94	1.71	1.43	1.00
	6.63	**4.61**	**3.78**	**3.32**	**3.02**	**2.80**	**2.64**	**2.51**	**2.41**	**2.32**	**2.18**	**1.88**	**1.53**	**1.00**

Table A.5 Chi-square distribution for different degrees of freedom f at different probabilities P, $\chi^2(P;f)$.



f	$P = 0.01$	0.05	0.10	0.25	0.50	0.75	0.90	0.95	0.99
1	0.00016	0.0039	0.0158	0.102	0.455	1.32	2.71	3.84	6.63
2	0.0201	0.103	0.211	0.575	1.39	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.21	2.37	4.11	6.25	7.81	11.3
4	0.297	0.711	1.06	1.92	3.36	5.39	7.78	9.49	13.3
5	0.554	1.15	1.61	2.67	4.35	6.63	9.24	11.1	15.1
6	0.872	1.64	2.20	3.45	5.35	7.84	10.6	12.6	16.8
7	1.24	2.17	2.83	4.25	6.35	9.04	12.0	14.1	18.5
8	1.65	2.73	3.49	5.07	7.34	10.2	13.4	15.5	20.1
9	2.09	3.33	4.17	5.90	8.34	11.4	14.7	16.9	21.7
10	2.56	3.94	4.87	6.74	9.34	12.5	16.0	18.3	23.2
11	3.05	4.57	5.58	7.58	10.3	13.7	17.3	19.7	24.7
12	3.57	5.23	6.30	8.44	11.3	14.8	18.5	21.0	26.2
13	4.11	5.89	7.04	9.30	12.3	16.0	19.8	22.4	27.7
14	4.66	6.57	7.79	10.2	13.3	17.1	21.1	23.7	29.1
15	5.23	7.26	8.55	11.0	14.3	18.2	22.3	25.0	30.6
16	5.81	7.95	9.31	11.9	15.3	19.4	23.5	26.3	32.0
17	6.41	8.67	10.1	12.8	16.3	20.5	24.8	27.6	33.4
18	7.01	9.39	10.9	13.7	17.3	21.6	26.0	28.9	34.8
19	7.63	10.1	11.7	14.6	18.3	22.7	27.2	30.1	36.2
20	8.26	10.9	12.4	15.5	19.3	23.8	28.4	31.4	37.6
21	8.90	11.6	13.2	16.3	20.3	24.9	29.6	32.7	38.9
22	9.54	12.3	14.0	17.2	21.3	26.0	30.8	34.0	40.3
23	10.2	13.1	14.8	18.1	22.3	27.1	32.0	35.2	41.6
24	10.9	13.8	15.7	19.0	23.3	28.2	33.2	36.4	43.0
25	11.5	14.6	16.5	19.9	24.3	29.3	34.4	37.7	44.3
26	12.2	15.3	17.2	20.8	25.3	30.4	35.5	38.8	45.6
27	12.8	16.1	18.1	21.7	26.3	31.5	36.7	40.1	46.9
28	13.5	16.9	19.9	22.6	27.3	32.6	37.9	41.3	48.2
29	14.2	17.7	19.7	23.5	28.3	33.7	39.0	42.5	49.5
30	14.9	18.4	20.6	24.4	29.3	34.8	40.2	43.7	50.8
40	22.1	26.5	29.0	33.6	39.3	45.6	51.8	55.7	63.6
50	29.7	34.7	37.6	42.9	49.3	56.3	63.1	67.5	76.1
60	37.4	43.1	46.4	52.2	59.3	66.9	74.4	79.0	88.3
70	45.4	51.7	55.3	61.7	69.3	77.5	85.5	90.5	100.4
80	53.5	60.3	64.2	71.1	79.3	88.1	96.5	101.9	112.3
90	61.7	69.1	73.2	80.6	89.3	98.6	107.6	113.1	124.1
100	70.0	77.9	82.3	90.1	99.3	109.1	118.6	124.3	135.8

Table A.6 Kolmogorov–Smirnov test statistic $d(1-\alpha, n)$ to test for a normal distribution at different significance levels α.

