Intelligent Systems Reference Library 67

Oleg Granichin
Zeev (Vladimir) Volkovich
Dvora Toledano-Kitai

Randomized Algorithms in Automatic Control and Data Mining

Springer

Intelligent Systems Reference Library

Volume 67

Series editors

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

Lakhmi C. Jain, University of Canberra, Canberra, Australia
e-mail: Lakhmi.Jain@unisa.edu.au

For further volumes:
http://www.springer.com/series/8578

About this Series

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included.

Oleg Granichin · Zeev (Vladimir) Volkovich
Dvora Toledano-Kitai

Randomized Algorithms in Automatic Control and Data Mining

Springer

Oleg Granichin
Department of Mathematics and Mechanics
Saint Petersburg State University
St. Petersburg
Russia

Dvora Toledano-Kitai
Department of Software Engineering
ORT Braude College
Karmiel
Israel

Zeev (Vladimir) Volkovich
Department of Software Engineering
ORT Braude College
Karmiel
Israel

ISSN 1868-4394 ISSN 1868-4408 (electronic)
ISBN 978-3-642-54785-0 ISBN 978-3-642-54786-7 (eBook)
DOI 10.1007/978-3-642-54786-7
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014934173

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

The authors start their book with basic question: *Why is randomization beneficial in the context of algorithms?* Or, say it another way: *When random choice is better than deterministic one?* This more general problem relates to a lot of situations beyond the mathematical framework. One can remember that random decisions were performed in ancient times, and the procedure of drawing lots was very common. Moreover political events such as election of governing officers in Athens were randomized. In our pragmatic age the field of random decisions in everyday life became narrower, but surprisingly mathematical algorithms opened the door for randomness. One of the first examples is the mixed strategy in two-person games, invented by von Neumann. However in the situations under consideration there is no "enemy" to mislead, the randomness is introduced to accelerate algorithms or make them more reliable. The first technique of this sort is the famous Monte-Carlo method.

The authors carefully examine numerous applications of randomized algorithms in statistics, optimization, control, data mining, machine learning. They demonstrate serious advantages of the algorithms in various cases and explain the reasons. Many of the proposed methods are new and are of wide interest. If compared with some other books on randomized algorithms, this monograph is not so specialized and is devoted to very broad field of problems. One of the peculiarities of the monograph - it summarizes deep researches in Russian-language literature which are not widely known to the Western audience.

I believe that the readers will find the book "Randomized Algorithms in Automatic Control and Data Mining" both fascinating and useful.

Boris Polyak,
Institute for Control Science, Moscow
IFAC Fellow
February 2014

Contents

Introduction

In 1948 Norbert Wiener's famous book
proclaimed the establishment of a new science of CYBERNETICS
in which the information–control relation
in the phenomena of the material world
plays the role of its fundamental property.
Vladimir Fomin, 1984 [118]

This text contains an exposition of randomized algorithms and their applications, thus providing some answers to the question: *Why is randomization beneficial in the context of algorithms?*

From Scalability toward Adaptability: Perspectives of Computing

In the new field of computers, the Olympic motto "Citius, Altius, Fortius" has been transformed into a new slogan: "Fast, Powerful, Miniature". In fact, these ostensibly conflicting goals join together into one: the technology approaching the creation of "mobile" artificial intelligence. Soon intelligent embedded computing devices will be able to perform such functions that computer users never even "dreamed" of a few years ago.

Throughout the history of civilization, there have been several information revolutions — transformation of social relations based on the dramatic changes in information processing and information technology. In each case, new qualities of the human community were achieved every time as consequences of these changes.

The first information revolution began with the separation of human from nature, as evidenced in rock painting and the appearance of the language that was able to handle abstract concepts. The invention of writing knowledge to be extended and preserved for transmission to future generations.

The next stage was the invention of printing, which radically changed the society and culture. Books became the mass universal distributor and the custodian of large volumes of information.

The era of electricity brought with it the capabilities of telegraph, telephone, radio and television that allow information to be sent quickly to any corner of the Earth.

In the wake of the Second World War and post-war economic development, research studies in nuclear and micro-molecular physics, solid state electronics and border effects led to the creation of the first industrial computing devices, supporting the growth of a new industrial revolution. For a quarter century these developments paved the way for a rapid burst of information technologies.

When the universal, multi-functional, automatic electronic devices known as computers were invented, they took over the majority of data processing, filing and recording functions.

Let's look at the history of the development of computer technology. For six decades it has paved the way from electronic lamps through transistors and integrated circuits to very large-scale integrated circuits. What will happen next? The main question is how to handle the huge amount of data being generated?

At the beginning of the twenty-first century we see that the technology-driven complexity of economy and society is increasing. During the last quarter of the twentieth century the human community was still an urban society with an industrial economy and mass products technology, but now it is increasingly characterized as a global society with a knowledge economy, and digital technology. Earlier, *a key resource* was capital, but now it is knowledge/information that dominates. Where earlier *distribution* focused early on a motorways, now the digital networks have come to the forefront. Previously the *scope* was a regional but now it is mainly global. Economy of scale, the undisputable key *success factor* the industrial economy, is less and less important as the complexity (and dynamics) of the knowledge economy increases. The new key success factor is ADAPTABILITY, which is the ability to rapidly produce a positive response to unpredictable changes [280].

People have always been interested in artificial intelligence. By the late twentieth century, the modeling of artificial intelligence was marked by two main approaches:

- machine intelligence, involving the setting of strict results of operations;
- artificial mind, based on the modeling of the internal structure of automated systems to correspond to the structure of a human brain.

But as of today, humanity has made a major move from the theories to a new reality: the cybernetic future. Many signs indicate that we have now entered a new phase — *a cybernetic revolution.*

New requirements and the challenges of globalization along with the exponential increase in the complexity of computing systems have already pushed us to think in practical terms about the prospects and possible changes in the computing paradigm:

"What is the process of computing?"

Today's objective trends are toward miniaturization and toward improvement of processor performance. These trends have brought technology to the threshold of traditional computing devices, as was predicted by Moore's Law [239]. Manufacturers are moving from priorities of increasing clock frequency and the power of one CPU toward multi-core processors, parallelism and so on.

Indeed, the standard for laptops today is to use multi-core processors, and, of course, supercomputer processors have many more cores. Now that "the genie is out of the bottle", there must be consequences. Soon systems will have dozens, and then thousands, of cores. Completely new architectures will emerge. Cores will be combined into complex blocks, different computing clusters will have parallel and simultaneous access to data, and computing units will communicate through a common memory. In fact, many aspects of the computing paradigm will change, including the nature of computing devices and of computational processes.

The traditional understanding of what is inside a computer and what constitutes a computing system will also change. These changes will lead to transformations in programming style and in the way in which computational devices are used.

The transition to a new paradigm of computing will probably cause the architecture of computing devices to shift toward a set of concurrent asynchronous models of interacting dynamical systems (functional elements). The properties of stochastic, hybrid, asynchronous and cluster behavior (among them the absence of rigid centralization and the dynamic clustering into classes of related models) will be more apparent and dominant among the new features of the future paradigm.

Stochasticity. It is well known that computers are becoming smaller and smaller. The size of an elementary computational element (a gate) now approaches the size of a molecule or an atom. At this scale, the laws of classical physics are not applicable and quantum laws begin to act, which, due to Heisenberg's uncertainty principle, conceptually do not give precise answers about a system's current state. On the other hand, stochasticity is a well known property of complex dynamical systems comprising a large number of components.

The *hybrid* nature of future computing necessitates the examination of a combination of continuous and discrete processes, i.e., registering the continuous evolution of physical processes during the work of this or that model

and abruptly switching from one model to another. The increase in the speed of computing devices and the reduction in their size inevitably lead to the need for operations with "transitional" processes. A serious limitation of the classical computation model is the separation of memory into isolated bits. From a certain level, the reduction in the length of a clock cycle time (strobe impulse) and in the distance between the bits makes it impossible to consider bits to be isolated, due to the operation of the laws of quantum mechanics. In the future it will be natural to switch from primitive operations with classical bits to operations definable by certain micro-dynamic models that operate with sets of related "bits". In this case, classical operations with bits may continue to be as simplest "models".

Success in solving traditional complex multidimensional problems (such as new algorithms working "per clock cycle") is the rationale for examining a wider class of models. Often it is possible to get an answer as a result of a physical adiabatic process. For example, the classical operation on bits is the transition of a physical system (trigger) from state "1" to "0". P. Shor suggested the quantum Fourier transform algorithm which can be performed for a time proportional to $(\log_2 N)^2$ and not for $N \log_2 N$, like the classical fast Fourier transform [293]. For the 10th Hilbert problem, [316] discusses solving the hypothetically possible "physical" method. The considered approach is based on the quantum adiabatic theorem and the algorithm works finite time. In [325] the powerful quantum algorithm was proposed for "per clock cycle" computation of efficient estimation of the gradient vector of the multidimensional function defined with a high degree of uncertainty. The operations typical of mathematical algorithms such as functions convolution can fully be found "in nature". Recent studies of similar models show that, due to the inherent nature of the capacity for self-organization, their performance is not necessarily separated into simpler computing blocks, i.e. they cannot always be written in the form of classical algorithms. One of the possible examples of an "analog" implementation of a function convolution using a large regular array of quantum dots with typical sizes of up to 2 nm can be found in [147].

Asynchrony. The refusal to use the standardized simple computing primitives inevitably leads to the refusal to synchronize the work of various components having significantly different physical characteristics and their own internal durations of "clocks". Within the framework of the classical set theory a controversial interpretation of the unified "clock cycle" concept is expressed in the insolubility of the problem of the continuum in the terms of Zermelo-Fraenkel axioms.

Clustering. Among the unexpected results of numerous attempts at developing complex stochastic systems (the creation of an adequate description of behavior and control) is the promising multi-agent systems model, in which the agents' connections topology changes in time. In this case, the notion of an agent may match both some dynamical model (a system component) and a specific set of models. In the absence of rigid centralization, such

systems are sufficiently capable of dealing effectively with complex problems by separating them into parts and by autonomously reallocating resources at the "lower" level. Efficiency is often enhanced by self-organization of the agents and dynamical clustering into classes of related models.

Information, Signals, Data, Knowledge and Control

All people know these words and have their own internal understanding of them. But for theory we need definitions. First we consider their features and the differences between them.

What does *information* means?

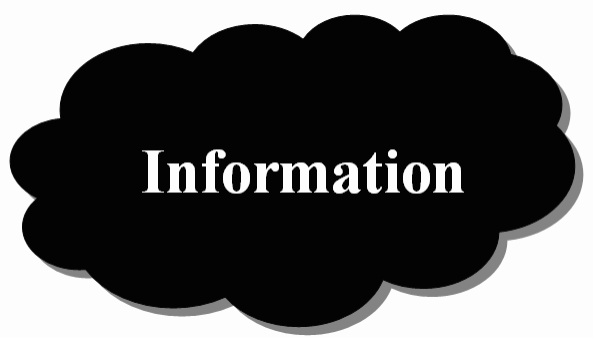

Fig. 1. What does *information* means?

Usually we consider *information* to be a message that something has happened. For example one important feature of information is that something has changed (an information object). But this information arises in the mind of someone or in something (an information subject).

The value of information depends on its ability to force the subject to certain actions, that is, on the subject's ability to formulate a control action based upon it. For example, information that a person is ill stimulates a control action — to take the medicine.

In fact, the two outer terms in the title of this section — information and control — are closely related to each other and cannot exist without each other (just like the two-faced Janus, or yin and yang in Chinese philosophy).

Information that does not push for action is meaningless, while many actions are meaningless without the underlying information.

More than sixty years ago Norbert Wiener proclaimed the era of the new science of *cybernetics*. He was one of the first to clearly notice that links between information and control are an essential part of any event in nature and wildlife [351].

Formalization of the decision-making process leads to the need to define the following concepts: signals, data and knowledge.

Information shows itself through changes in certain physical or social phenomena. For example, when a person is ill, some outward signs of illness appear, his body temperature may rise or the chemical composition of his blood may change.

Changes in physical and social phenomena and processes can be registered by the senses or by instruments we refer to as *signals*. The results of this registration are called *data*. Awareness of the links between the registered data and information, as well as the methods for selecting the control actions based on this or that information is called *knowledge*.

For example, registration of a physical signal — the temperature of the human body — may be fixed at 38.5^oC. This is the received data indicating that a particular individual is ill. The conclusion is based on our a priori knowledge.

In workable systems, existing knowledge allows forming control actions based on received data in such a way that these actions provide a compensation of negative information. Without knowledge, consideration of the control problem is practically meaningless.

New knowledge can be obtained over time by means of data processing. A collection of accumulated knowledge is referred to as *ontology*. In the context of control problems, ontology serves as a kind of database (knowledge) used for selection of the most adequate control.

The process of extracting knowledge from data determines a range of problems considered in the rapidly growing field of *Data Mining*.

Data Mining + Clustering

Generally speaking, data mining refers to automatic or semi-automatic processes intended to study large amounts of data to discover hidden appealing patterns. From a commercial perspective data mining applications are used in areas where a lot of data has been collected and warehoused, such as web data, e-commerce, credit card management and bank transactions. Patterns can be presented as groups of similar data items intended to be uncovered by cluster analysis methods or as groups of abnormal records considered within anomaly detection methodology and inner data dependencies handled by association rule mining.

Clustering is actually the most important unsupervised learning technique. It is used to structure a collection of unlabeled data such that the data may be divided into groups (clusters) whose members are similar in some way (see, e.g., [160]). Hence a cluster is constructed as a collection of points that are similar to each other and dissimilar to those located in other clusters. Cluster procedures group items based only on the inner data information regarding

the objects and their relationships. In many applications, the notion of a cluster is not formally defined.

Various clustering approaches describe clusters in terms of their prototypes, for example, centroids in the classical k-means that represent the clusters. These prototypes are used in many applications, data analysis or data processing techniques. Indeed, these cluster images are the result of averaging the cluster members and can be used for data compression by replacing each element by its clusters label.

This type of compression is called vector quantization and is often applied to image, sound, and video data, for example in cases of many highly similar objects. In such cases the volume of data may be considerably decreased and this process is constrained by suitable loss of information. In fuzzy clustering approaches (also referred to as soft clustering), points can be situated in more than one cluster such that each item is a set of membership levels (fuzzy memberships in clusters) representing the strength of the association between a given point and a particular cluster. Each point has a probable association with each cluster. Hard clustering is a particular case, where each item deterministically belongs to a single cluster.

Nearly all partitioning clustering methods can be divided into one of two categories: discriminative (or similarity-based) approaches and generative (or model-based) approaches. Approaches from the first category usually optimize objective functions involving pair-wise similarities in order to maximize the average similarities within clusters and minimize the average similarities between clusters. Methods in the second category are essentially distinct in that they aim to optimize the fit (global likelihood optimization, like the famous EM-algorithm) between the data and a certain mathematical model. For the most part there are no efficient and precise solutions, and some formulations are NP-hard. Considering the complexity of exact solving, for the most part approximation is used, either through polynomial-time approximation algorithms that give guarantees on the outcome attribute or their results, which give no such guarantees. The quality of a heuristic is frequently expressed by some bounds constructed to provide a tradeoff between approximation factors and running times. Various approximate clustering algorithms are able to supply solutions that are arbitrarily close to optimal (see, e.g., [2, 345]. Essential part of approaches to reduce the complexity provided by using of randomized algorithms which solve simplified problems with randomly sampled sets of data proceed. On the other hand, the clustering methodology actually supplies preprocessing intended to reduce the design complexity in control problems.

Control and Data Processing "On the Fly"

Changing a set of knowledge over a time is an important feature that allows for adaptation to changing conditions.

A reaction to information (a control action) may be previously defined "outside" in accordance with certain rules (laws) stored in the ontology, or it may be formed "inside" and adapt to changes (that is, trying to find the best decision for behavior in the new situation).

The development of control tools and computer science now allows for solving many practical problems "on the fly" when "smart" blocks are embedded into the control loops of simple and complex systems, and into a variety of decision support systems.

Usually, extracting information from data in a real time environment is marked by:

- substantial resource restrictions;
- an insufficient amount of data with the necessary diversity.

Thus, new difficulties arise in the process of control and data processing. Can these difficulties be overcome?

The processes of controlling and acquiring knowledge are often mutually contradictory.

The goal of control is to achieve some kind of generally stable state (if possible one that does not change over time). In this state, invariability provides very little information, making it impossible to identify or establish new links, values, and so on. For example, inspecting the surface of a stone lying in the dust on the side of a road cannot provide any knowledge. The stone must be turned over, lifted up, kicked or split to get any information.

The synthesis of control laws must often confront the problem of insufficient *variability* in the sequence of observations. For example, if the goal of adaptive control is to minimize the deviation of the system's state vector from a given path, a degenerate sequence of observations may emerge at a time when diversity of observations must be provided for the unknown system parameters to be identified. The known Feldbaum concept of *dual control* was formulated in [109]: *control must not only direct but must also learn.*

An adequate understanding of the question of control law synthesis is very important not only for experts in the field of control, but also for professionals in the area of data mining. As in the example of the stone lying near the road, a study in which data has only passively been recorded will not reveal the underlying laws and links in many important cases. A more complete investigation requires choosing such data mining strategies that will have a major impact on the processes under study. One of the typical problems of data mining is the clustering of data and their classification into classes on various grounds. In the context of control, problems of different classes usually compare different control strategies. The multivariance of phenomena is not always manifested when they are studied passively. In the case of natural and social phenomena, the system is normally in a stationary (almost unchanged) state for an extended period of time. Therefore, to study the system in detail, some disturbances must somehow be added.

Do We Need to Divide Data Processing and Control?

The current paradigm of using of computing devices is based on the historical division between data processing and decision-making (after treatment). The grounds of this division can be traced in the history of computer equipment. Initially, there were very few computers that occupied a great deal of space and required special conditions for use. Special computer centers were therefore established to provide integrated solutions and perform a variety of different tasks in one place. Today the creation of supercomputers still remains a major priority. Embedded devices are traditionally assigned the role of data collection or are used to implement certain control actions. In some cases, they were used as controls in simple feedback loops. Supercomputers are now taking on data mining tasks.

But we must be clearly aware of the validity of this traditional paradigm. In nature and in society, information-control relations are the basis of all phenomena and processes. If we artificially separate data processing and control, we have essentially reduced our potential opportunities to use information and communication technologies.

Since the beginning of the twenty-first century there has been a noticeable surge of interest in control theory and control subjects in networks, collective interaction, multi-agent technology, and so on [142], largely due to technological advances. Miniaturization and computer technology performance have now reached such a level that some tiny embedded real-time systems have computing power comparable to that of twentieth century high-performance computers. Simple embedded systems are increasingly being replaced with intelligent embedded devices.

New alternatives allow for a fresh look at the area of data mining, which is becoming more and more conventional. The literature has begun focusing on the notion of reviving the science of Cybernetics with a capital C. This science began in the nineteenth century with the controllers of mechanical systems, and by the end of the twentieth century it reached the stage of profound integration with digital data processing and decision technologies. In the twenty-first century control theory has begun to focus more and more on plant networks. One of the natural outcomes of this process is the revival of the notion of cybernetics in a new light, as an amalgam of the three main components of progress made during the second half of the twentieth century: Control Theory, Communication Theory, and Computer Science $= C^3$ [17].

Is it possible that applying the cybernetic paradigm in data processing and data mining, with processes of knowledge extraction and information obtaining taking into account (and relying on) the inextricable link between information and control, can yield a new entity?

"Yes, it is possible!"

Vladimir Yakubovich, founder of the Theoretical Cybernetics Department at St. Petersburg State University, loved the following joke about the connection between data processing and controls.

One day, two mice on a dairy farm both fell off shelves of cheese at the same time, and each one landed in a different can of milk. The cans were partially full. The mice had to face the problem of how to survive. Their time was very limited, they could not breathe under the surface and they did not have sufficient forces to flounder around much. They could adopt one of two possible strategies.

1. Analyze the situation, look at the flat wall stretching above them and try to find a way to survive.
2. Begin to flounder around actively trying to do something, physically fighting to remain alive.

The first mouse could not think of a way to survive and drowned.

The second mouse floundered around so much that the milk in the can was shaken up and turned into thick cream. The mouse lived on the surface for a couple of days and finally found a way to get out of the can.

Purposes and Structure

The main purpose of this book is to provide a variety of examples to illustrate that using the cybernetic paradigm in data processing and data mining can improve the efficiency of data processing and control in cases when knowledge extraction and information obtaining processes take into account and rely upon the inextricable link between information and control. Under conditions of uncertainty this improvement is achieved by randomization of inputs (control actions) and the use of closed control strategies.

The traditional approach in designing a variety of systems and control algorithms involves the execution of deterministic algorithms comprising a deterministic sequence of steps. This approach can be generalized to include *randomization*. In randomized algorithms, one or more steps are based on a random rule under which one of many deterministic rules is chosen according to a random pattern. Randomization allows the introduction of a new term: "a probabilistic successful algorithm". In many cases, when a deterministic success cannot be achieved, the probabilistic successful algorithm is proposed as a meaningful alternative. Randomization is a powerful tool for solving a number of problems in the organization of complex systems that are considered intractable using deterministic methods.

This book describes the general notion of randomized algorithms as used in control and data processing.

The book is organized as follows. Chapter 1 presents a general vision of randomized algorithms in automatic control and data mining. Chapter 2 provides a historical overview. The main ideas and approaches of randomized stochastic approximation are described in Chapter 3, and linear models are considered in Chapter 4. Chapter 5 discusses possibilities for randomization in the feedback channel of a linear control plant. Chapters 6 and 7 are devoted to several aspects of the conventional clustering theory. The most widespread

clustering approaches, k-means and EM, are presented together with fairly new spectral clustering techniques in Chapter 6, while the more recently developed cluster validation methods presented in Chapter 7 form the general amalgamated standpoint of probability distances.

Acknowledgments

This work has been partially supported by a grant from the ORT Braude College research committee, the European Union seventh framework program via the SYSPATHO project (grant number 260429) and RFBR (grant number 13-07-00250). Some parts of the work were done with partial support of the Institute of Problems in Mechanical Engineering and the Research Laboratory for Analysis and Modeling of Social Processes of Saint Petersburg State University (grant numbers 2.50.2219.2013 and 6.37.181.2014).

Basic Notations

We use the following notations in the text of the book:

x — information, usually unknown, or an estimated (optimal) point (often a vector variable);
u — a control (controllable inputs);
y — an observed variable;
v — a noise in observations or measurements;
w — a non-controllable perturbation (usually a random vector);
$\widehat{x}$ — an estimate of x (scalar, vector or sometimes matrix);
$\widehat{X}$ — an estimate of a set which can contain x;
$\mathcal{U}(\cdot)$ — a control strategy;
Δ — a trial simultaneous perturbation (test probing signal);
t — a time instant (discrete or sometimes continuous);
z^{-1} — time shift operator: $z^{-1}y_t = y_{t-1}$;
$\mathbf{a}, \mathbf{b}, \ldots$ — vectors;
a_i — an i-th element of a vector $\mathbf{a}$;
$\mathbf{A}, \mathbf{B}, \ldots$ — matrices (or operators); $\mathbf{I}_{\mathrm{d}}$ — the identity matrix of size $d \times d$;
$a_{i,j}$ — an element on i-th row and j-th column of a matrix $\mathbf{A}$;
$\cdot^{\mathrm{T}}$ — vector or matrix transpose operation;
$\mathbf{A} > 0$ ($\mathbf{A} \geq 0$) — matrix $\mathbf{A}$ is symmetric and positive definite (or non-negative);
$\lambda_{\min}(\mathbf{A})$ and $\lambda_{\max}(\mathbf{A})$ — the minimum and maximum absolute value of matrix $\mathbf{A}$ eigenvalues;
$\mathrm{Tr}[\mathbf{A}]$ — trace of $\mathbf{A}$ (sum of the diagonal elements);
$\mathrm{diag}(\lambda_1, ..., \lambda_d)$ — a diagonal matrix;
$\langle \cdot, \cdot \rangle$ — vectors scalar product. $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_i^d a_i b_i$ if $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$;
$\| \cdot \|$ — the Euclidean norm of vector or matrix;
$\| \cdot \|_\rho$ — the ℓ_ρ-norm of vector $\mathbf{a}$: $\|\mathbf{a}\| = (\sum_i |a_i|^\rho)^{\frac{1}{\rho}}$,
$\mathbb{N}$ — the set of the natural numbers;
$\mathbb{Z}$ — the set of the integers;
$i..j$ — the subset $\{i, \ldots, j\}$ of the set $\mathbb{Z}$;

$\mathbb{Q}$ — the set of the rational numbers;
$\mathbb{R}$ — the set of the real numbers;
$\mathbb{C}$ — the set of the complex numbers;
$\mathbb{R}_+$ — the subset $[0, +\infty)$ of the set $\mathbb{R}$;
d — the size (dimension) of $\mathbf{x}$ if it has a vector structure;
$\mathbb{R}^d$ — the Euclidean space of the dimension d;
$\mathbb{W}$ — a subset of the Euclidean space;
$|\mathbb{W}|$ — the cardinality number of the set $\mathbb{W}$;
$f'(\cdot), \nabla f(\cdot)$ — the vector-gradient of $f(\cdot)$;
$\mathrm{Prob}\{\cdot\}$ — a probability of the event;
$\mathrm{P}(\cdot)$ — the probability distribution function;
$\mathcal{F}$ — σ-algebra of probability events;
$E\cdot$, $E\{\cdot\}$ — an expectation;
$E\{\cdot|\cdot\}$, $E_{x}\cdot$, $E_{\mathcal{F}}\cdot$ — a conditional expectation;
$\mathbf{1}_A(w)$ — the characteristic function of a set A (the Heaviside function)

$$\mathbf{1}_A(w) = \begin{cases} 1, & \text{if } w \in A, \\ 0, & \text{otherwise;} \end{cases}$$

$\mathcal{N}(x, \sigma)$ — the density of the one variable normal distribution having the mean x and the standard deviation σ;
$G(\cdot|\mathbf{x}, \mathbf{\Gamma})$ — the multivariate Gaussian density with the mean $\mathbf{x}$ and covariance matrix $\mathbf{\Gamma}$;
$sample(\mathbb{W}, m)$ — a procedure of drawing a sample of size m from the population $\mathbb{W}$ without replacing;
Cl_k — a clustering algorithm (a clusterer) for partition into k clusters;
k — the examined number of clusters;
$\mathcal{X}^k(\mathbb{W})$ — a partition of the set $\mathbb{W}$ into k clusters;
$\gamma^k_{\mathcal{X}} : \mathbb{W} \to 1..k$ — a labeling function presents the partition $\mathcal{X}^k(\mathbb{W})$;
Ψ_k — the set of all possible permutations of $1..k$;
$\mathrm{C}^s_T = \begin{pmatrix} T \\ s \end{pmatrix}$ — the number of combinations of T elements by s;
$\forall$ — quantifier "for all";
$\exists$ — quantifier "exist".

Part I

Randomized Algorithms

A *randomized algorithm* is an algorithm that, in addition to its input data, receives a stream of random bits that it can use to make random choices [182].

Randomized methods have been actively developed for the solving of many kinds of problems. A randomized algorithm is a procedure where one or more steps are based on random rules. That is, at some stage we "flip a coin" (call on fate to choose for us) instead of making a decision ourselves. Such an approach is contrary to the typical approach of scientists. It is usually assumed that scientists have a lot of knowledge and seek to ensure the right choice based on their knowledge.

Why should randomized methods be any good?

The conscious choice to use randomized methods demands that we question ourselves about this issue.

On the one hand, when a problem requires a large amount of a "brute force" in selecting among options, algorithms based on the random selection of alternatives allow the achievement of a good result with a certain probability for a restricted time and significantly reduce the volume of operations. Moreover, their accuracy usually weakly depends on the data dimension.