n	0.01	0.05	0.10	0.15	0.20
4	0.417	0.381	0.352	0.319	0.300
5	0.405	0.337	0.315	0.299	0.285
6	0.364	0.319	0.294	0.277	0.265
7	0.348	0.300	0.276	0.258	0.247
8	0.331	0.285	0.261	0.244	0.233
9	0.311	0.271	0.249	0.233	0.223
10	0.294	0.258	0.239	0.224	0.215
11	0.284	0.249	0.230	0.217	0.206
12	0.275	0.242	0.223	0.212	0.199
13	0.268	0.234	0.214	0.202	0.190
14	0.261	0.227	0.207	0.194	0.183
15	0.257	0.220	0.201	0.187	0.177
16	0.250	0.213	0.195	0.182	0.173
17	0.245	0.206	0.189	0.177	0.169
18	0.239	0.200	0.184	0.173	0.166
19	0.235	0.195	0.179	0.169	0.163
20	0.231	0.190	0.174	0.166	0.160
25	0.200	0.173	0.158	0.147	0.142
30	0.187	0.161	0.144	0.136	0.131
$n > 30$	$\frac{1.628}{\sqrt{n}}$	$\frac{1.358}{\sqrt{n}}$	$\frac{1.224}{\sqrt{n}}$	$\frac{1.138}{\sqrt{n}}$	$\frac{1.073}{\sqrt{n}}$

Multiplication of a vector $\boldsymbol{x}$ with a scalar $l \in \mathrm{R}^n$ provides the following vector:

$$l \cdot \boldsymbol{x} = \begin{pmatrix} lx_1 \\ lx_2 \\ \vdots \\ lx_n \end{pmatrix}$$

A *matrix* of elements of real numbers consisting of n rows and m columns, that is, a $n \times m$-matrix, is defined by

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & a_{2m} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \quad \text{example} : \boldsymbol{A} = \begin{pmatrix} 2 & 4 & 6 \\ 3 & 1 & 5 \\ 5 & 8 & 9 \end{pmatrix}$$

A *square matrix* has the same number of rows and columns, that is, its dimension is $n \times n$.

In a transposed matrix $\boldsymbol{A}^{\mathrm{T}}$, the rows and columns are interchanged, giving for the matrix $\boldsymbol{A}$

Table A.7 Coefficients for computing first derivatives (Savitzky and Golay, 1964, see chapter 3.2 in General Reading Section).

Points	25	23	21	19	17	15	13	11	9	7	5
−12	−12										
−11	−11	−11									
−10	−10	−10	−10								
−9	−9	−9	−9	−9							
−8	−8	−8	−8	−8	−8						
−7	−7	−7	−7	−7	−7	−7					
−6	−6	−6	−6	−6	−6	−6	−6				
−5	−5	−5	−5	−5	−5	−5	−5	−5			
−4	−4	−4	−4	−4	−4	−4	−4	−4	−4		
−3	−3	−3	−3	−3	−3	−3	−3	−3	−3	−3	
−2	−2	−2	−2	−2	−2	−2	−2	−2	−2	−2	−2
−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
−0	−0	0	0	0	0	0	0	0	0	0	0
+1	−1	1	1	1	1	1	1	1	1	1	1
+2	−2	2	2	2	2	2	2	2	2	2	2
+3	−3	3	3	3	3	3	3	3	3	3	
+4	−4	4	4	4	4	4	4	4	4		
+5	−5	5	5	5	5	5	5	5			
+6	−6	6	6	6	6	6	6				
+7	−7	7	7	7	7	7					
+8	−8	8	8	8	8						
+9	−9	9	9	9							
+10	−10	10	10								
+11	−11	11									
+12	−12										
NORM	1300	1012	770	570	408	280	182	110	60	28	10

$$\boldsymbol{A}^{\mathrm{T}} = \begin{pmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & & a_{n2} \\ \vdots & & & \vdots \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{pmatrix} \quad \text{example}: \boldsymbol{A}^{\mathrm{T}} = \begin{pmatrix} 2 & 3 & 5 \\ 4 & 1 & 8 \\ 6 & 5 & 9 \end{pmatrix}$$

If the transpose of a matrix is identical to the original matrix in every element, that is, $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}$, it is called a *symmetric matrix.*

A *diagonal matrix* is a special case of a symmetric matrix. In a diagonal matrix, only the main diagonal contains values other than zero and all off-diagonal elements are zero:

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & a_{\mathrm{nm}} \end{pmatrix} \quad \text{example}: \boldsymbol{A} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{pmatrix}$$

Table A.8 Coefficients for computing second derivatives (Savitzky and Golay, 1964).

Points	25	23	21	19	17	15	13	11	9	7	5
−12	92										
−11	69	77									
−10	48	56	190								
−9	29	37	133	51							
−8	12	20	82	34	40						
−7	−3	5	37	19	25	91					
−6	−16	−8	−2	6	12	52	22				
−5	−27	−19	−35	−5	1	19	11	15			
−4	−36	−28	−62	−14	−8	−8	2	6	28		
−3	−43	−35	−83	−21	−15	−29	−5	−1	7	5	
−2	−48	−40	−98	−26	−20	−48	−10	−6	−8	0	2
−1	−51	−43	−107	−29	−23	−53	−13	−9	−17	−3	−1
−0	−52	−44	−110	−30	−24	−56	−14	−10	−20	−4	−2
+1	−51	−43	−107	−29	−23	−53	−13	−9	−17	−3	−1
+2	−48	−40	−98	−26	−20	−48	−10	−6	−8	0	2
+3	−43	−35	−83	−21	−15	−29	−5	−1	7	5	
+4	−36	−28	−62	−14	−8	−8	2	6	28		
+5	−27	−19	−35	−5	1	19	11	15			
+6	−16	−8	−2	6	12	52	22				
+7	−3	5	37	19	25	91					
+8	12	20	82	34	40						
+9	29	37	133	51							
+10	48	56	190								
+11	69	77									
+12	92										
NORM	26 910	17 710	33 649	6 783	3 876	6 188	1 001	429	462	42	7

The diagonal matrix that has all 1s on the diagonal is termed the *identity matrix:*

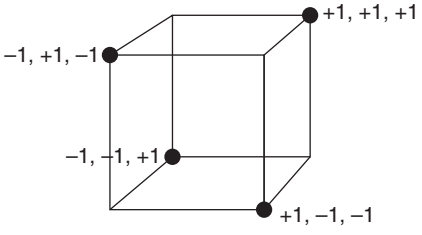
$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

The following examples describe *matrix addition* and *matrix subtraction:*

$$\boldsymbol{A} + \boldsymbol{B} = \begin{pmatrix} 2 & 4 \\ 1 & 3 \end{pmatrix} + \begin{pmatrix} -1 & 2 \\ 5 & -3 \end{pmatrix} = \begin{pmatrix} 1 & 6 \\ 6 & 0 \end{pmatrix}$$

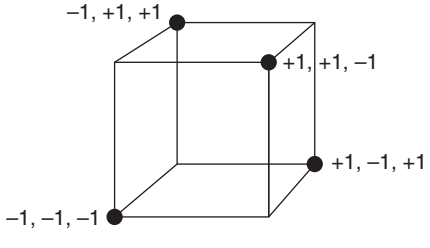
$$\boldsymbol{A} - \boldsymbol{B} = \begin{pmatrix} 2 & 4 \\ 1 & 3 \end{pmatrix} - \begin{pmatrix} -1 & 2 \\ 5 & -3 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ -4 & 6 \end{pmatrix}$$

Table A.9 Two-level designs (*half-cell designs*) for three, four, and five factors.



2^{3-1} design

Run	Factors		
	x_1	x_2	x_3
1	−1	−1	+1
2	+1	−1	−1
3	−1	+1	−1
4	+1	+1	+1



2^{3-1} design

Run	Factors		
	x_1	x_2	x_3
1	−1	−1	−1
2	+1	−1	+1
3	−1	+1	+1
4	+1	+1	−1

Table A.9 (*Continued*)

2^{4-1} design

Run	Factors			
	x_1	x_2	x_3	x_4
1	−1	−1	−1	−1
2	+1	−1	−1	+1
3	−1	+1	−1	+1
4	+1	+1	−1	−1
5	−1	−1	+1	+1
6	+1	−1	+1	−1
7	−1	+1	+1	−1
8	+1	+1	+1	+1

2^{4-1} design

Run	Factors			
	x_1	x_2	x_3	x_4
1	−1	−1	−1	+1
2	+1	−1	−1	−1
3	−1	+1	−1	−1
4	+1	+1	−1	+1
5	−1	−1	+1	−1
6	+1	−1	+1	+1
7	−1	+1	+1	+1
8	+1	+1	+1	−1

2^{5-1} design

Run	Factors				
	x_1	x_2	x_3	x_4	x_5
1	−1	−1	−1	−1	+1
2	+1	−1	−1	−1	−1
3	−1	+1	−1	−1	−1
4	+1	+1	−1	−1	+1
5	−1	−1	+1	−1	−1
6	+1	−1	+1	−1	+1
7	−1	+1	+1	−1	+1
8	+1	+1	+1	−1	−1
9	−1	−1	−1	+1	−1
10	+1	−1	−1	+1	+1
11	−1	+1	−1	+1	+1
12	+1	+1	−1	+1	−1
13	−1	−1	+1	+1	+1
14	+1	−1	+1	+1	−1
15	−1	+1	+1	+1	−1
16	+1	+1	+1	+1	+1