On the other hand, in many cases the randomization of observations can offset the negative influence of systematic errors (the bias effect of arbitrary noise). Hence, in the following sections we seek to show that randomization facilitates enrichment of the observation data and the design of speedy algorithms capable of annihilating systematic errors.

1

Feedback, Averaging and Randomization in Control and Data Mining

1.1 Feedback

1.1.1 Information and Control

We use the notation x to denote the *information* and u to denote the *control*. The process of making control decisions can be written as

$$u = \mathcal{U}(x), \tag{1.1}$$

where $\mathcal{U}(\cdot)$ is a function of x. Often the information x is identified with the state vector of the system. In this case, we say that the relation (1.1) defines the state feedback.

Realization of a control affects the information contributing to new changes in the object of information. Formed control u enters the system and affects the state x changing it in many cases.

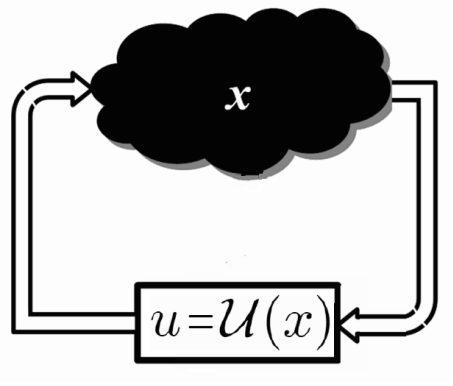

Fig. 1.1. Information and control

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_1,

The relation between information and control can be represented by the circuit of Figure 1.1.

We use the following notations f for *signals*, and y for *data*. A typical scheme of the signal registration process is shown in Figure 1.2.

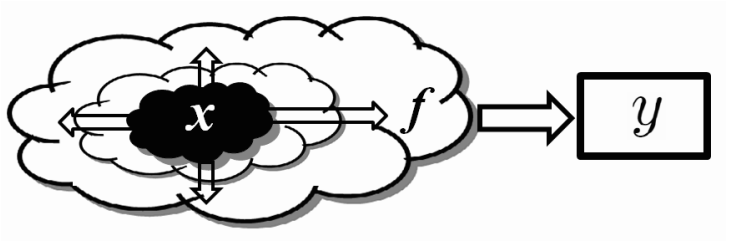

Fig. 1.2. Data acquisition

If all we have are available observations, the decision-making process involves generating the control action through the registered data

$$u = \mathcal{U}(y). \tag{1.2}$$

Control laws of this type are referred to as feedback on observations.

Until now we have left open the question of selecting a feedback function. In classical control theory, the concepts of observability and controllability correspond to opportunities to restore the state based on observations or to move the plant into any given state [117]. A detail description of these issues is beyond the scope of this book.

In workable systems, the existing *knowledge* allows the control actions u to be formed on the basis of the received data y in a way that yields winnings or somehow compensates for the negative information. Without knowledge consideration of the control problem is almost meaningless, though even if a priori knowledge is absent, knowledge can often be obtained over time.

Typically, the feedback function $\mathcal{U}(\cdot)$ most adequate to the current situation is chosen according to the available (accumulated) knowledge (ontology).

Information processing and control decision-making may be schematized as follows (see Figure 1.4). We assume that the information x regarding a variable phenomenon (or control plant, or some element of the environment) important for the researcher and mediated by some forms of communication Φ manifests itself via the signal f. The signal f may be observed using some recording instrumentation or special sensors. The process of observation (data acquisition) is usually sufficiently complicated and may be treated as the application of some observation operator H to the signal f. The data y are obtained as the result of interaction with the recording instrumentation and used by the researcher in an attempt to restore (estimate) the information x by generating the estimate $\hat{x}$ through some *knowledge* K. The

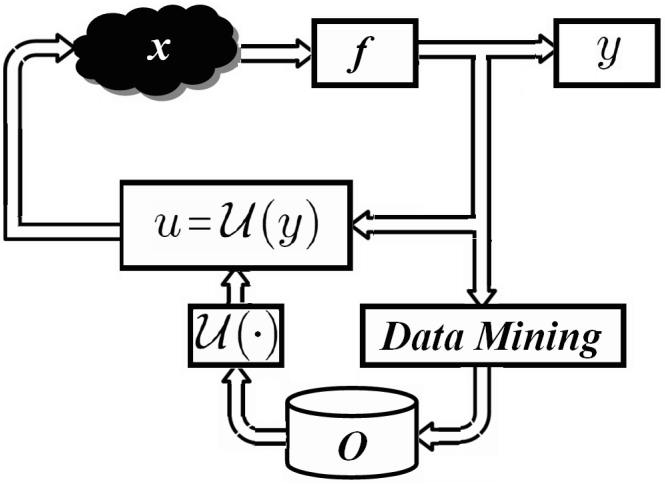

Fig. 1.3. Selecting of control functions $\mathcal{U}(\cdot)$

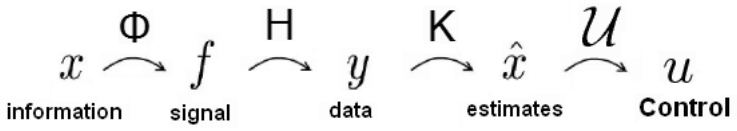

Fig. 1.4. Diagram of a data acquisition and processing

estimation results to a certain extent characterize the changes occurring in the phenomenon of interest to the researcher and are then used to make one or another control decision.

1.1.2 Signals and Data Sizes

As noted above, a user can obtain some information x through a signal f with which it is associated.

For the formal mathematical description we need to understand the possible sets that may contain the signals and data. It is usually assumed that the signals take values in a real or complex space and that the data are integers or real numbers. For the sake of simplicity, we will usually consider the signals and the data to be real numbers.

What are the possible sizes (dimensions) of the signals and data spaces?

The key to answering this question is to understand that it is treated as changing. That is, already in the definition we introduce the time axis, which is divided into moments before and after the current.

Hence, signals and data can be parameterized by time. That is, we enter the time t as one of the dimensions. The signal f at time t is denoted by f_t, and respectively, y_t is the data recorded at time t.

Time can be considered as continuous or discrete. We further assume a discrete time in which the difference is clearer between the time before and the time after the current moment.

For a fixed t the volume of registered data is usually considered to be finite because it corresponds to the number of senses and recording devices used. The situation is complicated by complex systems consisting of components (subsystems), each of which registers its own set of data. Thus the total size of all recorded data can be infinite for an unlimited set of subsystems. Recently, however, a network of plants is considered for such problems since the single act of registering the data in any of the subsystems is not treated as the availability of data in a shared storage. Time and resource costs for data copying and transmission are critical in comparison with the total time for solving the problem, and in many practical cases considering the network (multi-agent) formulation of the problem is more promising.

A situation including the possible dimension of signal f_t is even more complicated. Many practice-oriented theories also suggest a finite dimensional signal space. But in reality this is a very strong constraint. The same information can reveal itself through different types of signals. Information about one of the system components actually has other facets if this component is included in different subsystems with other components. Each of these subsystems can generate its own signals, and so on.

How much data should be processed to get the required information? With respect to potentially large dimensions of registered data and signals, the answer to this question is extremely relevant for practical tasks.

What simplifications are possible and permissible?

1.1.3 Observations with Noise

Before answering the last question we consider a more complicated description.

In practice, an error (noise) typically adds to a signal when data is received by any registration device (see Figure 1.5).

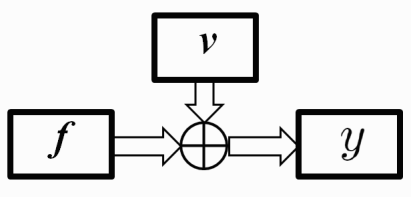

Fig. 1.5. Observations with noise

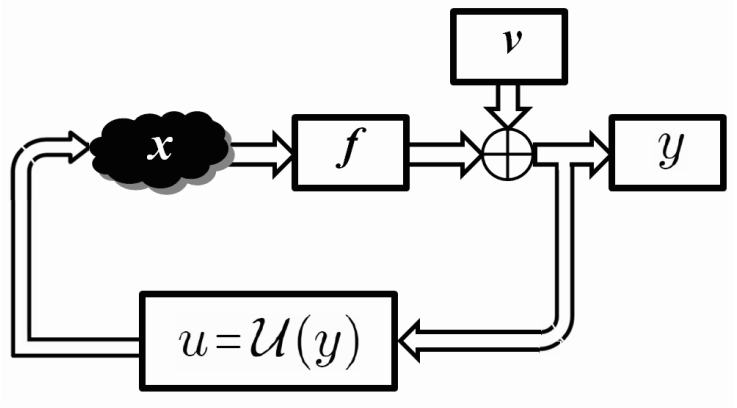

Fig. 1.6. Close loop on observations with noise

In the case of observations with noise, a block diagram of a system with feedback is shown in Figure 1.6.

If some signal f enters the recorder with noise v, an observation y can be written as

$$y = \mathbf{H}f + v, \tag{1.3}$$

where $\mathbf{H}$ is an operator acting from the signal space into the data space. For example, if T is a signal size and m is a data size, an operator $\mathbf{H}$ can be given by $m \times T$ matrix.

If we have only a single measurement with nonzero noise v then we have nothing to say about the required signal f or its mean value.

What are the possible options for addressing this case?

- We have the exact observation model without error: $v = 0$.
- In some cases, a well formulated experiment with careful measurements can reduce the error v to a minimum such that $v \approx 0$.
- If we initially did not have a "clean" experiment, we can try to make it clean over time, that is, $v_t \to 0$ as $t \to \infty$.

(These three assumptions are conventional and widely used.)

- In other cases, noises in observations of physical phenomena are often quite reasonably assumed statistically so that $\{v_t\}_{t \in 1..T}$ is a sequence of independent and identically distributed random variables with zero-mean and finite variance σ_v. Under such conditions we can apply the strong law of large numbers from probability theory (see [292]).
- Reasonable problems formulations are possible under arbitrary external noise. For these cases we can use randomized algorithms [148].

1.2 Averaging

The stage of data acquisition plays a key role. All processes and phenomena occur in time. In reality, signals often have an analog nature. The simplest example of data acquisition is represented by acquiring instantaneous values of a signal f. In applications, a very important problem is that the mean value of incoming signals is required.

In statistical mechanics and physics, methods based on data averaging have been used, beginning with the theories developed by Gibbs [125] and Lebesgue [210]. In the case of large and complex systems consisting of similar components, the Krylov-Bogolyubov approach has proved itself (see [200]).

The hypothesis of ergodicity [292] can be used to explain many physical and social phenomena in the absence of external influences. According to this hypothesis, the average value of the spatial characteristics of different system components calculated at a given time is the same as the average time value of one of the components.

Thus, the notion of averaging is in agreement with the construction of a conventional registering device, which accumulates characteristics of a physical phenomena for a certain period of time and calculates their mean value as a result of measurement.

Not only can averaging simplifies the description of events, in many cases it can even compensate for the effect of errors in the data.

1.2.1 Data Averaging

If our registering apparatus averages "instantaneous" scalar signals f_t coming at $t = 1, \ldots, T$ we then receive

$$y = \frac{1}{T}\sum_{t=1}^{T} f_t + \frac{1}{T}\sum_{t=1}^{T} v_t \tag{1.4}$$

as an output. This form corresponds to the general Equation (1.3) with $\mathbf{H}_T = (\frac{1}{T}, \ldots, \frac{1}{T})$, $\mathbf{f} = (f_1, f_2, \ldots, f_T)^{\mathrm{T}}$ and $v = \frac{1}{T}\sum_{t=1}^{T} v_t$.

Assume that $\{v_t\}_{t\in 1..T}$ is a sequence of independent and identically distributed random variables with mean value M_v and finite variance σ_v. Based upon the law of large numbers [292], for any small $\varepsilon > 0$ we have

$$\mathrm{Prob}\left\{\left|\frac{1}{T}\sum_{t=1}^{T} v_t - M_v\right| > \varepsilon\right\} \to 0 \;\; as \;\; t \to \infty.$$

Hence, we can use the first term of (1.4) and M_v to achieve a sufficiently accurate estimation of the average value of the signal f_t

$$\hat{f} = \frac{1}{T}\sum_{t=1}^{T} y_t - M_v = \mathbf{H}_T \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix} - M_v = \mathbf{H}_T\mathbf{y} - M_v.$$

Let us consider the example described above in more detail. What information have we received as a result?

Before measuring, we could have some a priori information about the average value of the signal f. For example, we might assume it to be zero, that is, we assume that it is virtually absent.

After measuring, we obtain the estimate (approximation) $\hat{f}$ for the mean value of the signal f. If this estimate is not consistent with our a priori concepts, it should be changed.

New obtained information may be inserted into the repository of knowledge, so that it becomes part of the ontology.

1.2.2 Averaging in Stochastic Control

Averaging methods are widely used in the theory of adaptive control of stochastic systems. The first result for the averaging of discrete stochastic control systems was published in [232], where a discrete-averaged model was used. The proximity of the probabilities of trajectories of stochastic and averaged systems was proven for independent v_t based on the Krylov-Bogolyubov averaging method [200]. Ljung [214] made a significant contribution to the dependent v_t generated by the controlled Markov chain. Here, too the Krylov-Bogolyubov method was used. The method continued to be developed in [203], in which the averages for some functionals of state vectors were studied.

Other results [91, 92] are based on the weak convergence of trajectories of stochastic systems to the trajectories of an ordinary (or stochastic) differential equation. An elegant approach to study asymptotic behavior was developed by Kushner [204], who used the theory of weak convergence of random functions.

1.3 Efficiency of Closed-Loop Strategies under Uncertainty

As noted in the Introduction, the current paradigm of using computing devices is based on the historical separation between data processing and decision-making. In contrast with this paradigm, we consider two examples of efficient data processing and control when the processes of knowledge extraction and information obtaining take into account and rely on the inextricable link between information and control. The first example described in this section shows how solution of an optimal control problem under uncertainty

is dependent on the choice of the class of permissible control strategies. The second example, which is described in the next section, deals with randomized inputs (controls) in an observation model.

Let us consider a control plant with inputs u_t and outputs y_t and assume that the initial condition $y_0 = 1$ and that the plant's dynamics for $t = 1, 2$ is described by the equation

$$y_t + y_{t-1}x = u_{t-1} + v_t \tag{1.5}$$

with two types of uncertainties v_t and x:

- disturbances varying with time v_t are unknown and bounded for all t: $|v_t| \leq 1$;
- the model coefficient x is unknown and also bounded: $x \in [1, 5]$, but it does not vary with time.

We can choose the inputs u_0 and u_1.
The objective is to minimize $|y_2|$.
The minimax optimal control problem is

$$J = \sup_{x\in[1,5]} \sup_{|v_1|\leq 1, |v_2|\leq 1} |y_2| \to \min_{u_0, u_1}. \tag{1.6}$$

We compare the performance of the minimax optimization for two classes of permissible control strategies:

- open-loop control
- closed-loop control.

a) Open-loop control. This class consists of all possible pair (u_0, u_1).
To choose u_0 and u_1 we rewrite (1.6), using the view of y_2 in (1.5) as a function of x, v_1, v_2, u_0, u_1:

$$J = \sup_{x\in[1,5]} \sup_{|v_1|\leq 1, |v_2|\leq 1} |x^2 - u_0 x + v_1 x + u_1 + v_2|.$$

After carrying out the above maximization with respect to x, v_1 and v_2, we obtain a function of the variables u_0 and u_1. Minimizing this function with respect to u_0 and u_1 yields the values $u_0 = 7,\ u_1 = 12.25$ and $J_{ol}^{opt} = 8.25$.

b) Closed-loop control. At time $t = 1$ we get the output y_1 and know the previous control u_0. The control strategies from a class of closed-loop controls are determined by the pair (y_1, u_0) and the feedback function $\mathcal{U}_1(y_1, u_0)$.

By virtue of (1.5) and conditions $|v_1| \leq 1$, for a fixed output y_1 and control u_0 the following inequality holds:

$$|y_1 + x - u_0| \leq 1.$$

Hence, we can define the interval

$$\widehat{X}(1) = [\widehat{x}_-, \widehat{x}_+] = [1, 5] \cap [-1 + u_0 - y_1, 1 + u_0 - y_1], \tag{1.7}$$

which contains x and whose bounds are calculated via the formulas:

$$\widehat{x}_- = \max\{-1 + u_0 - y_1, 1\},\ \widehat{x}_+ = \min\{1 + u_0 - y_1, 5\}.$$

To choose u_1 according to the Bellman's method of dynamic programming [32] we rewrite (1.6) allowing for (1.5) and (1.7) as follows

$$J = \sup_{x\in[1,5]} \sup_{|v_1|\le 1} \left(\sup_{x\in\widehat{X}(1)} \sup_{|v_2|\le 1} | - y_1 x + u_1 + v_2| \right). \tag{1.8}$$

The expression between brackets is determined by $\widehat{X}(1), y(1)$ and $u(1)$. Minimization with respect to u_1 yields the formula

$$u_1 = \mathcal{U}_1(y_1, u_0) = \frac{\widehat{x}_+ + \widehat{x}_-}{2} y_1. \tag{1.9}$$

Substituting (1.9) into (1.8), we obtain

$$J = \sup_{x\in[1,5]} \sup_{|v_1|\le 1} \left(\frac{\widehat{x}_+ - \widehat{x}_-}{2} |y_1| + 1 \right).$$

Now, eliminating y_1 in view (1.5) and performing the above maximization operations, we can rewrite J as a function of u_0. Minimizing with respect to u_0 yields the values

$$u_0 = -3, \qquad J_{cl}^{opt} = 2.125.$$

Thus, we have:

$$J_{cl}^{opt} = 2.125 \ << \ J_{ol}^{opt} = 8.25.$$

The general idea of this example appears to result from the arguments in [298].

The dependence of performance on the class of non-anticipating strategies is not obvious, since if all parameters of the control plant are known and there are no disturbances v_t, the classes of open and closed-loop controls are coincident.

1.4 Estimation under Arbitrary Noise

1.4.1 Can Smart Estimates Be Obtained?

Let us return to the example of averaging in a registering device from Subsection 1.2.1.

What can we do when noises v_t are not random (statistical)? For example, the sequence $\{v_t\}$ is a set of values of an unknown (arbitrary) function.

The problem of estimating the mean value of signal f_t under arbitrary noises seems absurd from the point of view of a classical data processing

paradigm. But the reason for this absurdity is not because it makes no practical sense, for this is a very important problem. It seems absurd because we cannot reasonably solve it based on the classical paradigm.

For the sake of simplicity, consider the scalar case. Modernize the problem by including a controllable input u into the observation model. According to the paradigm of information and control inseparability, we assume that the measured signal f at time t is directly determined by the current input u_t and some unknown parameter x (an unknown gain/attenuation coefficient of input)

$$f_t = u_t x. \tag{1.10}$$

The model of observations (1.3) can be rewritten as:

$$y_t = u_t x + v_t. \tag{1.11}$$

And we can

- choose the inputs (controls) $u_t,\ t \in 1..T$,
- measure the outputs y_t (see Figure 1.7).

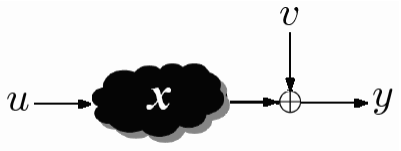

Fig. 1.7. The model of observations

When using $u_t \equiv 1$, we obtain the traditional problem of estimating an unknown parameter x observed with noise (see Subsection 1.2.1).

Even school pupils are faced with similar problems when they use physical experiments to measure the result of an impact on the system. For example, applying a different force to a spring, we get different lengths of tension or compression. But the obtained results are not arbitrary numbers. They are determined by the characteristics (parameters) of the spring itself (the elasticity coefficient etc.) and some external noise.

Let us examine the scheme shown in Figure 1.7. The considered system is a black box with input u_t and output y_t. The system is characterized by the parameter x, which is unknown (in the spring example, x is an elasticity coefficient). The experimenter can choose input u_t which is fed to the black box (in the spring example, we can stretch or compress it to a distance u_t). An external noise v_t is added to the output of the black box. This noise is not related to internal processes within the black box (in the spring example, the noise is a measurement error made by a dynamometer).

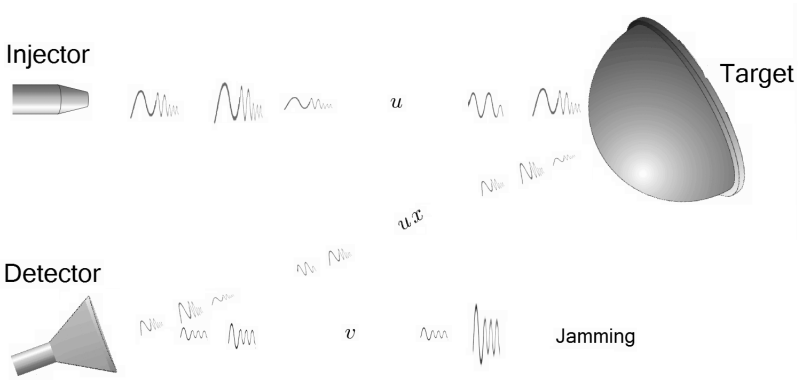

Fig. 1.8. The source, target and detector of the reflected signal

Another typical example is shown in Figure 1.8. This example may be from the field of materials research, either subsoil or a variety of remote sensing applications.

The source sends some signal into the space: either the flow of some particles (electrons, α-particle, and so on) or a wave of a different nature (acoustic or electromagnetic). The flow rate can be varied and accurately measured (for example, measuring the value of "departed charge" of a probing α-testing in medicine). When a signal meets the target, either it is reflected off the target (echo-acoustics, radar, and so on) or it interacts with the target, generating a new "reflected" signal (perhaps of a different nature). After that, the intensity of the reflected signal with noise (jamming) is measured by the detector. The problem is to determine the unknown parameter x, which is equal

- either to zero
- or to a positive value.

The first case corresponds to the absence of the target with desired reflection properties. In the second, we deal with certain physical characteristics: either the distance to the target, or the size, or the reflective properties of the material, or something else.

A common point of these examples is the experimenter's ability to actively influence the results of observations (to submit an input action).

What is the widest possible class of noise v_t for which it is still realistic to try to sensibly answer the problem of estimating an unknown parameter x?

When writing the equation (1.11) we can naturally assume that the second term on the right side v_t includes all the uncertainties affecting the output y_t that are not related to u_t because u_t is included only in the first term on the right side of (1.11). Such noise will be referred to as external noise,

emphasizing its independence from internal inputs feeding into the system. Is there a way to solve the problem for this class of uncertainties?

More precisely, the problem is *to find or estimate* the unknown parameter $x \in \mathbb{R}$ by the sequence of inputs and outputs $\{u_t, y_t\}$ without any restrictions on the sequence $\{v_t\}$ of external noises.

We already mentioned that from a deterministic point of view such a statement of the problem seems absurd. There is no deterministic algorithm that yields a sensible answer other than a meaningless solution for the entire real axis. For a fixed number of observations and any proposed numerical or interval answer, one can always choose v_t such that the following observation will be wrong for the proposed answer.

An algorithm for sequential estimation of an unknown parameter x from the right side of equation (1.11) consists of three steps (see Figure 1.9):

1. Control u_t selection and feed it to the system input.
2. Receive a response y_t from the system.
3. Estimate the parameter x based on the data obtained u_t, y_t (for example, calculation of an estimate $\widehat{x}(t)$ or set $\widehat{X}(t)$ containing x).
4. Repeat Steps 1–3.

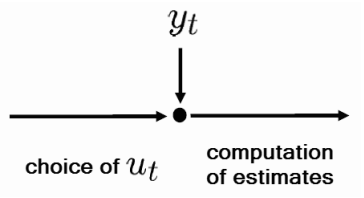

Fig. 1.9. A model of an estimation algorithm

In some experiments, the first two steps can occur simultaneously, but we have consciously chosen to separate them. We define the first step ourselves, while the second step is not controlled by us. The result of the second step is obtained as a response of the system that is external in relation to us.

For a more accurate argument from a mathematical point of view we need to specify the notion of a deterministic algorithm.

Definition 1.1. *An algorithm is called a deterministic algorithm if each of its steps defined by the user is given by deterministic rules using the results of previous steps, and obtained new data (output) is returned for use in subsequent steps.*

The above algorithm for estimating the unknown parameter x can be regarded as deterministic if we use only deterministic rules in the first and third steps.

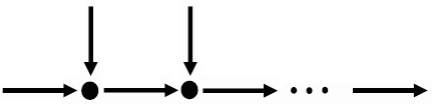

Fig. 1.10. A model of a deterministic algorithm

If in addition to the problem setting, we assume that noise v_t has a random (probabilistic) nature, we then can take $u_t \equiv 1$ and, under the conditions of the strong law of large numbers [292], we can consider estimating an unknown parameter x simply by averaging the observation data. Table 1.1 shows simulation results for $T = 7$ with real parameter $x = 3$ and observations made with noise v_t uniformly distributed over the interval $[-0.5, 0.5]$. Row 5 indicates the proximity estimates

$$\widehat{x}(7) = \mathbf{H}_7\mathbf{y} = \frac{1}{7}\sum_{i=1}^{7} y_i = 2.99$$

to the real value $x = 3$.

Table 1.1.

t	1	2	3	4	5	6	7
u_t	1	1	1	1	1	1	1
$v_t = rand() - 0.5$							
y_t	2.9	2.8	3.2	3.3	2.6	3.4	2.7
$\widehat{x}(t)$	2.9	2.85	2.97	3.05	2.96	3.03	2.99
$v_t = rand() - 0.5 + M_v,\ M_v = 1$							
y_t	3.9	3.8	4.2	4.3	3.6	3.9	4.2
$\widehat{x}(t)$	3.9	3.85	3.97	4.05	3.96	4.03	3.99

If the observations were also carried out with random noise but with the unknown expectation $M_v = Ev_t$ (e.g. $M_v = 1$, Table 1.1, row 6), the simulation results (Table 1.1, row 8) would then show that the algorithm failed: $\widehat{x}(7) = 3.99$, and that this value substantially exceeds $x = 3$.

Despite the seeming absurdity of a problem statement involving estimation under arbitrary external noise, for practical reasons such a problem often must be solved.

To get a meaningful answer to this problem we extend the class of considered algorithms by adding randomized algorithms.

Definition 1.2. *An algorithm is called a randomized algorithm when the execution of one or more steps, which have been defined by the user and referred to as randomized steps, is based on a random rule. More precisely, for each*

randomized step one rule is chosen randomly among many deterministic rules according to a probability P that is determined by the user.

Consider the following rule of random input selection for the first step

$$u_t = \bar{u}_t + \Delta_t,\ \Delta_t = \begin{cases} +1, \text{ with probability } \frac{1}{2} \\ -1, \text{ with probability } \frac{1}{2} \end{cases} \tag{1.12}$$

where $\bar{u}_t$ is a deterministic part of a control action and Δ_t is a randomization that allows the assumption

- *As1.1:* Δ_t does not depend on x and v_t.

More precisely if we carefully define the probability space and if either x or v_t are unknown but are deterministic in nature or do not depend on x or v_t in a probabilistic sense, Assumption *As1.1* holds.

At the second step we compute the value

$$\tilde{y}_t = (u_t - \bar{u}_t) \cdot y_t$$

using known values $\bar{u}_t, u_t, y_t$. This "new" sequence of observations $\{\tilde{y}_t\}$ yields a value similar to the model in (1.11):

$$\tilde{y}_t = \Delta_t^2 \cdot x + \tilde{v}_t = 1 \cdot x + \tilde{v}_t,$$

where $\tilde{v}_t = (u_t - \bar{u}_t) \cdot (\bar{u}_t x + v_t) = \Delta_t \cdot (\bar{u}_t x + v_t)$.

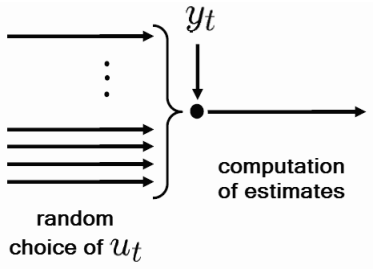

Fig. 1.11. Randomized algorithm

As in the simulation, let us suppose that v_t is a random noise but with an unknown mathematical expectation. If v_t is an external noise, it is natural to assume that this noise is independent of our randomized inputs (controls) at Step 1. Hence we have

$$E\tilde{v}_t = E\Delta_t \cdot (\bar{u}_t x + v_t) = E\Delta_t \cdot E(\bar{u}_t x + v_t) = 0 \cdot E(\bar{u}_t x + v_t) = 0,$$

i.e., for the new observation model the hard problem of estimating an unknown parameter x of (1.11) is converted by using the random selection rule

for inputs (controls) in Step 1 to the standard problem of estimating an unknown parameter x observed with an independent centered noise.

Assume for simplicity that a deterministic part of a control is absent: $\bar{u}_t \equiv 0$. Table 1.2 summarizes the corresponding simulation results for the new estimates

$$\widehat{x}(t) = \mathbf{H}_t \tilde{\mathbf{y}} = \frac{1}{t} \sum_{i=1}^{t} \tilde{y}_i. \tag{1.13}$$

Table 1.2.

t	1	2	3	4	5	6	7
u_t	-1	1	-1	1	1	1	-1
$v_t = rand() - 0.5 + M_v,\ M_v = 1$							
y_t	-2.1	3.8	-1.8	4.3	3.6	4.4	-2.3
$\tilde{y}_t$	2.1	3.8	1.8	4.3	3.6	4.4	2.3
$\widehat{x}(t)$	2.1	2.95	2.57	3.00	3.12	3.33	3.19

Comparison of the results in Table 1.2, row 6 with the previous ones from Table 1.1, row 8 shows that the new estimates are substantially better. Nevertheless, the quality of the evaluations turned out to be lower than in the more relevant results from Table 1.1, row 5, because the "new errors" $\tilde{v}_t$ have a greater variance compared to the variance of v_t.

The probability of making a wrong decision can be estimated asymptotically by assessing the correspondence mean rate of the convergence in [141] and using Chebyshev's inequality. For every t and for any $\varepsilon > 0$ we have

$$\text{Prob}\{|\widehat{x}(t) - x| \geq \varepsilon\} \leq \frac{1}{t} \frac{E v_t^2}{\varepsilon^2} + o\left(\frac{1}{t}\right). \tag{1.14}$$

For a finite number of observations ($T = 7$) under an arbitrary external noise v_t, a rigorous mathematical result of a guaranteed set of possible values of the unknown parameter x is presented in Chapter 5 based on ideas and methods described by Campi in [61]:

1. Let $m = 10$ and randomly select nine ($= m - 1$) different groups of four indexes $S_1, \ldots, S_9$.
2. Compute the partial sums $s_i = \frac{1}{4} \sum_{j \in S_i} \bar{y}_j,\ i \in 1..9$.
3. Build the confidence interval

$$\widehat{X}(7) = [\min_{i \in 1..9} s_i, \max_{i \in 1..9} s_i],$$

which contains x with the probability $p = 80\%$ $(= 1 - 2 \cdot 1/m)$.

For the data $\{u_t, y_t\}_{t \in 1..7}$ from Table 1.2 we obtain the following by the described method:

$$\mathbf{s} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \end{pmatrix} = \mathbf{H}\tilde{\mathbf{y}} = \frac{1}{4} \begin{pmatrix} 0\,1\,1\,1\,1\,0\,0 \\ 1\,0\,1\,1\,0\,1\,0 \\ 0\,1\,1\,0\,1\,1\,0 \\ 1\,1\,0\,0\,0\,1\,1 \\ 1\,0\,0\,1\,1\,0\,1 \\ 0\,1\,1\,0\,1\,0\,1 \\ 1\,0\,0\,1\,1\,0\,1 \\ 1\,1\,1\,0\,0\,1\,0 \\ 1\,0\,0\,1\,0\,1\,1 \end{pmatrix} \begin{pmatrix} -1 \cdot (-2.1) \\ 1 \cdot 3.8 \\ -1 \cdot (-1.8) \\ 1 \cdot 4.3 \\ 1 \cdot 3.6 \\ 1 \cdot 4.4 \\ -1 \cdot (-2.3) \end{pmatrix} = \begin{pmatrix} 3.375 \\ 3.15 \\ 3.4 \\ 3.15 \\ 3.075 \\ 2.875 \\ 3.275 \\ 3.025 \\ 3.275 \end{pmatrix}.$$

Hence,

- the unknown parameter x belongs to the interval $\widehat{X}(7) = [2.875; 3.4]$ with probability $p = 80\%$ (confidence parameter).

We can increase the probability $p = 80\%$ up to 90% if we consider 19 groups of four indexes ($m = 20$).

For the problem of estimating an unknown parameter under arbitrary external noise, which seems absurd and cannot be handled by any deterministic algorithm, randomization in the process of the input data selection can in principle yield quite reasonable results.

1.4.2 Randomized and Bayesian Approaches

An alternative probabilistic approach is a Bayesian estimation in which the probability of noise v_t is attributed a priori to the nature of Q. But from the practical point of view, Bayesian and randomized approaches are quite different from one another. In a Bayesian approach the probability Q describes the probability of a value of v_t in a comparison with other values. That is, the choice of Q is a part of the problem model. In contrast, the probability P in a randomized approach is selected artificially. P exists only in our algorithm, therefore eliminating the traditional problem of a bad model, as can happen with Q in the Bayesian approach.

1.5 Randomization for Reducing Computational Complexity

Historically, randomized algorithms were first introduced as way to reduce computational complexity. Many researchers have studied their relatively better performance compared to the more traditional deterministic approach.

Suppose that a deterministic algorithm requires a huge amount of computing resources to process all available data. Then we can intentionally give up part of the data and proceed to solve the simplified problem with a partial set of data. In this case, however, deterministic solvability may be impossible to achieve, but as above we can consider a randomized approach to define

solutions with a high probability of a success. The end result is a compromise between the full guarantee of success and computational feasibility (the opportunity to get a real answer for a limited time).

Note that the reason for using a randomized method remains the same: an insufficient amount of data, albeit by artificial means. It is possible to collect all the data and then to intentionally discard some of it to increase computational feasibility, thus making the data insufficient to solve the problem by traditional deterministic means.

Within this general idea impressive results have been obtained for a randomized approach in key areas of control theory and systems. For example, the approach has been applied to convex problems of robust control, including but not limited to a wide class of control problems that can be reduced to parameter-dependent linear matrix inequalities (LMIs). Many of these problems are NP-hard in the deterministic setting (see, e.g [46, 53, 245, 332]).

The success of randomized algorithms is promoted by the fact that their performance and probability of success can be estimated with sufficient accuracy by analytical methods, for example, by Chebyshev's [292] or Hoeffding's inequality [166].

1.6 Quantum Computing

Until recently quantum computing was regarded exclusively as a notional mathematical model. Of course, serious difficulties are still encountered in designing a quantum computer for everyday use. Nonetheless, intensive research and development projects are continuing in this field.

The representation of the randomized algorithm on Figure 1.11 is associated with something well known to those familiar with the fundamentals of quantum computing. Virtually all known effective quantum algorithms implement a similar scheme:

- preparing input "superposition",
- processing,
- measuring a result.

We now briefly outline the mathematic model of a quantum computer. States in quantum mechanics are often denoted by vectors of unit length in a vector space over a field of complex numbers. Observed quantities are represented by self-conjugate operators in this complex space [293]. An observed quantity is a method of obtaining information on the state. Measurement of the quantum system changes this information. By measurement, we obtain one of the eigenvalues of the observed quantity, and the system itself passes to the eigenvector corresponding to this eigenvalue. In measurement, an eigenvalue is chosen at random with a probability equal to the square of the projection of the state on the corresponding eigenvector. Clearly, measurement of the quantum system yields complete information only if the state

of the system coincides with one of the eigenvectors of the chosen observed quantity.

A quantum computer processes q-bits (quantum bits), which form a quantum system of two states (a microscopic system corresponding to the description of an excited ion or a polarized photon, or spin of the atomic nucleus). Mathematically, a q-bit takes its values from a complex projective (Hilbert) space $\mathbb{C}^2$. Quantum states are invariant to multiplication by a scalar. The basis of the q-bit state space is usually denoted by $|0\rangle$ and $|1\rangle$, corresponding to the notation $\{0, 1\}$ used in classical information theory. Not only can such a system exist in base states, it is also capable of storing more information than the corresponding classical system. Nevertheless, it passes into one of the base states during measurements (if the observed system is properly chosen), and the information it contains corresponds to some classical information. We assume that a quantum computer is equipped with a discrete set of basic components, called quantum circuits. Every quantum circuit is essentially an unitary transformation that acts on a fixed number of q-bits. One of the fundamental principles of quantum mechanics asserts that the joint quantum state space of a system consisting of r two-state systems is the tensor product of its component state spaces. Thus, the quantum state space of a system of r q-bits is the projective Hilbert space $\mathbb{C}^{2^r}$. A set of basis vectors of this state space can be parameterized by bit rows of length r. For example, for $r = 3$, the base vector $|0\rangle \otimes |0\rangle \otimes |0\rangle$ can be denoted by $|000\rangle$. Sometimes it is more convenient to use another form of expression $|0\rangle|0\rangle|0\rangle$ or $|00\rangle|0\rangle$. Therefore, the dimension of the space that a quantum computer uses grows exponentially with the number of q-bits. This property underlies the notion of quantum parallelism.

Let us assume that classical data, a row i of length l, $l \leq r$, is fed as input into a quantum computer. In quantum computation, r q-bits initialize in the state $|i00\ldots0\rangle$. An executable circuit is constructed from a finite number of quantum circuits acting on these q-bits. At the end of computation, the quantum computer passes into some state that is a unit vector in the space $\mathbb{C}^{2^r}$. This state can be represented as

$$\mathbf{x} = \sum_e x_e |e\rangle,$$

where summation with respect to e is taken over all binary rows of length r, $x_e \in \mathbb{C}$, $\sum_e |x_e|^2 = 1$. Here x_e are called probabilistic amplitudes and $\mathbf{x}$ is called a superposition of basis vectors $|e\rangle$. The Heisenberg uncertainty principle asserts that the state of a quantum system cannot be predicted exactly. Nevertheless, there are several possibilities for measuring all q-bits (or a subset of q-bits). To do this, the observation concept is defined as an operator in the state space. The state space of our quantum system is Hilbertian and an observed operator $\mathbf{U}$ is equivalent to the scalar product in this Hilbert space with some given vector $\mathbf{u}$:

$$\mathbf{U}|\mathbf{x}\rangle = \langle \mathbf{u}, \mathbf{x} \rangle.$$

The projection of each q-bit on the basis $\{|0\rangle, |1\rangle\}$ is usually used in measurement. The result of this measurement is the computation result.

Another important property of quantum states is their unique evolution. In other words, every transformation of q-bits in a quantum computer is a unitary operator in the corresponding complex space. Hence, every transformation of information (except for measurements) in a quantum computer must be invertible.

The simplest example of the typical efficiency of quantum computing is Deutsch's problem solving [93].

Let a function $f : \{0,1\} \longrightarrow \{0,1\}$ be defined as a black box and the process of its computation continues for 24 hours.

The following question *must be answered*: Is the function $f(u)$ constant or balanced?

In the classical case, obviously not less than 48 hours are necessary to answer the question.

Let the given quantum black box compute $f(u)$. More precisely, let us define two q-bit unitary transformations:

$$\mathbf{U}_f : |u\rangle|z\rangle \longrightarrow |u\rangle|z \oplus f(u)\rangle,$$

which flip the second q-bit if the value of f from the first q-bit is 1. Our task is to determine whether or not $f(0) = f(1)$. If we are limited to classical inputs $|0\rangle$ and $|1\rangle$, we must call the box twice ($u = 0$ and $u = 1$) in order to get the answer. But if we are allowed to introduce a coherent superposition of these classical states, calling the box one time is sufficient to answer the question!

In quantum computing the Hadamard transformation plays a special role. It is defined by the formula

$$\mathbf{H} : |u\rangle \longrightarrow \frac{1}{\sqrt{2}} \sum_z (-1)^{uz} |uz\rangle, \tag{1.15}$$

or $\mathbf{H} : \begin{pmatrix} |0\rangle \\ |1\rangle \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{pmatrix}$, that is $\mathbf{H} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$.

The Hadamard transform is used to prepare the superposition of the input data. In a certain sense, this transform can be interpreted as randomized inputs.

To solve Deutsch's problem on a one-time basis (using only one function evaluation), we consider the quantum circuit of Figure 1.12.

If the input of the circuit is a couple of quantum bits $|0\rangle|1\rangle$ then we obtain successively

$$|0\rangle|1\rangle \longrightarrow \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)\frac{1}{2}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle](|0\rangle - |1\rangle) \longrightarrow$$

$$\longrightarrow \frac{1}{2}\left[\left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle\right]\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

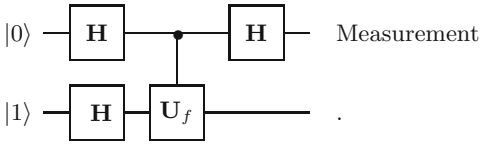

Fig. 1.12. Quantum algorithm

Hence, when we measure the first q-bit we obtain the result $|0\rangle$ with probability one if $f(0) = f(1)$ (that is, f is a constant function) and the result $|1\rangle$ with probability one if $f(0) \neq f(1)$ (balanced function).

This shows the advantage of quantum over classical computation due to quantum parallelism. As we feed the input entangling $|0\rangle$ and $|1\rangle$, the output is sensitive to both the values $f(0)$ and $f(1)$ even though we called the black box only once.

Note that not only are the schemes of the quantum algorithm in Figure 1.12 and randomized algorithm in Figure 1.11 similar, their internal content is similar as well.

2

Historical Overview

2.1 Game Theory

The minimax theorem is one of the first positive theoretical answers for the question: *Why should randomization be beneficial?* The theorem was proved by John von Neumann in 1928 [343]. He considers two-person zero-sum games defined by a matrix $\mathbf{A} = \{a_{i,j}\}$ with perfect information (i.e., in which each time players know all the moves that have taken place so far). In these games, $a_{i,j}$ is a win (profit) for the first player (a loss for the second player) when the first player has chosen row i and the second player has chosen column j.

As a rule, there is no optimal pure strategy $i^\star, j^\star$ such that

$$\max_i \min_j a_{i,j} = \min_j \max_i a_{i,j} = a_{i^\star,j^\star}.$$

But this is possible for a more wide class of randomized (mixed) strategies. For each player a randomized (mixed) strategy is to choose a set of probabilities for his decision-making rules (p_i for the first player choosing row i, $\sum_i p_i = 1$, or q_i for the second player choosing column j, $\sum_j q_j = 1$,). According to the minimax theorem [343], there exists a pair of randomized (mixed) strategies for both players $\{p_i^\star\}, \{q_j^\star\}$ that allows each player to minimize his maximum losses

$$\max_{\{p_i\}} \min_{\{q_j\}} \sum_{i,j} p_i q_j a_{i,j} = \min_{\{q_j\}} \max_{\{p_i\}} \sum_{i,j} p_i q_j a_{i,j} = \sum_{i,j} p_i^\star q_j^\star a_{i,j}$$

(hence the name, *minimax*).

Here, the role of randomness is to hide our strategy. In 1944 von Neumann improved and extended the minimax theorem to include games involving imperfect information and games with more than two players, publishing this result in his *Theory of Games and Economic Behavior* (written with Oskar Morgenstern) [344]. Public interest in this work was such that *The New York Times* ran a front-page story about it. In the book, von Neumann declared that in applying economic theory to minimax problems functional analytic

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_2, © Springer-Verlag Berlin Heidelberg 2015

methods must be used, especially convex sets and topological fixed point theorem, rather than the traditional differential calculus, since the maximum operator does not preserve differentiable functions. From the point of view of mathematical formulations of problems, minimax problems have stimulated significant development of new methods of ℓ_1-optimization, while traditional ℓ_2-optimization approaches are sufficient for many problems formulated in terms of traditional differential calculus. The inability to obtain analytical solutions for many important problems stimulated the active development of modern computer technology, and this development originated with the same person — John von Neumann.

2.2 Monte Carlo Method, Random Search

Randomized approaches began being widely used with the appearance of the Monte Carlo (MC) method of statistical simulation. This method was proposed by Metropolis and Ulam [234] during their work on the Manhattan Project at Los Alamos with von Neumann and Teller. The MC method offers a simple scheme to estimate the mean value (expectation) of a function based on samples of its values for some random arguments (randomization). Originally a randomized approach was used to approximate the multidimensional integration in transport equations arising in connection with the problem of the motion of a neutron in an isotropic medium.

More precisely, if the following formula is difficult to integrate analytically

$$F = \int \cdots \int_{\mathbb{W}} f(\mathbf{w}) d\mathbf{w},\ \mathbb{W} \subset \mathbb{R}^r,$$

then the MC method is easier to use.

Let the set $\mathbb{W}$ be a parallelepiped with the volume $Vol(\mathbb{W})$, and function f is bounded: $0 \leq f(\mathbf{w}) \leq f_{\max}$. The geometric sense of the integral is a volume below the graph of function f.

The MC method for this problem is as follows:

1. Fix a positive integer T.
2. Choose uniformly T independent identically distributed samples $z_1, z_2, \ldots, z_T$ from the parallelepiped $\mathbb{W} \times [0, f_{\ max}] \subset \mathbb{R}^{r+1}$.
3. Among the samples $z_1, z_2, \ldots, z_T$ count the number S of those z_i, $i \in 1..T$, whose last component does not exceed the value of the function f in the corresponding point defined by the first r coordinates.
4. Compute the estimate

$$\widehat{F}(T) = \frac{S}{T} Vol(\mathbb{W}) f_{\max}.$$

The estimate $\widehat{F}$ is sufficiently good for the value F of the integral: $\widehat{F} \approx F$. The needed accuracy of the approximation can be guaranteed *a priori* (when choosing T) by virtue the law of large numbers [292] and the Hoeffding's inequality [166]

$$\forall \varepsilon > 0 \ \text{Prob}\left\{\left|\frac{S}{T} - \frac{F}{Vol(\mathbb{W}) f_{\max}}\right| \geq \varepsilon\right\} \leq 2e^{-2T\varepsilon^2}.$$

When the size r of the set $\mathbb{W}$ is small, in many cases it is possible to calculate an integral and to use traditional deterministic approximation methods. But when $r \gg 1$, the Monte Carlo method is preferable in practice since its complexity does not depend on a state-space size r.

It is not accidental that the invention and development of the Monte Carlo method coincided with the creation of the first computers. Even the earliest computers could rapidly generate pseudo-random numbers. A more detailed description of the MC method can be found, for example, in [103].

Today MC methods are used extensively in many applications, among them continuous and discreet optimizations, data analysis, signal and image processing, machine learning, classification and recognition problems, control (e.g., the generation of stabilizing controllers), robustness (generate uncertainties) and many others.

The key problem is how to uniformly generate points in a given set. Explicit algorithms for simple sets (e.g., boxes, balls, simplices) are included in many software toolboxes. For example, the Matlab function $rand(r+1;1)$ can be used for the uniform generation of points in the cube $[0,1]^{r+1} \subset \mathbb{R}^{r+1}$, and $randn(r+1;1) \sim \mathcal{N}(0, \mathbf{I})$ can be used for the normal (Gaussian) distribution in the space $\mathbb{R}^{r+1}$. For the closed subset $\mathbb{P} \subset \mathbb{R}^{r+1}$ it is easy to use the rejection method when we take the simple set $\mathbb{S} \supset \mathbb{P}$, that contains $\mathbb{P}$, generate points in $\mathbb{S}$ and reject those not in $\mathbb{P}$. However these methods are ineffective for important cases of huge size $r \gg 1$. The general technique is the random walk technique, i.e. Markov Chain Monte Carlo (see, e.g. [96]). Among these, the *Hit-and-Run method* is the most popular for a high size convex set $\mathbb{P} \subset \mathbb{R}^{r+1}$ [296, 322]. Another promising technique is the *random billiard walks* technique [260].

2.2.1 Random Search, Simulating Annealing, Genetic Algorithms

A natural consequence of the Monte Carlo approach was the appearance of broad classes of *random search* (RS) algorithms [271, 358] that can solve the unconditional optimization problem

$$F(\mathbf{x}) \to \min_{\mathbf{x} \in \mathbb{R}^d}.$$

RS works by iteratively moving to better positions in the search-space. These positions are sampled from a hypersphere surrounding the current

position until a termination criterion is met (e.g., number of iterations performed or adequate fitness reached). The basic RS algorithm scheme is

1. Initialization $n = 0$. Choose $\widehat{\mathbf{x}}(0)$ as a random vector in the search-space $\mathbb{R}^d$ and obtain the value $F(\widehat{\mathbf{x}}_0)$ of the function $F(\cdot)$ at the point $\widehat{\mathbf{x}}(0)$.
2. Iteration $n \to n + 1$.
   2a. $n := n + 1$.
   2b. Generate the random direction $\boldsymbol{\Delta}_n$ according to the uniform distribution on a unit sphere in $\mathbb{R}^d$ as follows:
   - $\widehat{\mathbf{z}}_n = randn(d; 1)$;
   - $\boldsymbol{\Delta}_n = \frac{1}{||\widehat{\mathbf{z}}_n||}\widehat{\mathbf{z}}_n$.

   2c. Choose a step-size β_n and to compute the next (input) measurement point
   $$\mathbf{u}_n = \widehat{\mathbf{x}}(n-1) + \beta_n \boldsymbol{\Delta}_n.$$
   2d. Obtain the new value $F(\mathbf{u}_n)$ of the function $F(\cdot)$ at the point $\mathbf{u}_n$.
   2e. Compare the new value $F(\mathbf{u}_n)$ with previous one $F(\widehat{\mathbf{x}}(n-1))$ and to change the current estimate when the new value is better
   $$\widehat{\mathbf{x}}(n) = \begin{cases} \mathbf{u}_n, & \text{if } F(\mathbf{u}_n) < F(\widehat{\mathbf{x}}(n-1)); \\ \widehat{\mathbf{x}}(n-1), & \text{otherwise.} \end{cases}$$
3. Repeat Step 2 or to terminate the algorithm.

In many practical problems, the function $F(\cdot)$ is not given analytically. Its values may be the result of physical experiments or complicated computations.

RS is often used for optimization of the function

$$F(\mathbf{x}) = \int_{\mathbb{W}} f(\mathbf{x}, \mathbf{w}) d\mathbf{w},$$

which is computed by MC simulation. RS algorithms do not require the gradient of the function $F(\cdot)$ to be optimized, and RS can hence be used on functions that are not continuous or differentiable. Such numerical optimization methods are also known as direct-search, derivative-free, or black-box methods.

The main best features of RS algorithms are

- Easy, suitable for a task of enormous size.
- Do not need to calculate the gradient.
- Convergence to the global minimum.

Recently global optimization has often been solved using different metaheuristic methods that are also based on randomized rules. Among them are simulated annealing [193, 235], genetic algorithms [168], ant colony optimization [82], tabu search [128] and others as well.

2.2.2 Probabilistic Methods in a Control Syntheses

Many practical problems can be reformulated as a robust convex optimization problem (*Robust Convex Problem, RCP*), where a convex function has to be optimized under restrictions that are also given by convex functions.

Let $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^d$ be a *design parameter* of a system or a control plant. Consider a family of convex optimization problems parameterized by $\mathbf{w} \in \mathbb{W} \subseteq \mathbb{R}^r$:

$$RCP : \min_{\mathbf{x}\in\mathbb{X}} \langle \mathbf{c}, \mathbf{x} \rangle \ \text{ subject to } \ \{\mathbf{x} : g(\mathbf{x}, \mathbf{w}) \leq 0, \forall \mathbf{w} \in \mathbb{W}\},$$

where $\mathbb{X} \subseteq \mathbb{R}^d$ is a convex closed set, $\mathbf{c} \in \mathbb{R}^d$ is a given vector, the function $g : \mathbb{X} \times \mathbb{W} \to \mathbb{R}$ is a convex on $\mathbf{x}$ for all $\mathbf{w} \in \mathbb{W}$. Typically, the cardinality of the set $\mathbb{W}$ can be infinite: $|\mathbb{W}| = \infty$, that is, it can contain an infinite number of elements of $\mathbf{w}$.

The important special cases of RCP are robust linear programming problem in which $g(\mathbf{x}, \mathbf{w})$ is affine in $\mathbf{x}$, and the semi-definite programming where $g(\mathbf{x}, \mathbf{w}) = \lambda_{\max}(\mathbf{A}(\mathbf{x}, \mathbf{w}))$ and $\mathbf{A}$ a symmetric matrix such that

$$\mathbf{A}(\mathbf{x}, \mathbf{w}) = \mathbf{A}_0(\mathbf{w}) + \sum_{i=1}^{d} x_i \mathbf{A}_i(\mathbf{w}), \ \mathbf{A}_i(\mathbf{w}) = \mathbf{A}_i^{\mathrm{T}}(\mathbf{w}).$$

On the one hand, the traditional *design for "worst case"* leads to the necessity of solving NP-hard problems. On the other hand, it often introduces excessive conservatism since it forces the design to focus on all degenerate cases, even those with a low probability of appearance.

Randomization can be applied for solving minimax problems provided that the requirement "max" may be weakened in a probabilistic sense if one can be satisfied with a problem solution that ignores some restrictions.

Vidyasagar [332] proposed that randomized methods inspired by the MC method be used for robust control problems. This introduced a completely new area of applying randomized-based methods for control synthesis in the standard min-max robust control formulation. The task of optimizing the choice of controller is reformulated as a "chance-constrained" problem that can be solved by randomization.

The *probabilistic (randomized) robust design* paradigm assumes a probability measure on the set of uncertainties $\mathbb{W}$. Then for a given level of probability $\varepsilon \in (0, 1)$ the RCP is reformulated as a design of $\mathbf{x} \in \mathbb{X}$, which minimizes the objective function $\langle \mathbf{c}, \mathbf{x} \rangle = \mathbf{c}^{\mathrm{T}}\mathbf{x}$ under almost all restrictions except for a small part of them whose measure (probability) does not exceed a predetermined level ε. The randomized approach allows for setting the level ε and confidentiality $\beta \in (0, 1)$ parameters a priori to obtain a solution. This is a solution of the optimization problem with the given probability $(1 - \beta)$ that satisfies almost all restrictions, except for a set of measure that are at most ε.

Assume we have T independent identically distributed samples $\mathbf{w}_1, \mathbf{w}_1, \ldots,$ $\mathbf{w}_T$. A *scenario design problem* is a convex problem

$$RCP_T : \min_{\mathbf{x}\in\mathbb{X}} \langle \mathbf{c}, \mathbf{x} \rangle, \text{ subject to } \{\mathbf{x} : g(\mathbf{x}, \mathbf{w}_i) \leq 0,\ i \in 1..T\}$$

which is a discretization of RCP.

Fix two real numbers $\varepsilon \in (0, 1)$ (level parameter) and $\beta \in (0, 1)$ (confidence parameter). Calafiore and Campi [57] obtain that under sufficiently general conditions if T is choosing in such a way

$$T \geq \frac{2}{\varepsilon} \ln \frac{1}{\beta} + 2r + \frac{2r}{\varepsilon} \ln \frac{2}{\varepsilon} \tag{2.1}$$

then, with probability no smaller than $(1 - \beta)$, either the scenario problem RCP_T is unfeasible and, hence, also the initial robust convex program is unfeasible; or, RCP_T is feasible, and then its optimal solution $\widehat{\mathbf{x}}$ is ε-level robustly feasible in the sense that $\text{Prob}\{\mathbf{w} : g(\widehat{\mathbf{x}}, \mathbf{w}) > 0\} \leq \varepsilon$. The most outstanding feature of (2.1) is that parameter β shows up under the sign of logarithm so that it can be made very small (10^{-10} or even 10^{-20}) without significantly increasing T.