2^{5-1} design

Run	Factors				
	x_1	x_2	x_3	x_4	x_5
1	−1	−1	−1	−1	−1
2	+1	−1	−1	−1	+1
3	−1	+1	−1	−1	+1
4	+1	+1	−1	−1	−1
5	−1	−1	+1	−1	+1
6	+1	−1	+1	−1	−1
7	−1	+1	+1	−1	−1
8	+1	+1	+1	−1	+1
9	−1	−1	−1	+1	+1
10	+1	−1	−1	+1	−1
11	−1	+1	−1	+1	−1
12	+1	+1	−1	+1	+1
13	−1	−1	+1	+1	−1
14	+1	−1	+1	+1	+1
15	−1	+1	+1	+1	+1
16	+1	+1	+1	+1	−1

Multiplication of an $n \times n$ matrix $\boldsymbol{A}$ and an $n \times n$ matrix $\boldsymbol{B}$ gives the $n \times n$ matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \boldsymbol{AB} = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \\ a_{n1} & \cdots & a_{nk} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & & \\ b_{k1} & & b_{km} \end{pmatrix} = \begin{pmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & & \\ c_{n1} & & c_{nm} \end{pmatrix}$$

Table A.10 *Central composite design* for four factors with triplicate measurements in the center of the design.

Run	Factors			
	x_1	x_2	x_3	x_4
1	1	−1	1	−1
2	−1	−1	1	1
3	$-\alpha$	0	0	0
4	1	1	−1	1
5	0	0	0	0
6	−1	1	1	1
7	0	0	0	α
8	−1	1	1	−1
9	0	$-\alpha$	0	0
10	−1	−1	1	−1
11	0	0	0	0
12	−1	−1	−1	1
13	1	−1	1	1
14	0	0	α	0
15	−1	1	−1	−1
16	1	−1	−1	1
17	α	0	0	0
18	1	1	1	1
19	−1	1	−1	1
20	1	1	1	−1
21	1	1	−1	−1
22	0	0	0	0
23	0	α	0	0
24	−1	−1	−1	−1
25	1	−1	−1	−1
26	0	0	0	$-\alpha$
27	0	0	$-\alpha$	0

where $c_{ij} = \sum_{l=1}^{k} a_{il} b_{lj}$ for $1 \le i \le n$ and $1 \le j \le m$.

$$\text{Example}: \boldsymbol{C} = \begin{pmatrix} 2 & 3 & 1 \\ 3 & 4 & 1 \end{pmatrix} \begin{pmatrix} 7 & 4 \\ 1 & 3 \\ 5 & 0 \end{pmatrix} = \begin{pmatrix} 22 & 17 \\ 30 & 24 \end{pmatrix}$$

The *rank* of a matrix is the maximum number of linearly independent vectors (rows or columns) in an $n \times p$ matrix $\boldsymbol{X}$ denoted as $r(\boldsymbol{X})$. Linearly dependent rows or columns reduce the rank of a matrix.

Table A.11 *Box–Behnken design* for four factors with triplicate measurements in the center of the design.

Run	Factors			
	x_1	x_2	x_3	x_4
1	0	0	−1	1
2	1	1	0	0
3	0	−1	0	1
4	0	0	−1	−1
5	−1	1	0	0
6	0	−1	0	−1
7	1	0	1	0
8	−1	−1	0	0
9	1	0	0	1
10	0	−1	1	0
11	0	1	−1	0
12	0	0	1	−1
13	1	0	−1	0
14	1	0	0	−1
15	1	−1	0	0
16	0	0	1	1
17	−1	0	0	1
18	0	0	0	0
19	0	1	1	0
20	0	1	0	−1
21	−1	0	−1	0
22	0	1	0	1
23	−1	0	0	−1
24	0	0	0	0
25	0	0	0	0
26	0	−1	−1	0
27	−1	0	1	0

The *determinant* of a matrix is calculated by

$$D = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{vmatrix} = \sum_{i=1}^{n} (-1)^{i\text{th}} a_{ik} \det(\boldsymbol{M}_{ik})$$

$$\text{Example}: D = \begin{vmatrix} 2 & 4 & 6 \\ 3 & 1 & 5 \\ 7 & 8 & 9 \end{vmatrix} = 2(1\cdot 9 - 5\cdot 8) - 4(3\cdot 9 - 5\cdot 7) + 6(3\cdot 8 - 1\cdot 7) = 72$$

here $\boldsymbol{M}_{ik}$ is the $(n-1)\times(n-1)$ matrix where the ith row and kth column have been deleted.