The scenario approach can be used for detecting faults or unconventional behavior in a wide class of systems. This problem can be reformulated as the detection of function discontinuity. One possible way to solve the problem is described in [149] based on a scenario approach to the detection of discontinuity.

The reviews by Tempo, Calafiore, and Dabbene [58, 312] offer a broad survey of randomized methods in control systems as outlined in the discussed through this book general framework. Though we do not claim this is a complete listing, we additionally note the following papers in this area [27, 59, 124, 171, 181, 190, 250, 261, 262, 304].

2.3 Estimation and Filtering under Arbitrary External Noises

2.3.1 Randomized Stochastic Approximation

Surprisingly, for a long time researchers did not notice that in cases of noisy observations, search algorithms with sequential ($n = 1, 2, \ldots$) changes in the estimate $\widehat{\mathbf{x}}(n-1)$ along some random centered vector $\boldsymbol{\Delta}_n$

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \boldsymbol{\Delta}_n y_n,$$

may converge to the true vector of controlled parameters $\mathbf{x}^\star$ not only under "good" but also under "almost arbitrary" disturbances. This happens if observations y_n are taken at some point defined by the previous estimate

$\widehat{\mathbf{x}}(n-1)$ and by the randomized vector $\boldsymbol{\Delta}_n$ which is called the simultaneous test perturbation (disturbance). Such algorithms are called the *randomized estimation algorithms*. Their convergence under "almost arbitrary" noises is demonstrated through the stochastic (probabilistic) properties of the test perturbation. In the near future, experimenters are likely to radically change their current and sometimes cautious attitude to stochastic algorithms and their results. Modern computing devices will be supplanted by quantum computers, which due to the Heisenberg principle of uncertainty operate as stochastic systems. By virtue of the possibility offered by quantum parallelism, randomized-type estimation algorithms will most probably form the underlying principle of future quantum computing devices. For a detailed analysis of the possibilities of randomized algorithms in problems of estimation and optimization under arbitrary noise, see the book by Granichin and Polyak [148].

Stochastic approximation was introduced in an article published by Robbins and Monro [273] in the *Annals of Mathematical Statistics* in 1951 and was further developed for optimization problems by Kiefer and Wolfowitz [192]. This approximation method was originally introduced as a tool for statistical computations and was further developed within the separate field of control theory. Today this topic has a wide variety of applications in areas such as adaptive signal processing, adaptive resource allocation in communication networks, system identification and adaptive control.

Kiefer and Wolfowitz considered the problem of iterative determination of the stationary point $x^\star$ (point of a local minimum or maximum) of some twice continuously differentiable function $F(\cdot)$, when for each chosen value $u \in \mathbb{R}$ (which is called an input) one can observe the corresponding function $F(\cdot)$ value at the point u with random noise

$$y(u) = F(u) + \text{noise}.$$

In the noise-free case, the problem is to find a point $x^\star$ such that

$$\nabla F(x^\star) = 0.$$

The best iterative solution is the famous *Newton-Raphson method*:

1. Initialization $n = 0$. Choose $\widehat{x}(0) \in \mathbb{R}$.
2. Iteration: $n := n + 1$,

$$\widehat{x}(n) = \widehat{x}(n-1) - [\nabla^2 F(\widehat{x}(n-1))]^{-1} \nabla F(\widehat{x}(n-1)).$$

3. Repeat Step 2 or terminate the algorithm.

The more simple iteration procedure

$$\widehat{x}(n) = \widehat{x}(n-1) - \alpha \nabla F(\widehat{x}(n-1))$$

also works with a sufficiently small positive *step-size* parameter α when $\nabla F(x)$ is bounded and $\nabla^2 F(x) > 0$ (see, e.g., detailed analysis in [258]).

To solve the problem with random noise under some additional constraints, Kiefer and Wolfowitz [192] proved that for any $\widehat{x}(0) \in \mathbb{R}$ the recurrent sequence obeying the rule (Kiefer-Wolfowitz (KW) procedure)

$$u_n^{\pm} = \widehat{x}(n-1) \pm \beta_n,$$

$$\widehat{x}(n) = \widehat{x}(n-1) - \alpha_n \frac{y(u_n^+) - y(u_n^-)}{2\beta_n},$$

where $\{\alpha_n\}$ and $\{\beta_n\}$ are some given positive decreasing numerical sequences:

$$\sum_{n=1}^{\infty} \alpha_n = \infty, \ \sum_{n=1}^{\infty} \alpha_n^2 \beta_n^{-2} < \infty,$$

converges to the point $x^\star$. The main requirement is that the observation noise is conditional zero-mean. This requirement is usually assumed to be satisfied and can be formulated as follows: for a small β the conditional expectation of *statistics*

$$G(x, \beta) = \frac{y(x+\beta) - y(x-\beta)}{2\beta},$$

whose sampled values are precisely observed or calculated, is close to the gradient (derivative) of the function $F(\cdot)$

$$EG(x, \beta) \approx \nabla F(x).$$

The behavior of the sequence of estimates determined by the algorithm of stochastic approximation (SA) depends on the choice of the observed statistic functions $G(x, \beta)$. The convergence rate of SA algorithms estimates seems to be a main stimulus for modifying the original algorithms. Many books have closely examined the properties and generalizations of estimates of the KW procedure [49, 205, 206, 247, 258, 302, 323, 349]. The estimates convergence rate depends on the smoothness of the function $F(\cdot)$. If the function $F(\cdot)$ is twice differentiable, the mean-square error of the conventional KW algorithm decreases as $\mathcal{O}(n^{-\frac{1}{2}})$. If the function $F(\cdot)$ is three times differentiable, it decreases as $\mathcal{O}(n^{-\frac{2}{3}})$ [349]. Fabian [107] modified the KW procedure in that besides an approximation of the first derivative he used higher-order finite-difference approximations with certain weights. If the function $F(\cdot)$ has ℓ continuous derivatives, the Fabian algorithm provides the mean-square convergence rate of the order $\mathcal{O}(n^{-\frac{\ell-1}{\ell}})$ for odd ℓ. In computational terms, Fabian's algorithm is overcomplicated. The number of observations at each iteration grows rapidly with increasing smoothness and dimensionality. Moreover, at each step one has to invert the matrix. Polyak [259] and Juditsky [179] proposed improving the process of convergence by using averaging if a sequence of estimates converges to the desired point of $x^\star$.

In many applications, knowledge about the statistical characteristics of the measurement noises may be insufficient. For example, noises may be the values of an unknown deterministic function. In this case, appreciable difficulties

are encountered in motivating the applicability of the conventional Kiefer-Wolfowitz procedure, whose estimate often does not converge to the desired point. This is not to suggest, however, that in dealing with these problems one must abandon the easily representable SA algorithms. The observation can be enriched by adding a new random simultaneous perturbation Δ into the algorithm and observation channel, e.g. Δ is an observed realization of the Bernoulli random variable which is equal to ± 1 with the same probability. We modify the KW procedure by using the new *randomized statistics*

$$\tilde{G}(x, \beta, \Delta) = G(x, \beta\Delta)$$

instead $G(x, \beta)$.

In many practical cases, we can assume that Δ is independent of an observation noise, and, using the Taylor expanding formula for the function $F(\cdot)$ for new statistics $\tilde{G}(x, \beta, \Delta)$ we obtain

$$E\tilde{G}(x, \beta, \Delta) = E\frac{y(x + \beta\Delta) - y(x - \beta\Delta)}{2\beta\Delta} = \nabla F(x) + E\frac{1}{2\beta\Delta}(v^+ - v^-) + \mathcal{O}(\beta) =$$

$$= \nabla F(x) + \frac{1}{2\beta}E\frac{1}{\Delta}E(v^+ - v^-) + \mathcal{O}(\beta) = \nabla F(x) + \mathcal{O}(\beta).$$

For a sufficiently small β this means that the statistics $\tilde{G}(x, \beta, \Delta)$ is a "good" approximation "in the mean sense" of the gradient (derivative) of the function $F(\cdot)$. A simpler statistics

$$\bar{G}(x, \beta, \Delta) = \frac{\Delta}{\beta} y(x + \beta\Delta),$$

that uses only one observation at each iteration (step) has the same property. This statistics was used in [134] for constructing a sequence of consistent estimates for a dependent observation noise (almost arbitrary noise). The essence of the perturbation Δ is exciting because it is used mostly to make non-degenerate observations.

In [47], the stochastic approximation algorithm was extended to the multidimensional case. When $\mathbf{x} \in \mathbb{R}^d$, the conventional KW-procedure based on finite-difference approximations of the function gradient vector uses $2d$ observations at each iteration to construct the sequence of estimates (two observations for approximations of each component of the gradient d-vector). Let $\boldsymbol{\Delta}$ be a Bernoulli random d-vector consisting of ± 1. The randomized statistic $\bar{G}(\mathbf{x}, \beta, \boldsymbol{\Delta})$ uses a computationally simpler procedure with only one measurement of the function $F(\cdot)$ [135, 263, 301]. The generalization of the $\tilde{G}(x, \beta, \boldsymbol{\Delta})$ to the multidimensional case is

$$\tilde{G}(\mathbf{x}, \beta, \boldsymbol{\Delta}) = \begin{pmatrix} \frac{1}{\Delta_1} \\ \frac{1}{\Delta_2} \\ \vdots \\ \frac{1}{\Delta_d} \end{pmatrix} \frac{y(\mathbf{x} + \beta\boldsymbol{\Delta}) - y(\mathbf{x} - \beta\boldsymbol{\Delta})}{2\beta}.$$

This was used by Spall [300], who suggested *a simultaneous perturbation stochastic approximation (SPSA)* algorithm. He demonstrated that for a large d the probabilistic distribution of appropriately scaled estimation errors is approximately normal. He used the formula obtained for the asymptotic error variance and a similar characteristic of the KW procedure to compare overall performances of algorithms. He found out that, all other things being equal, the SPSA algorithm has the same order of convergence rate as the KW-procedure, even though in the multidimensional case (even $d \to \infty$) appreciably fewer (by the factor of d) observations are used.

In [9, 223], algorithms similar to SPSA were suggested to use for neuron network training because they can be implemented by a small set of simple logic components.

If randomized SA algorithms are used in problems with sufficiently smooth functions $F(\cdot)$, the asymptotic mean-square convergence rate can be increased without increasing the number of function measurements at each iteration. In cases where some generalized smoothness index of function $F(\cdot)$ is equal to γ ($\gamma = \ell + 1$ if all partial derivatives of orders up to ℓ inclusive satisfy the Lipschitz condition), Polyak and Tsybakov [263] proposed to use statistics

$$\tilde{G}_\gamma(\mathbf{x}, \beta, \mathbf{\Delta}) = \mathbf{K}_\gamma(\mathbf{\Delta}) \frac{y(\mathbf{x} + \beta\mathbf{\Delta}) - y(\mathbf{x} - \beta\mathbf{\Delta})}{2\beta} \tag{2.2}$$

and

$$\bar{G}_\gamma(\mathbf{x}, \beta, \mathbf{\Delta}) = \frac{1}{\beta} \mathbf{K}_\gamma(\mathbf{\Delta}) y(\mathbf{x} + \beta\mathbf{\Delta}), \tag{2.3}$$

where $\mathbf{K}_\gamma(\cdot)$ is a vector-function with a finite support (a differentiable kernel) determined by orthogonal Legendre polynomials of a degree smaller than γ. The corresponding randomized SA algorithms provide the mean-square convergence rate $\mathcal{O}(n^{-\frac{\gamma-1}{\gamma}})$ of a sequence of estimates. They also demonstrate that for a wide class of iterative algorithms this convergence rate is optimal in some asymptotically minimax sense, that is, it cannot be improved either for any other algorithm or for any other admissible rule for choosing the measurement points. (For odd ℓ this fact was established earlier by Chen [75]).

The algorithm with more general view of a gradient vector stochastic approximation

$$\widehat{G}_\gamma(\mathbf{x}, \beta^+, \beta^-, \mathbf{\Delta}) = \mathbf{K}_\gamma(\mathbf{\Delta}) \frac{y(\mathbf{x} + \beta^+\mathbf{\Delta}) - y(\mathbf{x} - \beta^-\mathbf{\Delta})}{\beta^+ + \beta^-} \tag{2.4}$$

will be considered in the Chapter 3. It was motivated by practical applications. The partial case with $\mathbf{K}_\gamma(\mathbf{\Delta}) \equiv \mathbf{\Delta}$ was proposed by Chen, Duncan and Pasik-Duncan [76] with $\beta^- \equiv 0$ and in [12] with some β^+ and β^-.

The algorithms which are based on statistics (2.2)–(2.4) have significant advantages:

- the asymptotically optimal rate of convergence;
- the minimum number of measurements within the current iteration;
- the consistency with almost arbitrary interference;
- the operability in nonstationary problems;
- their "natural" implementation on a quantum computer.

These properties have been established in practice and theoretically substantiated. The consistency of randomized algorithms of stochastic approximation in the multidimensional case under almost arbitrary noise has been established in [135, 264] and [76].

Polyak and Tsybakov [263] and Spall [301] noted that algorithms with one measurement asymptotically behave worse than those with two measurements. This is not quite true if one compares the algorithms in terms of the number of iterations multiplied by the number of measurements. Moreover, in many applications, such as the optimization of real-time systems, the dynamic processes underlying the mathematical model can be too fast to enable two successive measurements. In some problems, at one step of the algorithm it is virtually impossible to make two measurements such that the observation errors are uncorrelated with $\mathbf{\Delta}$ at both fixed points $\widehat{\mathbf{x}}(n-1)+\beta_n\mathbf{\Delta}$ and $\widehat{\mathbf{x}}(n-1)-\beta_n\mathbf{\Delta}$, yet this is one of the main conditions for applying the algorithm! The algorithm proposed in [76], which makes two consecutive observations at points $\widehat{\mathbf{x}}(n-1)$ and $\widehat{\mathbf{x}}(n-1)+\beta_n\mathbf{\Delta}$, sometimes allows avoiding the last problem.

The studies reported in [138] extend the convergence study of the randomized SA algorithm with one observation under almost arbitrary noise extended to the case of additional multiplicative noise and weaken the dependence between perturbation $\mathbf{\Delta}$ and observation noises.

In [281], the authors show that the optimal distribution for the components of $\mathbf{\Delta}$ is the symmetric Bernoulli distribution. The effectiveness of this simple distribution is also confirmed by many practice examples for the finite sample of observations. Sometimes choosing a different distribution in practical problems is desirable. For example, in [224] the robot controller uses a symmetric uniform distribution consisting of two parts when the neighborhood of zero is removed.

Deterministic convergence analysis of randomized algorithms of stochastic approximations can be found in [76] and [348]. The rate of algorithm convergence is examined in [126, 140].

Stochastic approximation algorithms were initially proven in case of the stationary functional. In [258] for time-varying functionals, the Newton method and gradient method are applied to problems of minimization but they are applicable only in case of two times differentiable functional and with known bounds of Hessian matrix. Both methods require possibility of a direct measurement of a gradient in an arbitrary point. The stochastic setting is not discussed there. The books [49, 206] use the ordinary differential equations (ODE) approach to describe stochastic approximation. It addresses

the issue of applications of stochastic approximation to tracking and time-varying systems. Randomized SA algorithms are considered for the case of time-varying quality functionals in [144, 326].

Most stochastic approximation algorithm applications are concerned with adaptive systems based on the fact that SA algorithms have properties useful for uncertain environments. These important properties allow these algorithms to track the typical behavior of such systems. Furthermore, these algorithms are memory and computationally efficient, making them applicable to real time dynamic environments. Due to these properties, the algorithms are applicable in such new fields as soft computing, where they are used for "parameter tuning". Notable among these are algorithms for neural network training and for reinforcement learning. They are used in popular learning paradigms for autonomous software agents, with applications in e-commerce, robotics and other fields. They are also widely applied in economic theory, providing a good model for collective phenomena when the algorithms are used to model the behavior of individual bounded rational agents. A randomized algorithm for stochastic approximation in the machine self-learning problem is proposed and justified in [146]. The study reported in [342] proposes a randomized stochastic approximation algorithm for the adaptive adjustment of a server parameter used to process the job queue. The study in [144] describes the use of randomized stochastic optimization algorithms in a load balancing problem for a distributed computing network. The SPSA method was proposed in [18] for estimating the centers of thermal updrafts for adaptive autonomous soaring of multiple UAVs. The study reported in [177] proposes leader-follower strategies for a multi-plant differential game based on a randomized stochastic approximation algorithm. In all these examples, traditional assumptions of independence and zero-mean for external noise do not hold due to the specifics of these problems. Not only does randomization speed up the data processing, it also reduces the negative influence of "almost" arbitrary external observation noise. A possible implementation of randomized stochastic approximation algorithms on a quantum computer is examined in [325].

The main ideas, approaches and applications of randomized stochastic approximation are detailed in Chapters 3. Additional descriptions along with a wide variety of applications can be found in [49, 206, 302].

2.3.2 Linear Regression and Filtering

Let $\mathbf{x}_n$, $n = 1, 2, \ldots$, denote a signal process that is not directly observable, and let $\mathbf{u}_n$ be an available (controllable) process. Consider a case in which a scalar product of $\mathbf{u}_n$ and $\mathbf{x}_n$ with an additive noise is observable. Let y_n denote such an observation process such that

$$y_n = \mathbf{u}_n^{\mathrm{T}} \mathbf{x}_n + \text{ observation noise.}$$

The goal is *to find an estimate* $\widehat{\mathbf{x}}(n+1)$ *of the vector* $\mathbf{x}_{n+1}$ *based on the observations up to time* n.

Traditionally, the observed noise is assumed to be a mutual independent and zero-mean. These hypotheses are often hard to justify in practice. Without them, the validity of many algorithms is doubtful in engineering applications. For example, using the standard "least-squares method" or the "maximum likelihood method" is inexpedient under conditions of a possible enemy counteraction. These algorithms have been known to give incorrect estimates if the observation noise has an "unknown-but-bounded" deterministic nature or is a probabilistic "dependent" sequence (the enemy jams the signal). Therefore, the filtering or linear regression (LR) parameter estimation capability should be investigated under minimal assumptions about the statistical characteristics of observation noise.

In the case of non-centered and even non-random correlated noise, the LR parameters can be efficiently estimated, though at first glance this seems surprising. This estimation can be made under certain conditions when the inputs (regressors) $\mathbf{u}_n$ are random and available. Moreover, the optimal algorithms of the LR parameter estimation have the same rates of convergence as in the "standard" case. The concept of using random inputs to eliminate the bias effect was suggested by Fisher [113] as a randomized principle in an experimental design. Apart from the experimental design problem where the regressors can be randomized by an experimenter, random inputs occur in many problems involving identification, filtration, recognition, maneuvered target tracking, telecommunications, and manufacturing.

Recurrent algorithms for estimating the regression parameters with random input signals are discussed in many studies, e.g., Albert [6], Ljung and Soderstrom [215], Tsypkin [324] and Young [355]. Polyak and Tsypkin [259, 265] studied the rate of convergence of such algorithms. They obtained the optimal algorithms with the best possible rates of convergence. All of these studies made standard assumptions about noises. Namely, noises were considered to be a sequence of independent or weakly dependent random variables with zero mean.

In the case of unknown but bounded non-random noises, the minimax problem or the H_∞ approach were usually considered (see, for example, [154, 228]. The advantage of these approaches is that they do not require any specific assumptions about the statistical properties of the noise. The disadvantage, however, is that the accuracy of the estimation depends directly on the noise level. The quality of estimates is not so good when the noise level is high.

The studies reported in [130, 133, 216] considered the LR parameter estimation problems under nonstandard assumptions regarding observation noise. Goldenshluger and Polyak [130] studied this problem for arbitrary noise in the partial case of centered random input signals and time-invariant LR parameters. Yet the algorithms proposed in [130] do not achieve the optimal

rate of convergence in the general case. In the case of practically arbitrary noise, Granichin [133] considered the problem with a time-varying vector of unknown parameters and proposed estimation algorithms for a mean vector of unknown parameters. Ljung and Guo [216] also discussed the possibility of getting strongly consistent parameters estimates when the noises were bounded and deterministic and the input sequence was suitably chosen. The competence of recurrent algorithms was considered in [216] under the case which is similar to the "almost" arbitrary noise case. Similar randomized versions of standard algorithms for filtering under arbitrary bounded noise in observations are considered in [10, 13, 141, 327]. A stochastic gradient type algorithm for closed-loop problems studied in [29].

2.3.3 Compressive Sensing

In the 21^{st} century, the volumes of information to be processed have increased dramatically, mainly due to mass transitions to processing flows of two-dimensional (2-D) and three-dimensional (3-D) data. The complexity of traditional signal quantization methods grows exponentially with dimensionality. Quantization of 1-D signals for $T = 10^3$ corresponds to $T = 10^6$ for the 2-D case and $T = 10^9$ for the 3-D, which is extremely high. In modern applications for digital photos and video cameras, the traditional requirement for the desired measurement frequency (Nyquist rate [249]) is so high that too much data must be compressed substantially before being stored or transmitted. In other applications, including display systems (medical scanners and radars) and high-speed analog-digit converters, increasing a measurement frequency has proven to be too costly.

We assume that *knowledge* makes it possible to restore the required information $\mathbf{x}$ from the data $\mathbf{y}$ acquired in the course of experiments or computations. For the sake of simplicity, one can often assume that the essential information about a phenomenon $\mathbf{x} \in \mathbb{X}$ under consideration is related to the available data $\mathbf{y} \in \mathbb{Y}$ by means of understanding the regularities of the phenomenon — i.e., the knowledge (operator)

$$\mathbf{y} = \mathbf{U}\mathbf{x} \ \ (= \mathbf{U}(\mathbf{x})).$$

If the operator $\mathbf{U}$ is invertible, it provides exhaustive knowledge to fully restore $\mathbf{x}$ from $\mathbf{y}$. It is known from matrix algebra that $\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$ for the case $\mathbf{y}, \mathbf{x} \in \mathbb{R}^T$ and the nonsingular $T \times T$ matrix $\mathbf{U}$.

A case in which the data are subject to the action of an uncontrollable disturbance

$$\mathbf{y} = \mathbf{U}\mathbf{x} + \mathbf{v}$$

is a characteristic of open systems. For an insignificant level of external disturbances $\mathbf{v}$ (or at their damping), the problem of restoring $\mathbf{x}$ from $\mathbf{y}$ comes down to one of inverting the operator $\mathbf{U}$, which usually is accomplished by

increasing the number of observations: choosing $m > T$ for $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{x} \in \mathbb{R}^T$.

From the practical point of view, it is extremely interesting to investigate the possibilities for restoring $\mathbf{x} \in \mathbb{R}^T$ from $\mathbf{y} \in \mathbb{R}^m$ for $m << T$ which is, of course, unattainable in the general case.

Modern information theory originating from the famous Nyquist-Shannon Theorem [249, 290] states that if an analog signal $f : \mathbb{R} \to \mathbb{R}$ from $L_2(\mathbb{R})$ has a limited spectrum, it can be restored uniquely without losses of its discrete readings taken at a frequency greater than the doubled maximal frequency of the spectrum. In many practical applications, the original notion of information $\mathbf{x}$ may be described much more simply than the actual signals f observed by the researcher. For example, to make a decision in some control system, one needs to know that a signal in the form of an acoustic or electromagnetic wave appeared in the registering channel. Of interest is a one-bit answer to the simple *yes/no* question, whereas the arriving and registered signal may take a complex form and be distributed in time and space (multidimensional vector). The rapidly progressing new paradigm of information processing relies on such specificity.

Yet recently a new paradigm of "compressive sensing" has been introduced in place of traditional signal processing theory. This paradigm makes it possible to restore the sparse information $\mathbf{x}$ [65, 97] with sufficient accuracy. The new methodology is based on a certain — usually randomized — selection of the matrix $\mathbf{U}$ and on the fact that the vector $\mathbf{x}$ resulting from ℓ_1-optimization has at most $s < m$ nonzero components, that is, is strongly sparse. This remarkable fact was established and used by one of the authors for constructing an ℓ_1-optimal stabilizing controller of a nonminimum-phase control plant that was initially presented in [132]. An extended and detailed explanation of why only a small number — equal to the codimension of the subspace $\mathbf{Ux} = 0$ — of vector components of the ℓ_1-optimization problem solution is not equal to zero was given in 1984 [24]. More precisely, the CS paradigm uses randomized measurements as nonadaptive linear projections with a measurement operator $\mathbf{H}$ — a random $m \times T$ matrix — retaining the structure of a signal $\mathbf{f} = \mathbf{\Phi x}$. The information $\mathbf{x}$ is recovered using, for example, the methods of ℓ_1-optimization: $\mathbf{x}$ is determined as the solution of a problem such as

$$||\mathbf{x}||_1 \to \min_{\mathbf{x}} \qquad \text{subject to :} \ \mathbf{y} = \mathbf{Ux}$$

where $\mathbf{U} = \mathbf{H\Phi}$ is an operator that converts $\mathbf{x}$ into the set of data $\mathbf{y}$.

During the last decade the term "compressive sensing" was has become used more frequently. The acquisition of information on the basis of compressive sensing may be more efficient than the traditional sampling of rare or compressed signals. Popular estimates using the least square method are inadequate in CS for good signal reconstruction. Therefore, other types of convex optimization are used. The domain of compressive sensing applications has recently gone far beyond the limits of coding/decoding theory and

now embraces problems of image classification and processing. In some applications, the actual arriving signals require no restoration at all, and the data are processed only in their compressed form. More detailed descriptions of this new paradigm are provided in Section 4.2.

2.3.4 Randomized Control Strategies

An insufficient variety of input signals complicates the identification problem. The opportunity for a control system to produce a special control (trial, test, probing) signal in the input channel can significantly alleviate the problem of reconstructing unknown parameters. For example, if a harmonic signal is sent to the input of a steady-state linear plant, after the transition process the plant's output will also be a harmonic signal (assuming there is no noise). The amplitude of this signal is proportional to the value of the plant transfer function at the point which corresponds to the frequency of the input harmonic signal. By varying the frequency, we can reconstruct the entire plant transfer function. Similarly, a series of single pulses in the input channel allows for reconstructing the plant pulse function (see, e.g., [7]).

For a linear control plant with almost arbitrary additive noises in the observations, identification is possible using randomized test signals as part of control actions. Noise does not necessarily possess any useful statistical properties and does not need to be random at all. Unknown parameter values are reconstructed based on the properties of a test signal that is mixed with a control signal. The including of a test signal in a control channel can deteriorate the control performance. However, when an appropriate decision is made about the intensity of a test signal, the output process will be indistinguishable from an optimal process through time. (If the intensity of a test signal is diminished rapidly with time, the identification process does not necessarily have to be complete.)

Consider a dynamical system

$$y_t = G^\star(z^{-1})u_t + v_t \tag{2.5}$$

with input u_t and output y_t shown in Fig. 2.1.

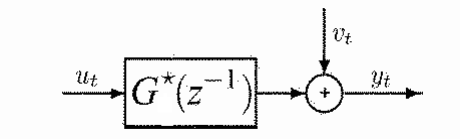

Fig. 2.1. Dynamical system

Noise v_t describes all other sources, apart from u_t, that cause variation in y_t. The transfer function $G^\star(z^{-1})$ belongs to a set of transfer functions

$G(\mathbf{x}, z^{-1})$, parameterized by $\mathbf{x}$, i.e., $G^\star(z^{-1}) = G(\mathbf{x}^\star, z^{-1})$ for some $\mathbf{x}^\star$. The structure of the model class $G(\mathbf{x}, z^{-1})$ is known, but $\mathbf{x}^\star$ itself is unknown. The problem under consideration is to determine a confidence region $\widehat{X}(T)$ for $\mathbf{x}^\star$ with a specified probability chosen by a user based on the input and output data collected at time $t = 1, 2, \ldots, T$. Moreover, $\widehat{X}(T)$ must be constructed without any a priori knowledge of the level, distribution, or correlation of the noise.

The standard approach to obtaining confidence regions is to use *an asymptotic theory* of system identification (see, e.g., [148, 217]). Although these results have been used successfully in many applications, asymptotic estimates are only reliable when the data volume T approaches infinity. When the number of points of data measurements is finite, the asymptotic theory may cause erroneous results even for large data sets. Another method, known as *Set Membership Identification*, assumes that the boundaries of all uncertain system components are known a priori. As a result, the guaranteed region of parameters is defined as a set of values that do not violate a priori boundaries [21, 44, 45, 156, 165, 191, 261, 266, 299].

Identification investigation techniques with test signals were used in [282] and subsequently extended in [283] to closed control systems. In these studies the a priori stability of the plant was assumed, the noise was assumed to be a white noise process, and in addition, a relatively limiting constraint was placed on the noisy control. Given the possibility of direct influence on the controlled processes through the system's input variables, in 1986 Granichin, Fomin [145] and Chen, Guo [77] proposed and justified that adaptive control synthesis would, through a control channel, include the randomized trial perturbation in addition to the base ("own") control inputs. The main result of [145] was a randomized strategy for the minimax adaptive optimal control problem that make it possible to achieve an asymptotically optimal level of outputs, as in [24] for the known plant parameters. The algorithm, where a special re-parametrization of a control plant model was used, allows for asymptotical identification of the unknown control plant parameters under almost arbitrary additive noise in a plant model. The procedure is valid for any noise v_t and does not require a priori knowledge of its characteristics. The noise may be non-random, white or correlated random with zero-mean or bias, and the signal-noise ratio may be high or low. This result was extended to the case of time-varying parameters in [13, 327]. The information about the maximum possible amplitude of the noise has only been used in formulas for estimating the rate of convergence. That is to say, this knowledge is not required for operability of an identification algorithm. The recovery of unknown parameter values is provided by the properties of randomized test signals, which are added together with an intrinsic adaptive control signal from a closed loop. This approach follows from Feldbaum's concept of *dual control* [109]. In 2010 Campi and Weyer [63] proposed a procedure that provides rigorously guaranteed non-asymptotic confidence regions for the unknown parameters of a linear dynamical control plant that is disturbed by

an arbitrary noise. It used a control strategy with randomized test signals at each iteration. Their procedure consists of simple input design steps followed by an algorithm called LSCR (Leave-out Sign-dominant Correlation Regions). A similar procedure was proposed in [14, 143] with the same reparametrization of a control plant model as in [145]. But it only includes randomized items in the input channel once per some interval, and it gives smaller dimensions of confidence regions (compare with [63]). In [11], both asymptotic and non-asymptotic approaches were consider together.

More detail descriptions are provided in Chapter 5.

2.4 Data Mining and Machine Learning

The problem of classification of input signals (stimuli, objects) is a typical problem in learning theory. The classification process yields signs that characterize a group of objects as a data set or class (clusters). On these grounds, each signal can be attributed to a particular class. Clustering results in the partition of signals into groups in the environment when classes are not defined in advance. The resulting partition naturally describes the structure of a data set and can be used in the future to define it.

When a signal (stimulus) $\mathbf{w}$ is fed as input, a classification system generates values of functions $\{h_i(\mathbf{x}, \mathbf{w})\}$ depending on parameters $\mathbf{x}$ and computes their sum $s(\mathbf{x}, \mathbf{w})$ with certain weights [116, 118, 148], thereby defining the sets (images)

$$\mathbb{X}_1(\mathbf{x}) = \{\mathbf{w} : s(\mathbf{x}, \mathbf{w}) > 0\},\ \mathbb{X}_2(\mathbf{x}) = \{\mathbf{w} : s(\mathbf{x}, \mathbf{w}) < 0\}.$$

A classification system can classify any input signal $\mathbf{w}$, assigning it either to the set $\mathbb{X}_1(\mathbf{x})$ or to the set $\mathbb{X}_2(\mathbf{x})$. This classification may vary if the parameters $\mathbf{x}$ are varied. By means of parameter variation, the system may fit its classification to some standard classification and thus demonstrate its learning or adaptation capabilities. Such a fitting requires certain additional classification information, which is usually obtained from a learning sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$ of specifically classified input signals. By refining this information, we obtain different formulations of the learning problem. The selection of parameters through a learning sequence is called *learning*. When learning is completed, weight coefficients are recorded and the corresponding sets $\mathbb{X}_1(\mathbf{x})$ and $\mathbb{X}_2(\mathbf{x})$ are taken to be the required partition, which may not coincide with the real partition. This difference, expressed in some manner, determines the quality of the learning system's operation. In reality, for any disjoint bounded sets separated by a positive distance, there exists a finite set of threshold functions that map these sets into linear separable sets. This fundamental fact underlies modeling using neural networks [162, 267, 331, 350].

This learning scheme can be implemented with perceptrons-complex networks consisting of threshold elements (formal neurons) intended for modeling processes in complex systems, particularly in living organisms. The functions $\{h_i(\mathbf{x}, \mathbf{w})\}$ act as the response of output neurons to an input stimulus $\mathbf{w}$ (in an analogy to visualization systems, the input stimulus is sometimes called the image). The network's output neuron signals are fed to an effector neuron, where they are added and compared with a threshold (zero in our example). The class to which the input stimulus must be assigned from among two classes is evaluated from the result of this comparison. This problem can obviously be extended to a number of classes greater than two. If the estimation procedure in learning is based on a teacher's instructions for classification of the learning sequence, it is referred to as *learning with a teacher*. Learning without the aid of a teacher's instructions is called *self-learning*, and learning in this case is reduced to determining an estimate sequence that minimizes a special functional [118].

2.4.1 Clustering

Procedures for classifying objects consistent with their seeming resemblance are quite prevalent in science. Cluster analysis encompasses algorithms and methods intended for grouping or classifying objects. Each item is presented as a set of measurements or as relationships between the item and other items. The objective of cluster analysis, or unsupervised classification, is to divide data (objects, instances, items) into groups (clusters) so that items belonging to the same group are more similar than items belonging to distinct groups. The crucial ingredient of most clustering algorithms is the similarity measure constructed to reflect the inner data structure. Cluster analysis deals with difficult problems. Hence, suitable clustering techniques are needed for a given clustering task. It is well known that no clustering method can work effectively with all kinds of cluster structures and data. Generally speaking, most existing clustering methods can be categorized into three groups: partitioning, hierarchical, and density-based approaches. Farley and Raftery [122] propose dividing the clusterings into two main groups: hierarchical and partitioning methods. Han and Kamber [158] propose grouping the approaches into three main categories: density-based methods, model-based clustering methods and grid-based methods. A different categorization based on the induction principle is suggested by [104].

A partition (clustering) $\mathcal{X}^k(\mathbb{W}) = \{\mathbb{X}_1, \ldots, \mathbb{X}_k\}$ of the set $\mathbb{W}$ is a set consisting of k non-empty clusters.

For any $k \in 1..k_{\max},\ k_{\max} \leq |\mathbb{W}|$ a set $\mathbb{W}$ has many different partitions $\mathcal{X}^k(\mathbb{W})$. The best partition is defined usually by the rule: items belonging to the same group are more similar than items belonging to distinct groups. When a probability distribution $\mathrm{P}(\cdot)$ is defined on a set $\mathbb{W}$, a standard approach is to associate some *distortion (penalty, quality) functions* q_i, defined

as "closeness" to the cluster i, with each cluster (group) $i \in 1..k$, and to consider the problem of minimization

$$F(\mathcal{X}^k) = Ef(\mathcal{X}^k, \mathbf{w}) \to \min_{\mathcal{X}^k},$$
$$f(\mathcal{X}^k, \mathbf{w}) = \max_{i \in 1..k} q_i(\mathcal{X}^k, \mathbf{w}).$$

One important question is to determine a more appropriate way to parameterize the partition $\mathcal{X}^k$. In St. Petersburg State University the theory of machine learning began developing in the 1960s under the leadership of Vladimir Yakubovich of the Theoretical Cybernetics Department. This theory is based on using approximation methods when input signals are projected to some finite set of basis functions and on parameterizing a partition by correspondence expansion coefficients [116].

Starting with the work of Vapnik and Chervonenkis [328], a *statistical theory of recovery dependency on empirical data* has been consistently developed in the scientific literature under the assumption that a partition can be characterized by parameters of a probability distribution. This theory is directed at finding the conditions of uniform convergence of empirical averages to expectations. Instead of evaluating $Ef(\mathcal{X}^k, \mathbf{w})$ as in the MC method, the goal is much more technically complex: to estimate $E\{f(\mathcal{X}^k, \mathbf{w})|\mathbf{y}\}$ where $\mathbf{y}$ is a random variable with respect to which the expectation is made. This problem of uniform convergence is closely related to learning goals: if the uniform convergence holds, the solution of the conditional minimization of the empirical error with respect to observations $\mathbf{y}$ returns an almost minimum expectation. That is, it provides a good estimate for the solution of a machine learning problem (see, e.g., [329, 331]).

2.4.2 Cluster Validation

Cluster analysis distinguishes between two types of methods. The first type, called clustering methods, deals with finding a partition of the data elements into a given number of clusters. No additional external information about the cluster process is available. Implementation of a clustering algorithm requires the suggested number of clusters that make up the considered dataset as one of the input parameters. The second type, called validation methods, deals with determining the optimal ("true") number of clusters in a dataset. Here, the clustering solutions are compared according to a predefined rule and the optimal number of clusters is chosen as the one that yields the optimal quality. This essential cluster analysis task may have more than one solution and thus is recognized as being "ill posed" [131, 175]. For instance, the result can be dependent on the units by which the data is measured (see, for example, [70]). The result can be also depend on the choice of clustering algorithm, as illustrated by the following example.

Let us consider partitions into two clusters provided for a dataset from a real line that has been simulated as a mix of three Gaussian components.

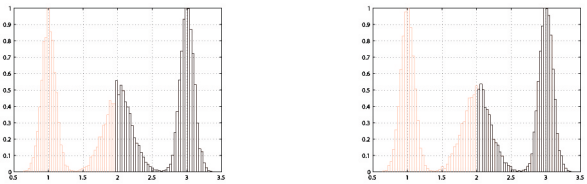

Fig. 2.2. Outcomes of repeated clusterings by means of the standard k-means algorithm

Figure 2.2 shows the outcomes of two repeated partitions into two clusters of the randomly initialized standard k-means algorithm. Figure 2.3 demonstrates the result obtained by a version of the known EM-algorithm.

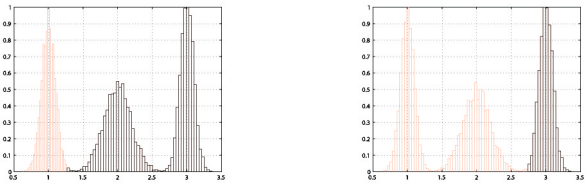

Fig. 2.3. Outcomes of repeated clusterings by means of the EM-algorithm

As can be deduced, the k-means algorithm recovers a "false" stable two-cluster construction. However, a more flexible EM approach hints at possible unsteadiness of the two-cluster structure. Here, we used the Classification EM-algorithm (CEM) introduced by Celeux and Govaert in [67] as a version of the classical EM method. Note that the standard k-means algorithm can be viewed as its partial case corresponding to equally-sized spherical clusters. Although many approaches have been proposed to solve these problems, to date none has been accepted as superior.

Solution methods for this problem can be roughly divided into the following groups:

- Methods established on geometrical properties of the clusters, such as within— and between—cluster dispersion.
- Methods based on the stability concept.
- Methods based on the goodness of fit tests.
- Methods based on information criteria.

The methods that can be attributed to the first group were described by Dunn [100], Hubert and Schultz [169] (C-index), Calinski-Harabasz [60], Hartigan [159], Krzanowski-Lai [201], Sugar James [309], Gordon [131], Milligan and Cooper [236], and Tibshirani, Walter and Hastie [314] (the *Gap Statistic*

method). The slope of the within-cluster dispersion matrix usually decreases as the number of groups rises, and may have a point at which it "drops" to a value at which the decrease rate is slower (i.e., its gradient obtains a minimal value over this point). In this approach the location of this "elbow" on the graph is used to indicate the actual ("true") number of clusters.

Methods belonging to the second group usually compare pairs of clustered samples from the same data source. Reiterating a clustering algorithm for those pairs allows measurement of the variability of clustering solutions. High consistency among the cluster properties is identified with partition reliability [79]. The number of clusters that maximizes cluster stability is frequently used to evaluate the actual number of clusters.

Levine and Domany [213], Ben-Hur, Elisseeff and Guyon [34], Ben-Hur and Guyon [35] measure stability according to the proportion of times that a pair of elements belonging to the same cluster in a run of the clustering algorithm maintains the same membership when the algorithm is rerun. Bel Mufti, Bertrand, and El Moubarki [241] exploit Loevinger's measure of isolation to determine a stability function. Another perspective employs external partition correlation indexes as a stability measure. For example, this methodology has been implemented in the known *Clest method* of Dudoit and Fridlyand [99]. Lange, Braun, Roth and Buhmann Roth et al. [278], and Lange et al. [208, 209] offer a theoretical justification of the cluster validation problem by means of the stability concept. In the suggested model, pairs of clustered samples are compared, and stability is defined as the proportion of times an element is assigned to the same cluster. Tibshirani and Walther [315] employ a similar prediction strength procedure. Jain and Moreau [174] considered the dispersions of empirical distributions as a stability measure.

Another methodology connected to the geometrical standpoint is provided by nonparametric density estimation. This method matches the clusters to the probability distribution function density peaks. Thus, the clustering assigns each item to one "domain of attraction" of the density modes. Wishart [353] apparently first proposed to look for modes in order to discover the cluster structure.

This idea was adopted by Hartigan [159, 160] to establish the concept of high density modes clusters. The cluster amount is understood as the number of areas in which the density is greater than a given threshold. From this standpoint, clusters are imaged as isolated islands of "high" density in an ocean of "low" densities. This concept was later implemented in numerous studies (see, for example, [87, 88, 308]).

Goodness of fit test procedures offer another approach, in particular the X-means algorithm proposed by Pelleg and Moore [253]. This algorithm is used to calculate the best score of the so-called *Bayesian Information Criterion (BIC)* [184] used to detect the true number of clusters. Hamerly and Elkan [157] apply another scoring criteria in the G-means algorithm. Here, a statistical projection procedure is used to test the hypothesis that the data in a cluster are drawn from a Gaussian distribution. This algorithm checks

an initial small number of clusters and divides those that were declined as Gaussian into two clusters.

In the spirit of the G-means algorithm, Feng and Hamerly [110] propose an algorithm called PG-means (PG stands for projected Gaussian) that also applies projections to the clustering model. PG-means employs the standard Gaussian mixture model using the EM-algorithm described in Subsection 6.4. Here, any Gaussian mixture-based technique can be applied. An *Information Theoretic perspective* on the cluster validation problem was discussed by Still and Bialek in [306].

As usual, decisions are based on heuristics such as the *Akaike's Information Criteria (AIC)* [5, 51, 52] or BIC [289]. AIC is used to assess the quality of the validation of statistical models and deals with the concept of information entropy, suggesting a relative measure of the information lost when a given model is applied to describe reality. In other words, AIC makes it possible to describe the exchange between bias and variance in model construction. BIC is a criterion for model selection among a finite set of models based on increasing the likelihood between models by adding parameters. To avoid over fitting, BIC presents a penalty term for the number of parameters in the model. The relationship between performance of information criteria and type of measurement of clustering variables was studied in [115]. A normalized entropy criterion was considered in [68].

Later, Volkovich et al. [336, 337] and Barzily et al. [30] suggest new methods using the goodness of fit tests. Etalon cluster distributions are constructed, relying on a model designed to represent well-mixed samples within the clusters. This idea apparently was implemented in *Binomial Model* [319, 335] and *Minimal Spanning Tree (MST)* model [30, 339]. According to [333], the actual number of clusters is the one that minimizes classification risk as a function of the number of clusters. Here, the risk is quantified by the agreement of clustering solutions with regard to the class labels formed by a clustering algorithm applied on two samples, with a sequential propagation from one solution to another.

A general cluster stability methodology that makes it possible to reformulate many existing approaches from a new generic point of view offering new cluster stability criteria was introduced by Volkovich et al. [334]. Clustered sample elements are considered as instances of random variables such that the partition quality is measured by probability metrics between these instances. This perception suggests statistical homogeneity of the clustered samples in the case of the actual number of clusters. In other words, it is presumed that sample occurrences in a cluster appear to be, in this situation, independent realizations of the same random variable. This point of view conceptually generalizes several previous methods referred to above. The approaches presented in [208, 209, 278, 315] employ Breckenridge's notion [55]. Here, the resemblance among the cluster solutions is evaluated based upon a classifier trained by means of a second (clustered) dataset. As usual, this similarity is characterized by the closeness of the measured cluster contents, in fact by the

indicator probability metric. A formal disadvantage of this scheme is that in order to predict cluster membership, an additional prediction procedure has to be employed. The studies reported in [30, 335, 336, 337] use variants of the goodness of fit tests based on the so-called well mingled samples concept. In the case of the actual number of cluster samples, occurrences in clusters are suggested to be mingled as they are mingled in the whole population. Obviously, content-based approaches can also be treated from this standpoint, and closeness of sample occurrences can be evaluated by simple probability metrics (two-sample test statistics). The content-based and the geometrical approaches actually implement the same general cluster model. Thus, they can be understood and studied from a uniform point of view.

Based on general ideas of *scenario approachers* [57] new randomized algorithms are proposed in [20, 149, 240] for finding confidence intervals of the actual number of clusters.

Part II

Randomization in Estimation, Identification and Filtering Problems under Arbitrary External Noises

Mathematical algorithms for search and optimization play a major role in finding the best options to solve many problems in physics, engineering, business and medicine, as well as in other natural and social sciences. In the case of prior information ambiguity, recursive algorithms are the most effective among many other approaches. Such algorithms start with an initial "guess" of a solution, and this assessment is updated on an iteration basis with the purpose of improving the measured (observable) objective function of the sample. In most practical problems, the relevant solution depends on more than one quantity, leading to the multivariate optimization problem of minimizing or maximizing the objective function, dependent on multiple variables. There is much interest in recursive optimization algorithms that rely only on direct measurements of the objective function, not on direct probing of its gradient. Such algorithms have the advantage of not requiring detailed modeling information describing links between parameters and the objective function to be optimized. Many systems involving human beings or computer simulations that are difficult to treat analytically might potentially benefit from applications of this kind of optimization approaches.

In the design of optimization or estimation algorithms, some useful statistical characteristics are usually attributed to noises, errors in measurements and model properties that are used in demonstrating the validity of the algorithm. For example, noise is often assumed to be random and centered. Algorithms based on the ordinary least-squares method are used in engineering practice for simple averaging of observation data. If noise is assumed to be centered without any valid justification, such algorithms are unsatisfactory in practice and may even be harmful. Such is the state of affairs under "opponent" counteraction. In particular, if noise is defined by a deterministic unknown function (opponent suppresses signals) or measurement noise is a dependent sequence, averaging of observations does not yield any useful result. Analysis of typical results in such situations shows that the estimates generated by ordinary algorithms are unsatisfactory, the observation sequences are believed to be degenerated and the problems are usually disregarded.

Another related problem in applying many recurrent estimation algorithms is that observation sequences do not have adequate variability. For example, the main aim in designing adaptive controls is to minimize the deviation of the state vector of a system from a given trajectory, and this often results in a degenerate observation sequence. Consequently, identification is complicated, and successful identification requires diverse observations.

Our new approach to estimation and optimization under poor conditions (for example, degenerate observations) is based on the use of test disturbances. The information in the observation channel can be enriched for solving many problems by introducing *a test disturbance (simultaneous trial perturbations)* with known statistical properties into the input channel of a control system or into the algorithm.

In control systems, test actions can be introduced via the control channel, whereas in other cases, a randomized observation plan (experiment) can be used as the test action. Sometimes one of the measured random processes that are in the system can be used as a test disturbance. In studying a renewed system with a test disturbance, which is sometimes simply the old system in a different form, results on the convergence and applicability fields of new algorithms can be obtained through traditional estimation methods. One remarkable characteristic of such algorithms is their convergence under almost arbitrary noise. An important constraint on their applicability is that the test disturbance and external noise are assumed to be independent. This constraint is satisfied in many problems. Such is the case if noise is taken to be *an unknown but bounded deterministic function* or some external random perturbation generated by some statistical properties of the test disturbance unknown to the opponent counteracting our investigation.

3

Randomized Stochastic Approximation

Multidimensional stochastic optimization plays an important role in the analysis and control of many technical systems. Randomized algorithms of stochastic approximation with perturbed input have been suggested for solving the challenging multidimensional problems of optimization. These algorithms have simple forms and provide consistent estimates of the unknown parameters for observations under almost arbitrary noise. They are easily incorporated into the design of quantum devices for estimating the gradient vector of a multivariable function.

3.1 Mean-Risk Optimization

Many practical applications need to optimize one or another mean risk functional. Although the extremal values can sometimes be established analytically, engineering systems often deal with an unknown functional whose value or gradient can be calculated at the given points.

Let $f(\mathbf{x}, \mathbf{w}) : \mathbb{R}^d \times \mathbb{W} \to \mathbb{R}$, $\mathbb{W} \subset \mathbb{R}^r$, be a $\mathbf{x}$-differentiable function and let $\mathbf{u}_1, \mathbf{u}_2, \ldots$ be a sequence of measurement points chosen by the experimenter (observation plan), at which the value $y_1, y_2, \ldots$ of the function $f(\cdot, \cdot)$ is accessible to observation at every time instant $t = 1, 2, \ldots$, with additive external noise v_t

$$y_t = f(\mathbf{u}_t, \mathbf{w}_t) + v_t, \tag{3.1}$$

where $\mathbf{w}_t \in \mathbb{W}$, $t = 1, 2, \ldots$, are non-controllable random variables (vectors).

Problem formulation. Using the observations $y_1, y_2, \ldots$ and inputs $\mathbf{u}_1, \mathbf{u}_2, \ldots$, construct a sequence of estimates $\{\widehat{\mathbf{x}}(n)\}$ of an unknown vector $\mathbf{x}^\star$ minimizing the *mean-risk functional*

$$F(\mathbf{x}) = E_{\mathbf{x}} f(\mathbf{x}, \mathbf{w}) \to \min_{\mathbf{x}}. \tag{3.2}$$

A maximum likelihood estimation is widely used for some likelihood functions $L(\mathbf{x}, \mathbf{w})$. In this context, a minus log likelihood function $-\ln L(\mathbf{x}, \mathbf{w})$

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_3,

corresponds often the special case of a mean-risk functional minimization problem with

$$f(\mathbf{x}, \mathbf{w}) = -\ln L(\mathbf{x}, \mathbf{w}). \tag{3.3}$$

For example, in the case of *Gaussian likelihood function* $L(\mathbf{x}, \mathbf{w}) = G(\mathbf{w}|\mathbf{x}, \mathbf{\Gamma})$ with the mean $\mathbf{x}$ and covariance matrix $\mathbf{\Gamma}$, we have

$$f(\mathbf{x}, \mathbf{w}) = -\ln G(\mathbf{w}|\mathbf{x}, \mathbf{\Gamma}) = (\mathbf{w} - \mathbf{x})^{\mathrm{T}} \mathbf{\Gamma}^{-1} (\mathbf{w} - \mathbf{x}). \tag{3.4}$$

Minimization of the function $F(\mathbf{x})$ is usually studied with a simpler observation model

$$y_t = F(\mathbf{u}_t) + v_t.$$

The generalization used in the formulation (3.1) has several motivations.

First, it takes into account the case of multiplicative perturbations in observations:

$$y_t = \mathbf{w}_t \bar{f}(\mathbf{u}_t) + v_t.$$

Second, it allows for separation of the observation noise with "good" (e.g., zero-mean independent and identically distributed) statistical properties $\{w_t\}$ and arbitrary additive external noise $\{v_t\}$. Of course, this separation is not needed when we can assume that $\{v_t\}$ is a random zero-mean and independent and identically distributed as well.

Third, we also encounter problems where the optimized functional may vary in time and its extremum point may drift. In such cases, the problems may be posed differently depending on the optimization goals and measurable data. Usually two options are considered for the behavior of the drift $\{\mathbf{x}_t\}$ of the functional's point of minimum. These options differ in the answer to the following question: Is there some random independent and identically distributed sequence with an expectation $\mathbf{x}^\star$? The case of a positive answer is included in problem statement (3.2) since we can consider additional disturbances $\mathbf{w}_t = \mathbf{x}_t - \mathbf{x}^\star$. A more general non-stationary statement of the problem is

$$F_t(\mathbf{x}) = E_{\xi_t, \mathbf{x}} f_{\xi_t}(\mathbf{x}, \mathbf{w}_t) \to \min_{\mathbf{x}}, \tag{3.5}$$

where $\{f_\xi(\mathbf{x}, \mathbf{w})\}_{\xi \in \Xi}$, is a family of $\mathbf{x}$-differentiable function: $f_\xi(\mathbf{x}, \mathbf{w}) : \mathbb{R}^d \times \mathbb{W} \to \mathbb{R}$, $\mathbb{W} \subset \mathbb{R}^r$, and, for a chosen sequence $\mathbf{u}_1, \mathbf{u}_2, \ldots$ we can, with additive external noise v_t, observe

$$y_t = f_{\xi_t}(\mathbf{u}_t, \mathbf{w}_t) + v_t, \tag{3.6}$$

where $t = 1, 2, \ldots$, $\{\xi_t\}$ is a non-controllable sequence: $\xi_t \in \Xi$, $\mathbf{w}_t \in \mathbb{W}$ are non-controllable random variables (vectors).

More precisely, the time-varying point of minimum $\mathbf{x}_t$ of the function $F_t(\mathbf{x})$ *needs to be estimated*.

The problem (3.2) is a partial case of (3.5) when $\mathbf{x}_t \equiv \mathbf{x}^\star$ and $\Xi = \{\xi_1\}$.

Let us formulate the main conditions that are usually assumed.

As3.1: The function $F(\cdot)$ is strongly convex in the first argument, i.e. it has a unique minimum point $\mathbf{x}^\star$ and

$$\langle \mathbf{x} - \mathbf{x}^\star, \nabla F(\mathbf{x}) \rangle \geq \mu \|\mathbf{x} - \mathbf{x}^\star\|^2, \ \ \forall \mathbf{x} \in \mathbb{R}^d$$

with a constant $\mu > 0$.

As3.2: $\forall \mathbf{w} \in \mathbb{W}$ the gradient $\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{w})$ satisfies the Lipschitz condition: $\forall \mathbf{x}', \mathbf{x}'' \in \mathbb{R}^d$

$$\|\nabla_{\mathbf{x}} f(\mathbf{x}', \mathbf{w}) - \nabla_{\mathbf{x}} f(\mathbf{x}'', \mathbf{w})\| \leq M \|\mathbf{x}' - \mathbf{x}''\|$$

with a constant $M \geq \mu$.

As3.3: Local Lebesgue property (condition for the possibility swap operations of integration and differentiation): $\forall \mathbf{x} \in \mathbb{R}^d \ \exists$ the neighborhood $U_{\mathbf{x}}$ and the function $G_{\mathbf{x}}(\mathbf{w}) : E_{\mathbf{x}} G_{\mathbf{x}}(\mathbf{w}) < \infty$ and $||\nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{w})|| \leq G_{\mathbf{x}}(\mathbf{w})$ $\forall \mathbf{x} \in U_{\mathbf{x}}$.