Table A.12 Mixture designs (lattice designs) for three and four factors.

Run	Factor		
	x_1	x_2	x_3
1	1	0	0
2	0.67	0.33	0
3	0.67	0	0.33
4	0.33	0.67	0
5	0.33	0.33	0.33
6	0.33	0	0.67
7	0	1	0
8	0	0.67	0.33
9	0	0.33	0.67
10	0	0	1

Run	Factor			
	x_1	x_2	x_3	x_4
1	1	0	0	0
2	0.67	0.33	0	0
3	0.67	0	0.33	0
4	0.67	0	0	0.33
5	0.33	0.67	0	0
6	0.33	0.33	0.33	0
7	0.33	0.33	0	0.33
8	0.33	0	0.67	0
9	0.33	0	0.33	0.33
10	0.33	0	0	0.67
11	0	1	0	0
12	0	0.67	0.33	0
13	0	0.67	0	0.33
14	0	0.33	0.67	0
15	0	0.33	0.33	0.33
16	0	0.33	0	0.67
17	0	0	1	0
18	0	0	0.67	0.33
19	0	0	0.33	0.67
20	0	0	0	1

For *inversion* of matrix $\boldsymbol{A}$, we get for $\boldsymbol{A}^{-1}$

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{A}^{-1} = \begin{pmatrix} \frac{a_{22}}{D} & -\frac{a_{12}}{D} \\ -\frac{a_{21}}{D} & \frac{a_{11}}{D} \end{pmatrix}$$

where D is the determinant of the matrix.

$$\text{Example}: \boldsymbol{A} = \begin{pmatrix} 4 & 2 \\ 3 & 1 \end{pmatrix} \quad \boldsymbol{A}^{-1} = \begin{pmatrix} -0.5 & 1 \\ 1.5 & -2 \end{pmatrix}$$

In this example, we have inverted a 2×2 matrix. Perhaps an inversion by head could also be performed in the case of a 3×3 matrix. For larger matrices, however, a computer algorithm is necessary. In addition, matrix inversion is a very sensitive procedure, so that powerful algorithms, such as singular value decomposition (cf. Section 5.2), are to be applied.

A *linear transformation* from R^n to R^m (case a) or from R^m to R^n (case b) is possible by

a) multiplication of an n-dimensional column vector by an n-by-m-matrix forming a m-dimensional vector:

$$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A} = (x_1, x_2, \ldots, x_n)\begin{pmatrix} a_{11} & \cdots & a_{1m} \\ a_{21} & & a_{2m} \\ \vdots & & \\ a_{n1} & \cdots & a_{nm} \end{pmatrix} = \left(\sum_{i=1}^{n} x_i a_{i1}, \ldots, \sum_{i=1}^{n} x_i a_{im} \right)$$

b) multiplication of an $n \times m$-matrix by an m-dimensional row vector:

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ a_{21} & & a_{2m} \\ \vdots & & \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} x_i a_{1i} \\ \sum_{i=1}^{m} x_i a_{2i} \\ \vdots \\ \sum_{i=1}^{m} x_i a_{ni} \end{pmatrix}$$

Vectors that do not change their direction during a linear transformation are important. They are termed *eigenvectors* of the matrix $\boldsymbol{A}$. For every eigenvector of $\boldsymbol{A}$, there exists a real number λ, the eigenvalue, for which the following equation holds:

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

Eigenvector analysis is necessary, for example, in "Factorial Methods" Section for projection of multidimensional data.

Software

General Statistics

Statgraphics
Manugistics, Inc.
Statistica
StatSoft Inc.

Excel (Analysis functions)
Microsoft Inc.
SPSS
SPSS Inc.

Chemometrics

Unscrambler
CAMO – Computer Aided Modeling A/S
Pirouette
Infometrix, Inc.
Matlab
The MathWorks Inc.
R
The R Foundation for Statistical Computing

Neural Networks

Neuralworks
NeuralWare Inc.

Index

Chemometrics: Statistics and Computer Application in Analytical Chemistry, Third Edition. Matthias Otto.