Note, that conditions *As3.1–3.3* hold for a minus Gaussian likelihood function (3.4) when covariance matrix $\mathbf{\Gamma} > 0$:

$$\mu = \frac{1}{\lambda_{\max}(\mathbf{\Gamma})}, \ M = \frac{1}{\lambda_{\min}(\mathbf{\Gamma})}.$$

In the special case $\mathbf{\Gamma} = \sigma^2 \mathbf{I}$, we have

$$f(\mathbf{x}, \mathbf{w}) = \frac{1}{2\sigma^2} ||\mathbf{w} - \mathbf{x}||^2, \tag{3.7}$$

$\mu = M = \sigma^{-2}$.

3.2 Exciting Testing Perturbation as Randomization: Estimation Algorithms

Let $\mathbf{\Delta}_n, \ n = 1, 2, \ldots$ be an observed sequence of independent random vectors in $\mathbb{R}^d$, called the *simultaneous test perturbation*, with a distribution function $\mathrm{P}_n(\cdot)$, and let $\mathbf{K}_n(\cdot) : \mathbb{R}^d \to \mathbb{R}^d, \ n = 1, 2, \ldots,$ be some vector functions (kernels) with compact supports which, along with distribution functions of the test perturbation, satisfy the conditions

$$\begin{gathered} \int \mathbf{K}_n(\mathbf{x}) \mathrm{P}_n(d\mathbf{x}) = 0, \ \ \int \mathbf{K}_n(\mathbf{x}) \mathbf{x}^{\mathrm{T}} \mathrm{P}_n(d\mathbf{x}) = \mathbf{I}, \\ \sup_n \int \|\mathbf{K}_n(\mathbf{x})\|^2 \mathrm{P}_n(d\mathbf{x}) < \infty, \ \ n = 1, 2, \ldots. \end{gathered} \tag{3.8}$$

For example, we can chose a realization of a sequence of independent Bernoulli random vectors from $\mathbb{R}^d$ with each component independently assuming values $\pm\frac{1}{\sqrt{d}}$ with the probabilities $\frac{1}{2}$ as a sequence $\{\mathbf{\Delta}_n\}$ and $\mathbf{K}_n(\mathbf{x}) \equiv \mathbf{x}$ as kernel functions.

Let us take a fixed initial vector $\widehat{\mathbf{x}}(0) \in \mathbb{R}^d$ and choose sequences of positive numbers $\{\alpha_n\}$, $\{\beta_n^+\}$ and $\{\beta_n^-\}$. We design two algorithms for constructing sequences of points of observations $\{\mathbf{u}_t\}$ and estimates $\{\widehat{\mathbf{x}}(n)\}$. The first algorithm uses one observation at every step (iteration)

$$\begin{cases} \mathbf{u}_n = \widehat{\mathbf{x}}(n-1) + \beta_n^+ \boldsymbol{\Delta}_n,\ y_n = f(\mathbf{u}_n, \mathbf{w}_n) + v_n, \\ \\ \widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \frac{\alpha_n}{\beta_n^+} \mathbf{K}_n(\boldsymbol{\Delta}_n) y_n, \end{cases} \tag{3.9}$$

and the second one uses two observations

$$\begin{cases} \mathbf{u}_{2n} = \widehat{\mathbf{x}}(n-1) + \beta_n^+ \boldsymbol{\Delta}_n,\ \mathbf{u}_{2n-1} = \widehat{\mathbf{x}}(n-1) - \beta_n^- \boldsymbol{\Delta}_n, \\ \\ \widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \frac{\alpha_n}{\beta_n^+ + \beta_n^-} \mathbf{K}_n(\boldsymbol{\Delta}_n)(y_{2n} - y_{2n-1}). \end{cases} \tag{3.10}$$

Algorithms (3.9) and (3.10) correspond to statistics (2.3) and (2.4) described above in Subsection 2.3.1.

3.3 Convergence of Estimates

The convergence of the estimates generated by algorithms (3.9) and (3.10) is studied in details in [138] for the cases of $\beta_n^- = \beta_n^+$ or $\beta_n^- = 0$. Here we present a similar result for the general case of algorithm (3.10).

Let $\mathcal{F}_{n-1}$ be the σ-algebra of probabilistic events generated by the random variables $\widehat{\mathbf{x}}(0), \widehat{\mathbf{x}}(1), \ldots, \widehat{\mathbf{x}}(n-1)$ formed by the algorithm (3.10), and denote $\bar{v}_n = v_{2n} - v_{2n-1}$, $\bar{\mathbf{w}}_n = \begin{pmatrix} \mathbf{w}_{2n-1} \\ \mathbf{w}_{2n} \end{pmatrix}$, $\beta_n = (\beta_n^+ + \beta_n^-)/2$.

We need to generalize Assumption *As1.1*. Assume that the following condition holds

As3.4: Random vectors $\bar{\mathbf{w}}_n$ and $\boldsymbol{\Delta}_n$ are independent, random vectors $\bar{\mathbf{w}}_2, \bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_{n-1}$ do not depend on $\bar{\mathbf{w}}_n$ and $\boldsymbol{\Delta}_n$. If values $\bar{v}_1, \ldots, \bar{v}_n$ are random, they also do not depend on $\bar{\mathbf{w}}_n$ and $\boldsymbol{\Delta}_n$.

Theorem 3.1. *Let conditions As3.1–3.4 and* (3.8) *be satisfied.*
If $\sum_n \alpha_n = \infty$ *and* $\alpha_n \to 0$, $\beta_n \to 0$, $\alpha_n^2 \beta_n^{-2}(1 + E\bar{v}_n^2) \to 0$ *as* $n \to \infty$,
then $E\|\widehat{\mathbf{x}}(n) - \mathbf{x}^\star\|^2 \to 0$ *as* $n \to \infty$.
Moreover, **if** $\sum_n \alpha_n \beta_n^2 < \infty$ *and*

$$\sum_n \alpha_n^2 \beta_n^{-2}(1 + E_{\mathcal{F}_{n-1}} \bar{v}_n^2) < \infty, \quad \textit{with probability } 1,$$

then $\widehat{\mathbf{x}}(n) \to \mathbf{x}^\star$ *as* $n \to \infty$ *with probability* 1.

Proof. The proof of Theorem 3.1 is similar to the corresponding proof in [138]. The sketch of the proof is following. By virtue of the algorithm (3.10) observation model (3.1) and conditions of Theorem 3.1 we can get the bounds

$$E_{\mathcal{F}_{n-1}}||\widehat{\mathbf{x}}(n)-\mathbf{x}^\star||^2 \le (1-c_1\alpha_n)||\widehat{\mathbf{x}}(n-1)-\mathbf{x}^\star||^2 + c_2\alpha_n^2\beta_n^{-2}(1+E_{\mathcal{F}_{n-1}}\bar{v}_n^2)$$

with constants c_1 and c_2 which are determined by the conditions of Theorem 3.1, and all conditions of the Robbins-Siegmund Lemma [274] that are necessary for the convergence of $\{\widehat{\mathbf{x}}_n\}$ to the point $\mathbf{x}^\star$ as $n \to \infty$ with probability 1 are satisfied. To prove the result of Theorem 3.1 for the mean-square convergence, let us examine the unconditional mathematical expectation of both sides of the last inequality

$$E||\widehat{\mathbf{x}}(n)-\mathbf{x}^\star||^2 \le E||\widehat{\mathbf{x}}(n-1)-\mathbf{x}^\star||^2(1-c_1\alpha_n) + c_2\alpha_n^2\beta_n^{-2}(1+E\bar{v}_n^2).$$

The mean-square convergence of the sequence $\{\widehat{\mathbf{x}}(n)\}$ to the point $\mathbf{x}^\star$ is implied by Lemma 5 of [258]. This completes the sketch of the proof of Theorem 3.1.

Remarks 1. Instead mean-square boundedness of noise and disturbances, we can use weaker assumptions about their statistical moments of an order ρ : $1 < \rho < 2$ (see [325]).

2. The conditions of Theorem 3.1 hold for the function $f(\mathbf{x}, w) = w\bar{f}(\mathbf{x})$ if the function $\bar{f}(\mathbf{x})$ satisfies conditions *A3.1–3.2*.

3. The observation noise v_n in Theorem 3.1 can be said to be almost arbitrary since they may either be nonrandom but bounded or they may also be a realization of some stochastic process with arbitrary internal dependencies. In particular, to prove the results of Theorem 3.1, there is no need to assume that $\bar{v}_n$ and $\mathcal{F}_{n-1}$ are not dependent.

4. The assertions of Theorem 3.1 do not hold if the noise $\bar{v}_n$ depends on the observation point $\bar{v}_n = \bar{v}_n(\mathbf{u}_n)$ in a specific manner. If such a dependence holds, for example, due to rounding errors, the observation noise must be subdivided into two parts $\bar{v}_n(\mathbf{u}_n) = \tilde{v}_n + \zeta(\mathbf{u}_n)$ in solving such a problem. The first part may satisfy the conditions of Theorem 3.1. If the second part results from rounding errors, then as a rule it has good statistical properties and may not prevent the convergence of algorithms.

5. The condition that the observation be independent of the test perturbation can be relaxed. It is sufficient to assume that the conditional mutual correlation between $\bar{v}_n$ and $\mathbf{K}_n(\mathbf{\Delta}_n)$ approaches zero with probability 1 at a rate not less than $\alpha_n\beta_n^{-1}$ as $n \to \infty$:

$$\|\mathrm{E}\{\bar{v}_n\mathbf{K}_n(\mathbf{\Delta}_n)|\mathcal{F}_{n-1}\}\| = \mathcal{O}\left(\frac{\alpha_n}{\beta_n}\right).$$

6. Though algorithms (3.9) and (3.10) may look alike, algorithm (3.9) is more suitable for use in real time systems if observations contain arbitrary noise. For algorithm (3.10), the condition that observation noise v_{2n} and the

test perturbation $\mathbf{\Delta}_n$ be independent is rather restrictive, because the vector $\mathbf{\Delta}_n$ is used at the previous time instant $2n-1$ in the system. For the algorithm (3.10) the noise v_{2n} and the test perturbation vector $\mathbf{\Delta}_n$ simultaneously appear in the system and can be regarded as independent only in the case when $\beta_n^- = 0$.

7. The proof of Theorem 3.1 allows for consideration of the random sequences $\{\alpha_n\}$, $\{\beta_n^+\}$ and $\{\beta_n^-\}$ whose values at time instant n are measurable under correspondence σ-algebra $\mathcal{F}_{n-1}$. This fact is sometimes useful from a practical point of view.

3.4 Fastest Rate of Convergence

Here we consider a more concrete structure of distribution functions of test disturbances and some special kind of vector-valued functions $\mathbf{K}_n(\cdot)$, $n = 1, 2, \ldots$, which satisfies conditions (3.8) and can be used in algorithms (3.9) and (3.10). The estimates $\{\widehat{\mathbf{x}}(n)\}$ generated in this case achieve an asymptotic-optimal mean-square rate of convergence $\mathcal{O}(n^{-\frac{\gamma-1}{\gamma}})$ when the studied function satisfies the following smoothness condition.

As3.5: Fucntion $F(\cdot) \in C^\ell$, ℓ-times continuously differentiable and all its partial derivatives of the order up to ℓ inclusive satisfy on $\mathbb{R}^d$ the Hoelder condition of the order ν, $0 < \nu \leq 1$:

$$\left| F(\mathbf{u}) - \sum_{|\bar{k}| \leq \ell} \frac{1}{\bar{k}!} \mathcal{D}^{\bar{k}} F(\mathbf{x}^\star)(\mathbf{u} - \mathbf{x}^\star)^{\bar{k}} \right| \leq L \|\mathbf{u} - \mathbf{x}^\star\|^\gamma,$$

where L is a constant, $\gamma = \ell + \nu \geq 2$, $\mathbf{u} \in \mathbb{R}^d$,

$$\mathbf{u}^{\bar{k}} = u_1^{k_1} \cdots u_d^{k_d},\ \mathcal{D}^{\bar{k}} = \frac{\partial^{|\bar{k}|}}{(\partial u_1)^{k_1} \cdots (\partial u_d)^{k_d}},\ \bar{k} = \begin{pmatrix} k_1 \\ k_2 \\ \vdots \\ k_d \end{pmatrix}$$

is a multi-index, $| \bar{k} | = k_1 + \ldots + k_d, \bar{k}! = k_1! \cdots k_d!$, $k_i \geq 0$, $i \in 1..d$. For the case $\gamma = 2$ we assume $L = M/2$.

Let $\mathbf{\Delta}_n$, $n = 1, 2, \ldots$ be an observed sequence of mutually independent and identically distributed random vectors in $\mathbb{R}^d$, referred to below as the simultaneous trial perturbation. All components of the vector $\mathbf{\Delta}_n$ are mutually independent and have identical scalar distribution function $\mathrm{P}_\Delta(\cdot)$ with finite support. The kernel functions $\mathbf{K}_n(\cdot)$ do not depend on n and

$$\mathbf{K}_n(\mathbf{u}) = \mathbf{K}(\mathbf{u}) = \begin{pmatrix} \mathrm{K}_1(\mathbf{u}) \\ \mathrm{K}_2(\mathbf{u}) \\ \vdots \\ \mathrm{K}_d(\mathbf{u}) \end{pmatrix},\ \in \mathbb{R}^d,\ n = 1, 2, \ldots,$$

and its components $\mathrm{K}_i(\cdot),\ i \in 1..d$ obeying

$$\mathrm{K}_i(\mathbf{u}) = K_0(u_i) \prod_{j \neq i} K_1(u_j),\ i, j \in 1..d,\ \mathbf{u} \in \mathbb{R}^d, \tag{3.11}$$

where $K_0(\cdot)$ and $K_1(\cdot)$ are two scalar bounded functions (kernels) satisfying the conditions

$$\int uK_0(u)\mathrm{P}_{\Delta}(du) = 1, \int u^k K_0(u)\mathrm{P}_{\Delta}(du) = 0,\ k \in \{0\} \cup 2..\ell,$$

$$\int K_1(u)\mathrm{P}_{\Delta}(du) = 1,\ \int u^k K_1(u)\mathrm{P}_{\Delta}(du) = 0,\ k \in 1..\ell - 1. \tag{3.12}$$

We note that in the scalar case ($d = 1$), one function $K_0(u)$ suffices to define $\mathbf{K}(u)$ as $\mathbf{K}(u) = K_0(u)$.

It is easy to see that conditions (3.8) hold when the conditions (3.12) are satisfied for the function $\mathbf{K}(\cdot)$ and the distribution of the test disturbance is

$$\mathrm{P}_n(\mathbf{u}) = \mathrm{P}(\mathbf{u}) = \prod_{i=1}^{d} \mathrm{P}_{\Delta}(u_i).$$

Following Katkovnik [185] and with Polyak and Tsybakov [263], we show one possible method of constructing the kernel functions $K_0(\cdot)$ and $K_1(\cdot)$ satisfying conditions (3.12). Let $\{p_k(\cdot)\}_{k=0}^{\ell}$ be a system of orthogonal polynomials that is defined on the support of the distribution $\mathrm{P}_{\Delta}(\cdot)$. By analogy with the proofs of [263], one can readily verify that the functions

$$K_0(u) = \sum_{k=0}^{\ell} a_k p_k(u)\ , \qquad a_k = {p'}_k(0) / \int p_k^2(u)\mathrm{P}_{\Delta}(du)\ ,$$

$$K_1(u) = \sum_{k=0}^{\ell-1} b_k p_k(u)\ , \qquad b_k = p_k(0)\ /\ \int p_k^2(u)\mathrm{P}_{k\Delta}(du)$$

satisfy conditions (3.12).

It was proposed in [263] to take a distribution that is uniform over the interval $[-\frac{1}{2}, \frac{1}{2}]$ as the probabilistic distribution of the components of the trial perturbation $\mathrm{P}_{\Delta}(\cdot)$, and over this interval to construct the kernels $K_0(\cdot)$ and $K_1(\cdot)$ by the orthogonal Legendre polynomials. In this case, we obtain $K_0(u) = 12u$ and $K_1(u) = 1$, $|u| \leq 1/2$, for the initial values of $\ell = 1, 2$, that is, $2 \leq \gamma \leq 3$,$K_0(u) = 5u(15 - 84u^2)$ and $K_1(u) = 9/4 - 15u^2$, $|u| \leq 1/2$, for the subsequent values of $\ell = 3, 4$, that is, $3 < \gamma \leq 5$, and for $|u| > 1/2$ both functions are equal to zero.

Attention has been directed toward a type of probabilistic distribution of the trial perturbation that is more general than the one considered in [263].

This is because in practice stating the problem itself can define a certain type of distribution of the trial perturbation $\mathrm{P}_{\Delta}(\cdot)$ that can be conveniently modelled, whereas in some cases it is more suitable to use distributions only from some narrow fixed class. The possibility of choosing among various systems of orthogonal polynomials facilitates estimations of appropriate quality for the same asymptotic order of the convergence rate.

The following theorem from [140] establishes the conditions that are sufficient for optimality of the asymptotic convergence rate of algorithms (3.10).

Theorem 3.2. If *the following conditions are satisfied:* (3.12)*, As3.1–3.5,*

$$E\bar{v}_n^2 \le \sigma_v^2, \quad \alpha_n = \alpha\, n^{-1}, \quad \beta_n^+ = \beta_n^- = \beta\, n^{-\frac{1}{2\gamma}},\ \beta > 0,\ n = 1, 2, \ldots,$$

$$\gamma \ge 2,\ \alpha > \frac{\gamma - 1}{2\mu\gamma},$$

then *for the mean-square convergence rate of the sequence of estimates* $\{\widehat{\mathbf{x}}(n)\}$ *generated by algorithm* (3.10) *we have*

$$E\|\widehat{\mathbf{x}}(n) - \mathbf{x}\|^2 = \mathcal{O}(n^{-\frac{\gamma-1}{\gamma}}) \tag{3.13}$$

expressed asymptotically as $n \to \infty$.

A similar result is proved in [140] for algorithm (3.9) too. Moreover, it is shown in [263] that the asymptotic mean-square convergence rate (3.13) is optimal in the sense of minimax estimation among all possible iterative algorithms for the sufficiently wide class of functions $F(\cdot)$ which satisfy the condition *As3.5* with the same constants.

3.5 Tracking

In this section application of the randomized stochastic approximation algorithm (3.10) is considered for non-constrained optimization in the context of the minimum tracking problem. In the case of once differentiable functional and almost arbitrary noises, the upper bound of a mean square estimation error is derived. It can have a sufficiently small level compared to a significant level of noise.

To analyze the quality of estimates we apply the following definition for the problem of minimum tracking for mean-risk functional (3.5):

Definition 3.3. *A sequence of estimates* $\widehat{\mathbf{x}}(n)$ *has an asymptotically efficient mean-square upper bound* $\bar{L} > 0$ *of residues of estimation if* $\forall \varepsilon > 0\ \exists N$ *such that* $\forall n > N$

$$E\|\widehat{\mathbf{x}}(n) - \mathbf{x}_n\|^2 \le \bar{L} + \varepsilon.$$

In the non-stationary problem (3.5) stochastic approximation algorithms are commonly used with constant step-sizes [49, 326]. Let us consider the randomized SA algorithm with two measurements (3.10), where $\alpha_n \equiv \alpha$, $\beta_n^+ \equiv \beta$, $\beta_n^- \equiv 0$, $\mathbf{K}_n(\mathbf{\Delta}_n) = \mathbf{\Delta}_n$, and $\mathbf{\Delta}_n$ be a sequence of independent identically distributed Bernoulli random vectors whose components are independent and equal $\pm\frac{1}{d}$ with the same probability 1/2, taking the form

$$\begin{cases} \mathbf{u}_{2n} = \widehat{\mathbf{x}}(2n-1) + \beta\mathbf{\Delta}_n,\ \mathbf{u}_{2n-1} = \widehat{\mathbf{x}}(2n-1), \\ \widehat{\mathbf{x}}(2n-1) = \widehat{\mathbf{x}}(2n-2), \\ \widehat{\mathbf{x}}(2n) = \widehat{\mathbf{x}}(2n-1) - \frac{\alpha}{\beta}\mathbf{\Delta}_n(y_{2n} - y_{2n-1}). \end{cases} \tag{3.14}$$

Figure 3.1 shows a block diagram of algorithm (3.14) that is used for the recurrent optimization of a non-stationary functional (3.5).

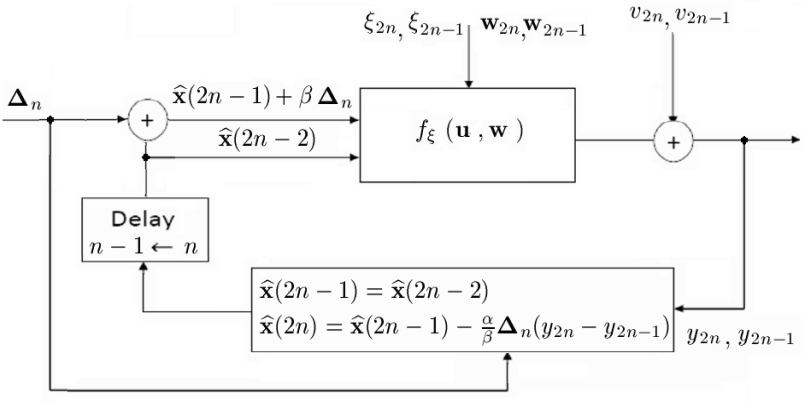

Fig. 3.1. The diagram of the algorithm (3.14)

We further assume that the additional following conditions are true.

As3.5: The gradient of ∇f_{ξ_t} is uniformly bounded at the minimum points: $E\|\nabla f_{\xi_t}(\mathbf{x}_t, \mathbf{w}_t)\|^2 \le \bar{g} < \infty$, ($\bar{g} = 0$ for the case $f_{\xi_t}(\mathbf{x}, \mathbf{w}) = F_t(\mathbf{x})$).

As3.6: Drift is bounded $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \le \delta_{\mathbf{x}} < \infty$ and for any arbitrary point $\mathbf{u}$: $E_{\mathcal{F}_{t-1}}\varphi_t(\mathbf{u})^2 \le \bar{a}\|\mathbf{u} - \mathbf{x}_{t-1}\|^2 + \bar{b}$, where $\varphi_t(\mathbf{u}) = f_{\xi_t}(\mathbf{u}, \mathbf{w}_t) - f_{\xi_{t-1}}(\mathbf{u}, \mathbf{w}_{t-1})$.

As3.7: The observation noise $\{v_n\}$ satisfies: $|v_{2n} - v_{2n-1}| \le c_v < \infty$.

The following theorem shows the asymptotically efficient upper bound of estimation residues by algorithm (3.14).

Theorem 3.4. *Let the conditions As3.1–3.7 hold.*
If $k = 2\mu - 2\alpha(M^2 + 2\frac{\bar{a}}{\beta}) < 1/\alpha$ *and* $\alpha < \mu/M^2$,

then *the sequence of estimates provided by the algorithm* (3.14) *has an asymptotically efficient mean-square upper bound which is equal to*

$$\bar{L} = \left(\frac{4\delta_{\mathbf{x}}}{k\alpha} + \delta_{\mathbf{x}}\right)^2 - \delta_{\mathbf{x}}^2 + \frac{l}{k}, \tag{3.15}$$

where

$$l = 4\alpha\left(\frac{c_v^2 + 2\bar{b}}{\beta^2} + 3M^2 + 2\bar{g}\right) + 4\beta M.$$

Proof. Let $\nu_n = \|\widehat{\mathbf{x}}(2n) - \mathbf{x}_{2n}\|$, $\mathbf{d}_n = \widehat{\mathbf{x}}(2n-2) - \mathbf{x}_{2n}$, $\bar{f}_n = f_{\xi_{2n}}(\mathbf{u}_{2n}, \mathbf{w}_{2n}) - f_{\xi_{2n-1}}(\mathbf{u}_{2n-1}, \mathbf{w}_{2n-1})$, $s_n = \frac{\alpha}{\beta}(\bar{f}_n + \bar{v}_n)\mathbf{\Delta}_n = \frac{\alpha}{\beta}(y_{2n} - y_{2n-1})\mathbf{\Delta_n}$. According to the algorithm (3.14), observation model (3.6) and assumption *As3.5,3.6*, we can bound ν_n^2 as follows:

$$\begin{aligned} \nu_n^2 &= \|\mathbf{d}_n\|^2 + s_n^2 - 2\langle \mathbf{d}_n, s_n\rangle, \\ \|\mathbf{d}_n\|^2 &\le \nu_{n-1}^2 + 4\delta_{\mathbf{x}}\nu_{n-1} + 4\delta_{\mathbf{x}}^2. \end{aligned} \tag{3.16}$$

Consider the difference $\bar{f}_n$. By virtue of the presentation of $f_{\xi_{2n}}(\mathbf{u}_{2n-1} + \beta\mathbf{\Delta}_n, \mathbf{w}_{2n})$ as a Taylor series, we conclude $\bar{f}_n =$

$$f_{\xi_{2n}}(\mathbf{u}_{2n-1} + \beta\mathbf{\Delta}_n, \mathbf{w}_{2n}) - f_{\xi_{2n}}(\mathbf{u}_{2n-1}, \mathbf{w}_{2n}) + \varphi_{2n}(\mathbf{u}_{2n-1}) =$$

$$\langle \nabla f_{\xi_{2n}}(\mathbf{u}_{2n-1} + \gamma\beta\mathbf{\Delta}_n, w_{2n}), \beta\mathbf{\Delta}_n\rangle + \varphi_{2n}(\mathbf{u}_{2n-1}) =$$

$$\langle \nabla f_{\xi_{2n}}(\mathbf{u}_{2n-1}, w_{2n}), \beta\mathbf{\Delta}_n\rangle + \langle \nabla_n(\rho), \beta\mathbf{\Delta}_n\rangle + \varphi_{2n}(\mathbf{u}_{2n-1}),$$

where $\rho \in (0,1)$, $\nabla_n(\rho) = \nabla f_{\xi_{2n}}(\mathbf{u}_{2n-1} + \rho\beta\mathbf{\Delta}_n, w_{2n}) - \nabla f_{\xi_{2n}}(\mathbf{u}_{2n-1}, w_{2n})$.

Taking the conditional expectation over σ-algebra $\tilde{\mathcal{F}}_{n-1} = \sigma\{\mathcal{F}_{n-1}, \mathbf{w}_{2n-1}, \mathbf{w}_{2n}, \xi_{2n-1}, \xi_{2n}\}$ of both sides of the inequality (3.16) we can bound $E_{\tilde{\mathcal{F}}_{n-1}}\nu_n^2$ as follows

$$E_{\tilde{\mathcal{F}}_{n-1}}\nu_n^2 \le \|\mathbf{d}_n\|^2 + 2\frac{\alpha^2}{\beta^2}c_v^2 + E_{\tilde{\mathcal{F}}_{n-1}}2\frac{\alpha^2}{\beta^2}\bar{f}_n^2 - 2\langle \mathbf{d}_n, \frac{\alpha}{\beta}E_{\mathcal{F}_{n-1}}\bar{f}_n\mathbf{\Delta}_n\rangle \tag{3.17}$$

since $E_{\tilde{\mathcal{F}}_{n-1}}\bar{v}_n\mathbf{\Delta}_n = E_{\tilde{\mathcal{F}}_{n-1}}\bar{v}_n E_{\tilde{\mathcal{F}}_{n-1}}\mathbf{\Delta}_n = E_{\tilde{\mathcal{F}}_{n-1}}\bar{v}_n \cdot 0 = 0$ by virtue of the assumption *As3.4*.

Assumptions *As3.2,3.6* make it possible to derive

$$E_{\mathcal{F}_{n-1}}\bar{f}_n^2 \le \beta^2(2M^2\|\mathbf{d}_n\|^2 + 2\bar{g} + 3M^2) + 2(\bar{a}\nu_{n-1}^2 + \bar{b}), \tag{3.18}$$

For the last term in (3.17), applying *As3.1–3.3*, we get

$$-2\langle \mathbf{d}_n, \frac{\alpha}{\beta}E_{\tilde{\mathcal{F}}_{n-1}}\bar{f}_n\mathbf{\Delta}_n\rangle \le -2\langle \mathbf{d}_n, \alpha F_n(\mathbf{u}_{2n-1})\rangle +$$

$$+2\frac{\alpha}{\beta}\|\mathbf{d}_n\|\|E_{\tilde{\mathcal{F}}_{n-1}}|\langle \nabla_n(\rho), \beta\mathbf{\Delta}_n\rangle\| - 2\langle \mathbf{d}_n, \frac{\alpha}{\beta}E_{\tilde{\mathcal{F}}_{n-1}}\varphi_{2n}(\mathbf{u}_{2n-1})\mathbf{\Delta}_n\rangle \le$$

$$\leq -2\mu\alpha\|\mathbf{d}_n\|^2 + 2\alpha M\beta \tag{3.19}$$

since $E_{\bar{\mathcal{F}}_{n-1}}\varphi_{2n}(\mathbf{u}_{2n-1})\mathbf{\Delta}_n = 0$ by virtue of the assumption *As3.4*.

Summing up the findings bounds (3.18),(3.19) and taking the conditional expectation over σ-algebra $\mathcal{F}_{n-1}$, we derive the following from the (3.17)

$$E_{\mathcal{F}_{n-1}}\nu_n^2 \leq (1-k\alpha)\nu_{n-1}^2 + 4\delta_{\mathbf{x}}\nu_{n-1} + 4\delta_{\mathbf{x}}^2 + \alpha l/2. \tag{3.20}$$

Using the inequality $2\delta_{\mathbf{x}}\nu_{n-1} \leq \epsilon\alpha\nu_{n-1}^2 + \frac{\delta_{\mathbf{x}}^2}{\epsilon\alpha}$, which is true $\forall\epsilon > 0$, and taking the unconditional expectation, we obtain

$$E\nu_n^2 \leq (1-(k-\epsilon)\alpha)E\nu_{n-1}^2 + 2\frac{\delta_{\mathbf{x}}^2}{\epsilon\alpha} + 4\delta_{\mathbf{x}}^2 + \alpha l/2. \tag{3.21}$$

For any $0 < \epsilon < k$ by Lemma 1 of Chapter 2 of [258] it follows that

$$\bar{L} \leq \frac{\frac{2\delta_{\mathbf{x}}^2}{\epsilon\alpha^2} + \frac{4\delta_{\mathbf{x}}^2}{\alpha} + l/2}{k-\epsilon}, \tag{3.22}$$

which is similar to (3.15) when $\epsilon = k/2$. This completes the proof of Theorem 3.4.

Remarks 1. Conditions about the drift *As3.5,3.6* hold for the drift with model

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \zeta,\ \mathbf{x}_n \in \mathbb{R}^d$$

when ζ is symmetrically distributed on the ball: $\|\zeta\| \leq 1$, and we can measure the squared distance $\|\mathbf{u}-\mathbf{x}_n\|^2$ between the chosen point $\mathbf{u}$ and $\mathbf{x}_n$ with additive non-random noise v_n: $\|v_n\| < 1$. In this case, we have $F_n(\mathbf{u}) = \|\mathbf{u}-\mathbf{x}_n\|^2$, $\delta_{\mathbf{x}} = 1$, $\bar{g} = 0$, $\bar{a} = 8E\|\zeta\|^2$, and $\bar{b} = 8E\|\zeta\|^2 + E\|\zeta\|^4$.

2. The first condition of the drift *As3.6* can be weakened for the case of a mean-squared bounded drift.

3. The result of the Theorem 3.4 shows that for the case without drift ($\delta_{\mathbf{x}} = 0$) the asymptotic upper bound is $\bar{L} = l/k$. This can be made infinitely small under any noise level c_v, simply by choosing α and β to be sufficiently small. At the same time in the case of drift, the bigger drift norm $\delta_{\mathbf{x}}$ can be compensated by choosing a bigger step-size α. This leads to a tradeoff between making α smaller because of noisy observations and making α bigger due to the drift of optimal points.

3.6 Algorithm Implementation and Quantum Computing

A sample program in MATLAB implementing N iterations of algorithm (3.9) can be written as follows:

$FOR\ n = 1 : N$
$\quad delta = round(2 * (rand(d, 1) - 1));$
$\quad xplus = x + b * delta;$
$\quad yplus = valueY(xplus);$
$\quad ghat = delta * yplus/b;$
$\quad x = x - a * ghat;$
$END;$
$x;$

The initialization algorithm is not shown, as it can differ depending on the application. The program is called an outer function $valueY(\cdot)$. The components of the test perturbation $\mathbf{\Delta}_n$ are generated according to the Bernoulli distribution ± 1.

Let us examine the choice of the best computer for implementing the randomized stochastic optimization algorithm with one measurement of the penalty function in iterations. The realization of algorithm (3.9) on a quantum computer is described in [325]. Recently, the terminology and axiomatics of quantum computation models have been greatly refined (see Section 1.6). Below we describe a method for implementing algorithm (3.9) on a hypothetical quantum computer, i.e., a method that is consistent with the general logic of quantum computation algorithms.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function satisfying the conditions *As3.1–3.3*. Let us assume that the quantum computer is an r-bits machine, i.e. it operates with r-bits numbers. The unitary operation realizing the function $f(\mathbf{x})$ on a quantum computer can be defined on all classical binary strings $\mathbf{x}$ of length dr, defining the argument of the function

$$\mathbf{U}_f : |\mathbf{x}\rangle|z\rangle \to |\mathbf{x}\rangle|z \oplus f(\mathbf{x})\rangle,$$

where z is an arbitrary binary chain of length r and $\oplus$ is a bit-by-bit operation of logical AND. This is a method of defining an operator on base vectors. On all other vectors, the operator is continued linearly. Clearly, the operator thus constructed is invertible and acts in a complex space of dimension 2^{dr+r}.

We estimate the minimum of a function using algorithm (3.9) where the iteration takes the following form

$$\begin{aligned} \mathbf{u} &= \widehat{\mathbf{x}} + \beta\mathbf{\Delta}, \\ y &= f(\mathbf{u}) + v, \\ \widehat{\mathbf{x}} &:= \widehat{\mathbf{x}} - gamma(\mathbf{u} - \widehat{\mathbf{x}})y, \end{aligned}$$

where $\gamma = \alpha/\beta^2$.

As input, let us prepare a superposition of $2d$ perturbed values of the current estimate vector

$$\mathbf{u} = \frac{1}{2^{\frac{d}{2}}} \sum_{\mathbf{\Delta}_i \in \{-1,+1\}^d} |\widehat{\mathbf{x}} + \beta\mathbf{\Delta}_i\rangle = \mathbf{H}_\beta|\widehat{\mathbf{x}}\rangle,$$

where ± 1 are regarded as r-bits numbers, $\mathbf{H}_\beta$ is the unitary operator corresponding the Hadamard transformation (1.15) in Section 1.6.

Applying the unitary operator $\mathbf{U}_f$ to $|\mathbf{u}\rangle|0\rangle$, we obtain

$$\mathbf{U}_f|\mathbf{u}\rangle|0\rangle = \frac{1}{2^d}\sum_{\boldsymbol{\Delta}_i\in\{-1,+1\}^d} |\widehat{\mathbf{x}}+\beta\boldsymbol{\Delta}_i\rangle|f(\widehat{\mathbf{x}}+\beta\boldsymbol{\Delta}_i)\rangle.$$

By the general properties of the quantum computation model, after a state measurement, we obtain a vector

$$|\widehat{\mathbf{x}}+\beta\boldsymbol{\Delta}_i\rangle|f(\widehat{\mathbf{x}}+\beta\boldsymbol{\Delta}_i)\rangle, \quad \boldsymbol{\Delta}_i \in \{-1,+1\}^d.$$

with probability $\frac{1}{2^d}$. Using the first dr bits of this vector, we can easily determine a random perturbation vector $\boldsymbol{\Delta}_n$. According to algorithm (3.9), its coordinates must be multiplied by the corresponding value of the loss function at a perturbed point, i.e., by the value at the last r bits of the measurement result.

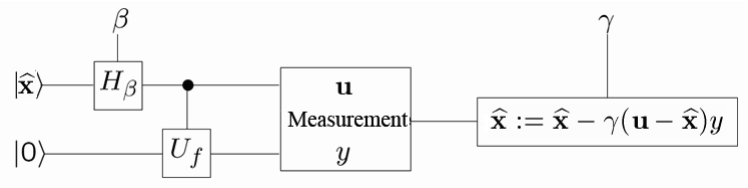

Fig. 3.2. The quantum circuit for "on the fly" computing of the gradient

Figure 3.2 shows the quantum circuit for "on the fly" computing of the function f gradient.

3.7 Applications

3.7.1 Optimization of a Server Processing Queue

The problem of efficient maintenance of the server job queue is one of the classic problems of queuing theory. Different algorithms have been developed and numerous studies have been conducted to solve this problem. The possibility of using adaptive methods in this task has not been thoroughly studied. A number of methods proposed and studied in [206, 310] are based on the use of infinitesimal perturbation analysis technology. In [342], a randomized algorithm of a stochastic approximation was proposed for adaptive tuning of a server switching interval (or scheduling step-size).

For example, consider utilization of some corporate network. At the beginning and end of the day the network is dominated by short-term and frequent

requests to the server, while at the middle of the day the jobs comes fewer and more time consuming. Not only can such fluctuations occur during the day, they can also depend on the season, day of the week, and so on. Note that for more efficient server operation, server performance can be adjusted to the type of incoming jobs, yet the question of how to define the boundaries of the periods still remains. Perhaps research and monitoring can yield approximate desired values. This method is not very good because in the real world it is difficult to identify the direct function of such factors. Moreover, such research is very expensive and requires repetitive work, even for minor changes. The goal is to be able to react to changes in these values in real time, depending on the type of incoming jobs.

The problem of improving server efficiency for servicing jobs in the queue where the arrival process is random can be formalized as follows.

Let us assume that the probability distribution of the server service time of jobs depends on a real parameter x which we want to select (to make a control action) in order to minimize the average waiting time of customers $L(x)$ with a cost value $q(x)$ of a parameter x

$$F(x) = q(x) + L(x) \equiv q(x) + \lim_{T} \frac{1}{T} \sum_{t=1}^{T} Ey_t(x) \ \rightarrow \ \min_{x},$$

where $y_t(x)$ is the time of t-th job processing and waiting in a server queue. Here the index t means that t-th job has a numbering with order when they were finished.

The problem is *to find the parameter* $x^\star$ that minimizes $F(x)$ for $x \in [x_-, x_+] \subset \mathbb{R}$.

Generally speaking, the function $F(x)$ is very difficult to calculate. Let x be fixed. By watching the queue for a long time (recording the time the order is received, the service time and sending time for each customer), we can use the data to estimate $\widehat{G}$ the functional derivative $\nabla F(x)$ for the current estimate which is a result of some gradient algorithm. Here the term estimate of the derivative is understood in a wide sense, as it can turn out to be biased. In fact, we have to assume that the resulting estimate of the derivative depends not only on the chosen point of x but also on a set $\mathbf{w}$ of uncontrollable disturbances that are determined by the server contexts on the time interval selected for research, including the sequence of requests actually implemented. The main difficulty, common to many real-time systems, is that optimization should be obtained only from observations of the specific implementation of the trajectory of a random variable. The experimenter cannot choose the values of the function in a neighborhood of x to generate estimates for the derivative at this point. That is, this optimization problem cannot be solved by traditional tools, and a standard finite difference method for evaluating the quality of the functional derivative is not appropriate to this problem. A more adequate approach to the solution is to use algorithms that are based on empirical functionals $f(x, \mathbf{w})$ counted at certain intervals such as

$$F(x) = E_{\mathbf{w}} f(x, \mathbf{w}).$$

In addition to these difficulties, the optimal value x of a server switching interval drifts usually (as in the above example of variations in the parameters of the input streams into corporate computer networks). Therefore, tracking the minimum point changing is a no less urgent problem than the optimization problem. More precisely, let $f_t(x, \mathbf{w})$ be a random function of a discrete-time $t = 1, 2, \ldots$, a server parameter x, and a vector of uncontrolled disturbances $\mathbf{w}$. Denote by F_t the "current" mean-risk functional:

$$F_t(x) = E_{\mathbf{w}} f_t(x, \mathbf{w}).$$

Let

$$x_t = \arg\min_x F_t(x)$$

be the minimum point of the functional F_t.

By choosing x at certain time intervals, we assume that we can measure the value of the corresponding "current" empirical functional $f_t(x, \mathbf{w})$. In this case, we naturally assume that the drift rate of an optimal parameter is so low that we are able to obtain a series of values of $f_t(x_i, \mathbf{w}_i),\ i = 1, 2, \ldots$.

A sequence of estimates $\{\widehat{x}(t)\}$ tracking the optimal parameters x_t *must be constructed* based on observations of the random variables $f_t(x_i, \mathbf{w}_i),\ i = 1, 2, \ldots$ (perhaps with additional noise). That is, we would like

$$|\widehat{x}(t) - x_t| \to \min$$

in some reasonable sense.

Input data determine the sequence of pairs of values $\{(t_i^{in}, d_i)\}$ where the first one corresponds to the time of the job's arrival on the server, and the other is the time required for executing the job (job duration). Naturally, the value d_i is a priori unknown for the server.

For simplicity, we assume that the server processes the job as follows:

- a server switching interval (scheduling step-size) x is specified as $x \in [x_-, x_+] \subset \mathbb{R},\ x_- > 0$;
- a first-job starts immediately after arrival into the server;
- incoming jobs come into a queue;
- at any given time, the server handles only one current job;
- at switching times, which are multiple selected intervals x (scheduling step-size), if the server is free it switches to the execution of the first task in the queue; if it is occupied, the currently processed job is interrupted and goes to the back of the queue, and the server switches to the first job from the queue. This server "loading-unloading" operation always takes a fixed time interval d_{load}.

We choose a positive sufficiently large N (for example, $N = 1000$). We divide the total server time into regular intervals — bars: $t_0, t_1, \ldots, t_n, \ldots$, according to the rule:

t_0 — the start of the server;
t_1 — the end time of the execution of the first N tasks;
...;
t_n — the end time of the execution of n-th series of N tasks;
...

Note that we can actually set the bars in different ways: either by starting a new time interval for completion of the processing of some fixed number of jobs (as described above), or after a fixed number of jobs enter the server, or by choosing a fixed period of server time, and so on. Reasonable rules for the selection of intervals are largely equivalent (see [310]).

We describe a method for obtaining the values $y_n,\ n = 1, 2, \ldots$ of the empirical functional $f(x, \mathbf{w})$. We suggest that for every n-th interval an appropriate value x for adjustable parameter is somehow chosen. For convenience, we number the jobs performed by the server according to the order of the end of their processing. Along with a pair (t_i^{in}, d_i) which was previously defined at the time of its arrival, for each job we additionally associate — t_i^{out} — the end time of processing. The value y_n of the empirical quality function is determined as the sum of the handler's cost of loading and unloading and the average useless waiting time of tasks before completing them

$$y_n = \frac{t_n - t_{n-1}}{x} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{n-1}, t_n]} (t_i^{out} - t_i^{in} - d_i). \tag{3.23}$$

Adaptation Algorithms. *SPSA with one measurement:*

1. *Initialization.* Set a counter index $n = 0$. Choose $\widehat{x}(0) \in [x_-, x_+]$ and fairly small step-sizes $\alpha > 0$ and $\beta > 0$.
2. *Iteration* $n \to n + 1$.
   2a. $n := n + 1$.
   2b. Generate the random value Δ_n according to the Bernoulli distribution
   $$\Delta_n = \begin{cases} +1, \text{ with probability } \frac{1}{2}; \\ -1, \text{ with probability } \frac{1}{2}. \end{cases}$$
   2c. Compute the next input (measurement point) by the rule
   $$u_n = \mathcal{P}_{[x_-, x_+]}(\widehat{x}(n-1) + \beta \Delta_n)$$
   where $\mathcal{P}_{[x_-, x_+]}(\cdot)$ is a projector into the interval $[x_-, x_+]$.
   2d. Start the server with the parameter u_n and wait until the next N jobs are finished.
   2e. Obtain the new empirical value y_n by the formula (3.23).

$$y_n = \frac{t_n - t_{n-1}}{u_n} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{n-1}, t_n]} (t_i^{out} - t_i^{in} - d_i).$$

2f. Get the new estimate

$$\widehat{x}(n) = \mathcal{P}_{[x_-,x_+]}\left(\widehat{x}(n-1) - \frac{\alpha}{\beta}\Delta_n y_n\right).$$

3. Repeat Step 2.

SPSA with two measurements differs from the previous algorithm in only three operations:

2d. 2d-1. Start the server with the parameter $\widehat{x}(n-1)$ and wait until the next N jobs are finished.
2d-2. Start the server with the parameter u_n and wait until the next N jobs are finished.

2e. Obtain the new empirical values y_{2n-1} and y_{2n} by the following:

$$y_{2n-1} = \frac{t_{2n-1} - t_{2n-2}}{\widehat{x}(n-1)} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{2n-2}, t_{2n-1}]} (t_i^{out} - t_i^{in} - d_i),$$

$$y_{2n} = \frac{t_{2n} - t_{2n-1}}{u_n} d_{load} + \frac{1}{N} \sum_{t_i^{out} \in [t_{2n-1}, t_{2n}]} (t_i^{out} - t_i^{in} - d_i).$$

2f. Get the new estimate

$$\widehat{x}(n) = \mathcal{P}_{[x_-,x_+]}\left(\widehat{x}(n-1) - \frac{\alpha}{\beta}\Delta_n (y_{2n} - y_{2n-1})\right).$$

Simulation. The input data were chosen as a pseudorandom sequence of pairs (t_i^{in}, d_i) generated by exponential distributions with parameters η_{in} and η_d respectively:

$$t_i^{in} = t_{i-1}^{in} - \frac{1}{\eta_{in}} \ln \omega_i^{in},$$
$$d_i = -\frac{1}{\eta_d} \ln \omega_i^{d},$$

where ω_i^{in}, ω_i^{d} are some samples that are uniformly distributed over the interval $[0, 1]$.

Figure 3.3a shows the typical dependence of average empirical functional values on a server switching interval (scheduling step-size) x when simulation parameters are chosen as $d_{load} = 0.05$, $\eta_{in} = 0.5$ and $\eta_d = 0.75$. The minimum value corresponds to the point x which is close to 0.17. Figure 3.3b illustrates the unstable behavior of a server when it works with a fixed parameter $x = 0.05$. The figure indicates the low efficiency of the processing job queue. In contrast, using the discussed randomized algorithm for server switching interval adaptation under the same starting conditions over time can reach a level that is close to the average minimum possible value. That is, the server started to work with a non-optimal parameter x, over time,

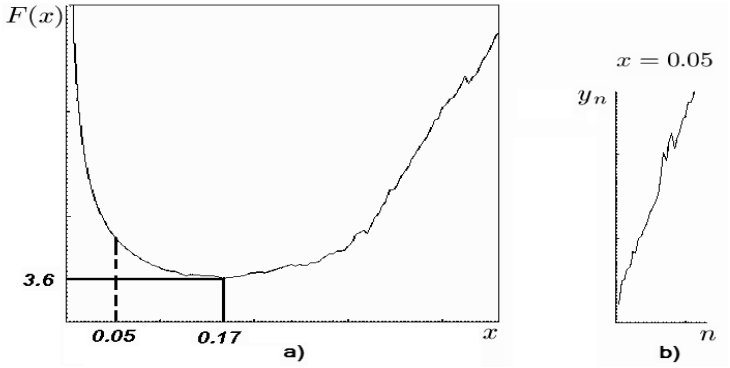

Fig. 3.3. a) Optimization of an empirical functional with $d_{load} = 0.05$, $\eta_{in} = 0.5$, $\eta_d = 0.75$. b) Unstable behavior of a server for $x = 0.05$.

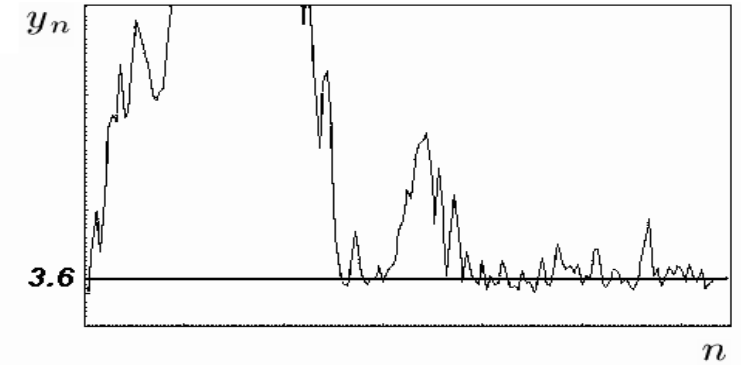

Fig. 3.4. Adapting to the optimal parameter

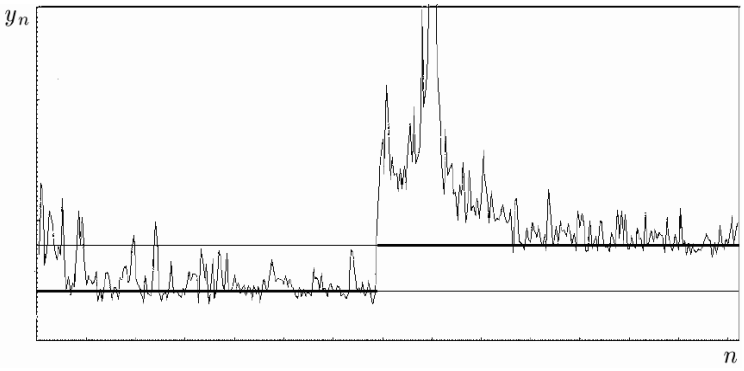

Fig. 3.5. The behavior of y_n for tuning abrupt changes in the parameters of the input stream

adapting to the specific (unknown) parameters of the input stream. Typical behavior of empirical functional values with fine-tuning x is shown in Figure 3.4.

Figure 3.5 shows an example of server adaptation to an abrupt change in the parameters of the input stream.

3.7.2 Load Balancing

Let us assume a system with m computing nodes (processors). The system works iteratively by processing the jobs package of a known size z_n at each iteration n Suppose that the entire package can be arbitrarily divided into m tasks u_j, $j \in 1..m$,

$$\sum_{j=1}^{m} u_j = z_n,$$

for all nodes, and the computation time of the node j is defined by

$$t_j(u_j) = x_j u_j,$$

where $x_j \in \mathbb{R}$ is a value equal to the inverse productivity (performance) of node j.

The problem *is to minimize* the total time of processing the jobs package z_n:

$$T(\mathbf{u}) = ||\mathbf{t}(\mathbf{u})||_\infty = \max_{j \in 1..m} t_j(u_j) \to \min_{\mathbf{u}}. \tag{3.24}$$

Here we denote $\mathbf{u} = (u_1, u_2, \ldots, u_m)^{\mathrm{T}}$, $\mathbf{t} = (t_1(u_1), t_2(u_2), \ldots, t_m(u_m))^{\mathrm{T}}$.

An ideal scheduling algorithm is one that keeps all the nodes busy executing essential tasks, and minimizes the internode communication required to determine the schedule and pass data between tasks. The scheduling problem is particularly challenging when the tasks are generated dynamically and unpredictably in the course of executing the algorithm. This is the case with many recursive divide-and-conquer algorithms, including backtrack search, game tree search and branch-and-bound computation.

When the productivity (performance) of nodes is known, the best control strategy is a proportional distribution of tasks such that

$$x_1 u_1 = x_2 u_2 = \cdots = x_m u_m.$$

This control strategy $\mathbf{u} = \mathcal{U}(\mathbf{x}, z_n)$ is called "load balancing". Here we denote $\mathbf{x} = (x_1, x_2, \ldots, x_m)^{\mathrm{T}}$

In practice, the productivity (performance) of nodes may be unknown. Moreover, they may be distorted because of side jobs, that is, $\mathbf{x}_n = \mathbf{x}^\star + \mathbf{w}_n$, or they may change with time: $\mathbf{x}_n = \mathbf{x}_{n-1} + \xi_n$, where $\mathbf{w}_n, \xi_n \in \mathbb{R}^m$ are vectors of independent random variables.

The usual method is to use estimates of the productivity (performance) $\widehat{\mathbf{x}}(n)$ at each iteration n which are defined in a such a way that

$$||\widehat{\mathbf{x}}(n) - \mathbf{x}_n||^2 \to \min \tag{3.25}$$

in some reasonable sense, and to compute $\mathbf{u}_n$ at the iteration n as

$$\mathbf{u}_n = \mathcal{U}(\widehat{\mathbf{x}}(n), z_n).$$

One reasonable quality functional suggested in [144] is

$$F(\widehat{\mathbf{x}}_n) = \sum_{j,k=1}^{m} (\bar{t}_n^j - \bar{t}_n^k)^2 \ \to \ \min_{\widehat{\mathbf{x}}_n}, \tag{3.26}$$

(We use notation $\bar{t}_n^j = t^j(\mathcal{U}^j(\widehat{\mathbf{x}}_n), z_n)/z_n,\ j = 1, 2, \ldots, m.$) Function $F(\cdot)$ has a minimum point which corresponds to the optimal control strategy minimized (3.24).

To identify the optimal vector $\mathbf{x}^\star$ or to tracking changes $\mathbf{x}_n$, it is advisable to use randomized stochastic approximation algorithms (or SPSA, simultaneous perturbation stochastic approximation).

Algorithms. *SPSA with two measurements:*

1. *Initialization and coefficient selection.* Set a counter index $n = 0$. Choose initial guess $\widehat{\mathbf{x}}(0) \in \mathbb{R}^m$ and fairly small step-sizes $\alpha > 0$ and $\beta > 0$.
2. *Iteration* $n \to n + 1$.
   2a. $n := n + 1$.
   2b. Generate the random vector $\boldsymbol{\Delta}_n$ according to the Bernoulli distribution of i.i.d. components that are equal to ± 1 with probability $\frac{1}{2}$.
   2c-1. Obtain the next jobs package z_{2n-1}.
   2d-1. Compute the next inputs by the rule

$$\mathbf{u}_{2n-1} = \mathcal{U}(\widehat{\mathbf{x}}(n-1), z_{2n-1}).$$

   2e-1. Start the cluster with input $\mathbf{u}_{2n-1}$ and wait until all tasks are finished.
   2c-2. Obtain the next jobs package z_{2n}.
   2d-2. Compute the next inputs by the rule

$$\mathbf{u}_{2n} = \mathcal{U}(\widehat{\mathbf{x}}(n-1) + \beta\boldsymbol{\Delta}_n, z_{2n}).$$

   2e-2. Start the cluster with input $\mathbf{u}_{2n}$ and wait until all tasks are finished.
   2f. Get the new estimate

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \frac{\alpha}{\beta}\boldsymbol{\Delta}_n(y_{2n} - y_{2n-1}).$$

3.7.3 UAV Soaring

Extending the endurance of the flight of UAVs (Unmanned Air Vehicles) is currently an area of major research interest because these vehicles are

very popular for aircraft missions that would be too dangerous or too boring for human pilots. Military surveillance missions or commercial atmospheric satellites need extremely long UAV flight endurance. Large birds and glider pilots commonly use updrafts caused by convection in the lower atmosphere to extend flight duration, increase cross-country speed, improve range and conserve energy. Small UAVs may also have the ability to exploit updrafts to improve performance. This subsection presents a SPSA-based algorithm for quick and precise detection of the center of a thermal updraft where the vertical velocity of the air stream is highest. This is a slight improvement on the algorithm considered in [18]. It allows maximizing the flight duration of a single UAV and of UAV groups using the thermal model developed by Allen at NASA Dryden [8].

The method takes into account the unstable behavior of updraft dynamics and the drift of its center over time. This method allows for effective treatment of the updraft center drift because of the tracking properties of SPSA shown above It also helps compensate the effect of horizontal wind considered as systematic (arbitrary) noise.

The airplane we modeled was based on a small unmanned powered glider. The objective of our vehicle is to conserve battery energy and soar as long as possible over the test area. In our experiments the UAV uses a very simple strategy. It flies along a predefined path, measuring the vertical airspeed using the readings of an onboard GPS module. Thermal updrafts are identified as areas with positive airspeed values. The UAV should locate thermal updrafts within its flight path and use them to gain altitude. After climbing to the maximum available altitude it should return to its course and use the energy obtained by switching to soaring mode, that is by gliding while keeping its engine off.

Based on observations, the maximum vertical speed in the updraft can be found at its center, while the edges remain relatively still. In fact, the vertical speed on the outskirts of the updraft is usually negative because the air is circulating inside the updraft. It is therefore vital for the UAV to detect the updraft center as quickly as possible in order to benefit from its energy.

Consider the problem of detecting the center of a thermal. Methods already used for this purpose usually yield inaccurate results. Moreover, either they do not take into account the drift of the updraft's center in time and the horizontal wind influence on the calculations or they use very complicated models for this purpose.

Constraints: It is absolutely necessary for the UAV to be able to glide with minimal altitude loss, in order to use the potential energy efficiently. The soaring flight can only take place during daytime and the UAV path may not cross large water basins, as updrafts do not form over water or in darkness.

We assume that the vehicle uses an onboard GPS receiver to calculate its vertical velocity at each point.

It is known that thermal updrafts often end with clouds at their tops. The wind conditions inside the clouds are known to be extremely severe, with rapid upward and downward streams. Entering the cloud level will usually result in losing the UAV. We therefore consider that an algorithm exists that will allow the UAV to detect the top of the updraft and prevent it from entering the clouds.

We introduce an altitude threshold above which the UAV will not run the algorithm of thermal center location. Upon detecting the area with a vertical velocity value above a fixed threshold and providing the unit altitude is below the altitude threshold, the UAV will initiate the SPSA algorithm with a 2-dimensional vector of coordinates $\mathbf{x}$ and velocity value as the profit function $F(\mathbf{x})$. In our model we assume that the updraft has a Gaussian velocity distribution. In the updraft center the velocity reaches its highest value. Thus, in order to use the energy of the updraft effectively we need an algorithm capable of detecting the maximum value of a function in a very noisy environment. Under these conditions we are facing a classical optimization problem.

The main assumption of the experiment was that stochastic gradient-free optimization methods are effective approaches for updraft center detection. The following step-by-step summary shows how SPSA iteratively produces a sequence of updraft center estimates.

Algorithm:

1. *Initialization and coefficient selection.* Set a counter index $n = 0$. Select an initial guess $\widehat{\mathbf{x}}(0) \in \mathbb{R}^2$ and a fairly small non-negative coefficient $\alpha > 0$. The initial guess in our implementation of the algorithm is the point where a positive updraft was first measured.
2. *Iteration* $n \to n + 1$. Set $n := n + 1$.
3. *Generation of the simultaneous perturbation vector.* Use Monte Carlo to generate a 2-dimensional random perturbation vector $\mathbf{\Delta}_n$ whose components are independently generated from a zero mean probability distribution satisfying the preceding conditions. A common choice for each component of $\mathbf{\Delta}_n$ is to use a Bernoulli ± 1 distribution with probability of $\frac{1}{2}$ for each ± 1 outcome.
4. *Proceeding to the new waypoints.* Proceed to next two points $\mathbf{u}_n^-$ and $\mathbf{u}_n^+$. They are intersections of the UAV trajectory projection on 2D plain and the line which goes through the point of the previous estimate $\widehat{\mathbf{x}}(n-1)$ in the direction of the vector $\mathbf{\Delta}_n$ (see Figure 3.6).
5. *Velocity function evaluations.* Obtain two measurements at the points $\mathbf{u}_n^{\pm}$ of the velocity function
$$y_n^{\pm} = F(\mathbf{u}_n^{\pm}).$$
6. *Computing the values* $\beta_n^{\pm}$. Measure the distances from the point of the previous estimate $\widehat{\mathbf{x}}(n-1)$ and points $\mathbf{u}_n^-$ and $\mathbf{u}_n^+$ and compute two values $\beta_n^{\pm}$ such that

$$\mathbf{u}_n^{\pm} = \widehat{\mathbf{x}}(n-1) + \beta_n^{\pm}\boldsymbol{\Delta}_n.$$

7. *Quasigradient calculation.* Calculate the quasigradient:

$$\widehat{g} = \boldsymbol{\Delta}_n \frac{y_n^+ - y_n^-}{\beta_n^+ + \beta_n^-}.$$

8. *Updating center estimation.* Use the standard stochastic approximation form

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) + \alpha\widehat{g}$$

to update the current center estimation.

9. *Iteration or termination.* Return to Step 2 or terminate the algorithm if there is little change in the estimations obtained on several successive iterations or if the maximum allowed number of iterations has been reached.
10. *Climbing in the updraft.* Circle around the estimated updraft center in order to climb.

This method yields good approximation of the updraft center using a small number of measurements and no a priori knowledge about updraft location.

Simulation Results. For this project, we chose to use a thermal updraft model developed by Allen at NASA Dryden [8] for a similar autonomous UAV

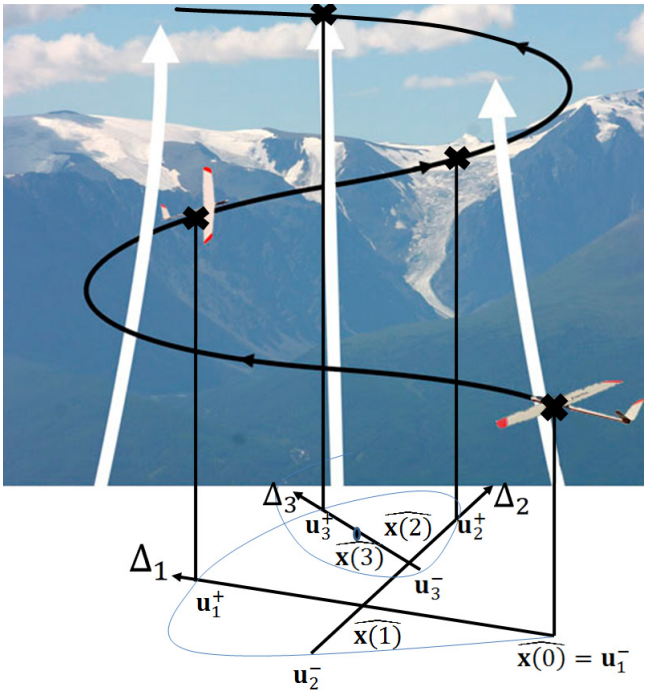

Fig. 3.6. The sequence of estimates and waypoints

soaring project. This model was developed using atmospheric data collected by NOAA in Nevada using rawinsonde balloons released every 12 hours over the course of a year.

We used Allen's model to create a dynamic field of thermal updrafts within the specified test area in which the UAV was constrained to fly. Updraft positions were randomly chosen and held for 20 min at a time. The 20-min thermal lifespan was chosen from personal observation of cumulus clouds. As updrafts have a finite lifetime, the value function estimate of a particular visited cell becomes less and less accurate depending on how recently the UAV has visited it. To reflect this increasing uncertainty over time, a discount factor 0.95 was applied at each time step such that the estimated velocity of each cell gradually decayed toward zero. In our experiment the UAV flies through randomly chosen waypoints. As the airplane flies it tries to detect the updrafts its path coincides with. When the vehicle finds any points with positive vertical velocity, and provided its altitude is less than a preset threshold, it starts the center detection algorithm to gain altitude using the upward directed vertical wind.

Table 3.1. Results of simulations

Average velocity value in the point where SPSA was initiated	Average velocity in the updraft center as found by SPSA
0.55	1.9
0.74	2.65
0.55	2.29
0.65	2.43

The results of the experiment are summarized in Table 3.1. In order to obtain these statistics we performed 100 experiments with SPSA updraft center detection using only 10 measurements. The vertical velocity varied between -0.12 and 2.75 over the test field.

4

Linear Models

4.1 Linear Regression and Filtering under Arbitrary External Noise

This section is concerned with parameter estimation and the filtering of linear models. We focus on algorithms which facilitate avoiding standard requirements for observation noise.

4.1.1 Linear Regression

In the case of non-centered correlated noise and even nonrandom noise, the parameters of linear regression (LR) can be estimated efficiently, although at first glance this seems surprising. This estimation can be accomplished under certain conditions when inputs (regressors) $\mathbf{u}_n$ are random and available. Moreover, the optimal algorithms of LR parameter estimation have the same rates of convergence as do those in the "standard" case.

Consider the linear regression model

$$y_n \;=\; \mathbf{u}_n^{\mathrm{T}}\,\mathbf{x}_n \;+\; v_n, \quad \mathbf{x}_n = \mathbf{x}^\star + \mathbf{w}_n\,, \quad n = 1, 2, \ldots, \tag{4.1}$$

where $y_n \in \mathrm{R}^1$ is an observation output made at time instant n, $\mathbf{u}_n \in \mathrm{R}^d$ is a random input vector which is available at time n, $v_n \in \mathrm{R}^1$ represents observation noise and $\mathbf{w}_n \in \mathrm{R}^d$ is a random disturbance.

The goal *is to find a sequence of estimates* $\{\widehat{\mathbf{x}}(n)\}$ that converges to the vector $\mathbf{x}^\star$ of unknown parameters. Each estimate $\widehat{\mathbf{x}}(n)$ should be based on the observations $y_i,\ \mathbf{u}_i,\ i \le n$.

Let $\mathcal{F}_n$ be the σ-algebra of probabilistic events generated by random vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$, $\mathbf{w}_1, \ldots, \mathbf{w}_n$, and values $v_1, v_2, \ldots, v_n$ if they are random.

We make the following assumptions.

As4.1: The inputs $\{\mathbf{u}_n\}$ form a sequence of random vectors with known bounded expectations $\|E\mathbf{u}_n\| \le M_{\mathbf{u}} < \infty$.

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining,* Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_4,

As4.2: For each n the centered random vector $\boldsymbol{\Delta}_n = \mathbf{u}_n - E\mathbf{u}_n$ has a symmetric distribution function $\mathrm{P}_n(\cdot)$ (i.e. $\mathrm{P}_n(\Omega) = \mathrm{P}_n(-\Omega)$ for any Borel set $\Omega \subset \mathbb{R}^d$), and

$$E\boldsymbol{\Delta}_n\boldsymbol{\Delta}_n^{\mathrm{T}} = \mathbf{B}_n > 0, \|\mathbf{B}_n\| \le \sigma_{\boldsymbol{\Delta}}^2 < \infty.$$

The vector $\boldsymbol{\Delta}_n$ is independent of $\mathcal{F}_{n-1}$.

As4.2': Assumption *As4.2* holds and

$$\mathbf{B}_n = \mathbf{B} > 0, \;\; E\|\boldsymbol{\Delta}_n\|^4 \le M_4 < \infty.$$

As4.3: $\forall n$ $E\mathbf{w}_n = 0$, $\mathbf{w}_n$ is independent of $\mathcal{F}_{n-1}$ and $\boldsymbol{\Delta}_n$.
As4.4: $E\|\mathbf{w}_n\|^2 \le \sigma_{\mathbf{w}}^2 < \infty$.
As4.5: $E\mathbf{w}_n\mathbf{w}_n^{\mathrm{T}} \le \mathbf{Q}_{\mathbf{w}} < \infty$.
As4.6: $\forall n$ if v_n is random then $\mathbf{w}_n$ and $\boldsymbol{\Delta}_n$ do not depend on it.
As4.7: $E_{\mathcal{F}_{n-1}} v_n^2 \le \sigma_v^2 < \infty$, a.s.
As4.8: $Ev_n^2 \le \sigma_v^2 < \infty$.
Here σ_v, $\sigma_{\mathbf{w}}$ are constants, $\mathbf{Q}_{\mathbf{w}}$ is a symmetric matrix.

Note, that in the LR parameters estimation problem with random inputs $\{\mathbf{u}_n\}$ the standard requirements on the observation noise $\{v_n\}$ are somewhat different (see, e.g. [265]). In particular, it is usually assumed that $\{v_n\}$ is a sequence of identically distributed random variables with zero-mean and that v_n are independent of each other and of $\mathbf{u}_n$.

We first examine a randomized algorithm of the stochastic approximation type (*randomized stochastic approximation (RSA) algorithm*) for the observation model (4.1)

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \alpha_n\boldsymbol{\Gamma}\boldsymbol{\Delta}_n(\mathbf{u}_n^{\mathrm{T}}\widehat{\mathbf{x}}(n-1) - y_n), \quad n = 1, 2, \ldots, \tag{4.2}$$

where $\alpha_n \ge 0$ is a nonrandom step–size and $\boldsymbol{\Gamma}$ is a positive definite symmetric matrix. We assume that the initial estimate $\widehat{\mathbf{x}}(0)$ is an arbitrary nonrandom vector from $\mathbb{R}^d$.

Theorem 4.1. *Let Assumptions As4.1–4.3 and As4.6 be satisfied for the inputs and noises of model* (4.1)*, and*

$$\alpha_n(1 + E\|\boldsymbol{\Delta}_n\|^4) \to 0 \quad \text{as} \quad n \to \infty, \quad \sum_{n=1}^{\infty} \alpha_n = \infty. \tag{4.3}$$

If *Assumptions As4.4 and As4.7 hold for the model* (4.1) *noises and*

$$\sum\nolimits_{n=1}^{\infty} \alpha_n^2(1 + E\|\boldsymbol{\Delta}_n\|^4) < \infty, \tag{4.4}$$

then *for estimates generated by the algorithm* (4.2) *we have* $\widehat{\mathbf{x}}(n) \to \mathbf{x}^\star$ *a.s. as* $n \to \infty$.

If *Assumptions As4.5 and As4.8 hold,* **then** $E(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)^{\mathrm{T}} \to 0$ *as* $n \to \infty$.

The proof of Theorem 4.1 is given in [137].

Note that for the case $\mathbf{B}_n = \mathbf{B} > 0$, the first statement follows directly from the second part of Theorem 3.1 from Section 3.3. Algorithm (4.2) with $\mathbf{K}_n(\mathbf{\Delta}_n) = \mathbf{\Delta}_n$ coincides with algorithm (3.10) from Section 3.2 for the function

$$f(\mathbf{x}, \mathbf{w}') = \frac{1}{2}(\mathbf{x} - \mathbf{x}^\star + \mathbf{w}')^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^\star + \mathbf{w}') \tag{4.5}$$

when $\mathbf{\Gamma} = \mathbf{I}$ and α_n from algorithm (4.2) corresponds to the $\alpha_n(\beta_n^+ + \beta_n^-)$ from algorithm (3.10).

The following theorem establishes the rate of convergence of the sequence of estimates generated by algorithm (4.2).

Theorem 4.2. *Let Assumptions As4.1–4.3,4.5 and As4.6,4.8 be fulfilled and* $\alpha_n = n^{-1}$, $En^{-1}\|\mathbf{\Delta_n}^4\| \to 0$ *as* $n \to \infty$.

If *there are matrices* $\mathbf{B} > 0$ *and* $\mathbf{U}$ *such that and* $-\mathbf{\Gamma B} + \frac{1}{2}\mathbf{I}$ *be a Hurwitz matrix (i.e., all its eigenvalues lie in the left half–plane),*

$$\|\mathbf{B}_n - \mathbf{B}\| = \mathcal{O}(n^{-1}), \;\; E\mathbf{\Delta}_n\mathbf{\Delta}_n^{\mathrm{T}}\mathbf{Q_w}\mathbf{\Delta}_n\mathbf{\Delta}_n^{\mathrm{T}} \le \mathbf{U} + \mathcal{O}(n^{-1}), \;\; \|\mathbf{U}\| < \infty,$$

then *for the estimates of algorithm* (4.2) *we have*

$$E(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)^{\mathrm{T}} \le n^{-1}\,\mathbf{S} \;+\; o(n^{-1}), \tag{4.6}$$

where the matrix $\mathbf{S}$ *is the solution of the matrix equation*

$$\mathbf{\Gamma}\,\mathbf{B}\,\mathbf{S} \;+\; \mathbf{S}\,\mathbf{B}\,\mathbf{\Gamma} \;-\; \mathbf{S} \;=\; \mathbf{\Gamma R \Gamma}, \tag{4.7}$$

where $\mathbf{R} = (\sigma_v^2(1 + \nu M_{\mathbf{u}}^2) + M_{\mathbf{u}}^2 \mathrm{Tr}[\mathbf{Q_w}])\mathbf{B} + \mathbf{U}$ *with any* $\nu > 0$.

The proof of Theorem 4.2 is also given in [137].

If $\mathit{\Gamma} = \mathbf{B}^{-1}$, $\mathrm{Tr}[\mathbf{Q_w}] = 0$, then equation (4.7) for matrix $\mathbf{S}$ can be explicitly solved: $\mathbf{S} = \mathbf{B}^{-1}\mathbf{R}\mathbf{B}^{-1}$. For algorithm (4.2) which assumes the form

$$\widehat{\mathbf{x}}(n) \;=\; \widehat{\mathbf{x}}(n-1) \;-\; (n\mathbf{B})^{-1}(\mathbf{u}_n - M_{\mathbf{u}})(\mathbf{u}_n^{\mathrm{T}}\widehat{\mathbf{x}}(n-1) - y_n), \tag{4.8}$$

we now obtain

$$E(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)^{\mathrm{T}}\} \le n^{-1}\sigma_v^2(1 + \nu M_{\mathbf{u}}^2)\mathbf{B}^{-1} \;+\; o(n^{-1}) \tag{4.9}$$

with any $\nu > 0$.

For the last algorithm (4.8), we obtained almost the same convergence rate as the best possible rate in the case where the noise v_n is an independent random variable with zero-mean, see [265]. Moreover, this choice of α_n and $\mathbf{\Gamma}$ has been shown by Polyak and Tsypkin [265] to be optimal for similar kind algorithms.

Remark: If Assumptions *As4.5* and *As4.8* in Theorem 4.2 are equalities, the inequality in the assessment of the convergence rate can also be replaced by an equality.

Unfortunately, the optimal RSA algorithm (4.8) is not applicable if matrix $\mathbf{B}$ is a priori unknown. Consider a randomized algorithm of stochastic approximation with averaging:

$$\begin{cases} \widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) \; - \; \alpha_n \boldsymbol{\Delta}_n (\mathbf{u}_n^{\mathrm{T}} \widehat{\mathbf{x}}(n-1) - y_n), \\ \\ \tilde{\mathbf{x}}(n) = (1 - n^{-1}) \tilde{\mathbf{x}}(n-1) \; + \; n^{-1} \widehat{\mathbf{x}}(n-1) \left(= \; n^{-1} \; \sum_{i=0}^{n-1} \widehat{\mathbf{x}}(i) \right), \end{cases} \tag{4.10}$$

which is similar to the one in [259] for the case of noise with zero mean.

For the sake of simplicity in the following statements we will consider a stronger Assumption *As4.2'* about the inputs.

Theorem 4.3. *Let Assumptions As4.1,4.2' and As4.6,4.8 be fulfilled for the inputs* $\{\mathbf{u}_n\}$ *and for the observation noise* $\{v_n\}$,
$\mathbf{w}_n = 0$,
$\alpha_n \to 0$ *and* $\alpha_n / \alpha_{n+1} \; = \; 1 \; + \; o(\alpha_n)$ *as* $n \to \infty$.

Then *the inequality* (4.9) *holds with any* $\nu > 0$ *for the mean square convergence rate of the algorithm* (4.10) *estimates.*

Proof of Theorem 4.3 is given in [148].

As above, the obtained upper bound (4.9) for the convergence rate is almost the same as the previously known upper bound in the case of independent noise with zero mean (see [259]). But this algorithm (4.10) is simpler than the optimal algorithm (4.8), and its asymptotic convergence rate is independent of the choice of α_n, satisfying only the condition $\alpha_n / \alpha_{n+1} \; = \; 1 \; + \; o(\alpha_n)$. Note that the last relation holds for $\alpha_n = \alpha \, n^{-\gamma}$, $0 < \gamma < 1$ but not for $\alpha_n = \alpha \, n^{-1}$.

Remark. Theorems 4.1–4.3 also hold assuming the zero of the third central moment $\mathbf{u}_n$ instead of symmetry in the distribution of $\boldsymbol{\Delta}_n$.

For the same regression model of observations (4.1), we now consider the estimates generated by the following randomized least squares method:

$$\begin{cases} \widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \boldsymbol{\Gamma}_n \boldsymbol{\Delta}_n (\mathbf{u}_n^{\mathrm{T}} \widehat{\mathbf{x}}(n-1) - y_n), \\ \\ \boldsymbol{\Gamma}_n = \boldsymbol{\Gamma}_{n-1} - \boldsymbol{\Gamma}_{n-1} \boldsymbol{\Delta}_n \boldsymbol{\Delta}_n^{\mathrm{T}} \boldsymbol{\Gamma}_{n-1} / (1 + \boldsymbol{\Delta}_n^{\mathrm{T}} \boldsymbol{\Gamma}_{n-1} \boldsymbol{\Delta}_n), \; \boldsymbol{\Gamma}_0 = \gamma_0^{-1} \mathbf{I}, \end{cases} \tag{4.11}$$

where $\gamma_0 > 0$ is some regularization parameter, see [215, 355]. We again assume that the initial estimate $\widehat{\mathbf{x}}(0)$ is an arbitrary nonrandom vector from $\mathbb{R}^d$.

Theorem 4.4. *Let Assumption As4.1,4.2' be fulfilled.*

If *Assumptions As4.3,4.4 and As4.6,4.7 are satisfied*

then *for the estimates generated by the algorithm* (4.11) *we have* $\widehat{\mathbf{x}}(n) \to \mathbf{x}^\star$ *a.s. as* $n \to \infty$.

If $|v_n| \; \le \; c_v, \;\; \|\mathbf{w}_n\| \; \le \; c_{\mathbf{w}}, \;\; \|\boldsymbol{\Delta}_n\| \; \le \; c_{\boldsymbol{\Delta}}$ *a.s. with some constants* c_v, $c_{\mathbf{w}}$, $c_{\boldsymbol{\Delta}} < \infty$

then *for the algorithm* (4.11) $E(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)(\widehat{\mathbf{x}}(n) - \mathbf{x}^\star)^{\mathrm{T}} \} \to 0 \quad \text{as} \quad n \to \infty$.

Proof of Theorem 4.4 is given in [141].

4.1.2 Application in Photoemission Experiments

The example of the application of RSA algorithm (4.2) to filtering systematic noise (SN) with non-zero mean value in photoemission data is considered in [108]. Authors analyze a series of 50 single-scan photoemission (PE) spectra of $W(110)$ surface where SN of unknown origin appears at some particular kinetic energies of photoelectrons. Such SN may introduce additional spectral features that cannot be expected from the electronic structure of sample. If this noise is not zero-mean, it cannot be eliminated by simply increasing the number of scans in PE experiment. It was found that the RSA-evaluated spectrum is in good agreement with the spectrum measured without SN. Based on these results, we anticipate wide application of RSA for evaluation of experimental data.

In the considered study photoemission spectra were taken from a $W(110)$ single crystal kept at room temperature. Experiments were conducted in the setup based on the hemispherical energy analyzer (SPECS PHOIBOS 150, www.specs.de). The overall system energy resolution accounting for the thermal broadening was set to 150 meV, and electrons were collected in angle-integrated mode around the surface normal. The base pressure was in the range of 1×10^{-10} mbar. Prior to the experiment, the $W(110)$ crystal was carefully cleaned by repeated cycles of heating up to 1300° C at oxygen ambient pressure of 5×10^{-8} mbar for 15 min each and subsequent flashing up to 2300° C. After this procedure the crystal was kept in a vacuum for 24 hours in order to passivate the surface of the crystal via absorption of residual gases in the experimental chamber. This step was necessary in order to stabilize the crystal surface for the long-term surface-sensitive PE experiment. As excitation light sources we used $He\,II\alpha$ resonance line ($h\nu = 40.8$ eV) and $Al\,K\alpha$ emission line ($h\nu = 1486.6$ eV) in order to generate studied PE signal (objective function) and independent non-zero mean noise, respectively.

The scheme of the experiment is shown in Figure 4.1(a). PE spectra were collected in the range of $25.8 - 37.8$ eV kinetic energies of emitted photoelectrons. In this case, the He I la radiation of fixed intensity was used each time to produce a single-scan PE spectrum of the valence band of W(110) surface (Figure 4.2, open circles). Different single-scan spectra were then excited with different $He\,II\alpha$ radiation intensities, which were randomly selected. The photocurrent emitted in this process can be expressed as

$$j(E_{kin}) = I \cdot DOS(E_{kin}) \cdot \frac{d\sigma(E_{kin})}{d\Omega},$$

where I is the intensity of the light source ($He\,II\alpha$), DOS denotes the electronic density of states of the $W(110)$ surface, $d\sigma(E_{kin})/d\Omega$ is the cross-section of the photoemission process, and $E_{kin} = h\nu - W - E_B$ stands for the kinetic energy of the photoelectron (W-work function of the material, E_B-binding energy of the electron in the solid). Since $d\sigma(E_{kin})/d\Omega$ is practically constant in the small energy range, the total photocurrent can be written as

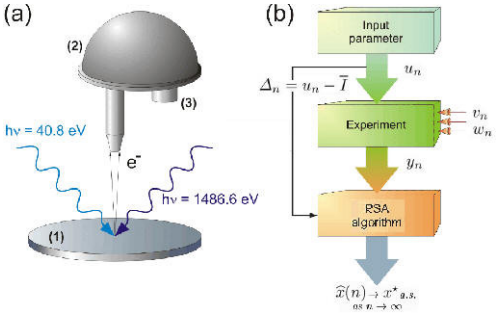

Fig. 4.1. (a) Scheme of the PE experiment, where sample (1) is illuminated by two light sources, $He\,II\alpha$ and $Al\,K\alpha$, electrons are analyzed by photoelectron spectrometer (2) and detector (3) registers the signal y_n. (b) Layout of the present study using the RSA (4.2) algorithm for eliminating noises of unknown nature [108].

$$j(E_{kin}) = I \cdot const \cdot DOS(E_{kin}).$$

It is proportional to density of states and to intensity of incoming radiation.

The problem is to study $x = const \cdot DOS(E_{kin})$ based on the observation $y = j(E_{kin})$ and controllable parameter $u = I$.

In every single-scan experiment, the systematic noise $v = N(E_{kin})$ was introduced by switching on the x-ray source when measuring in the range of $28.8 - 35.9$ eV kinetic energies (shaded area at the bottom of Figure 4.2). The resulting single-scan spectrum in this energy region is shown by the filled circles in Figure 4.2. Thus the generated noise represents secondary electrons in the x-ray spectrum of the $W(110)$ surface. We emphasize that this source of electrons is independent of the first source caused by the $He\,II\alpha$ radiation.

We can now rewrite the total photocurrent in the following form:

$$y = j(E_{kin}) = I \cdot const \cdot DOS(E_{kin}) + N(E_{kin}) = u \cdot x + v.$$

For a series of measurements, the previous expression can be rewritten as

$$y_n = u_n \cdot x_n + v_n, \quad n = 1, 2, \ldots, \tag{4.12}$$

where n is an iteration number. Observable (strictly measured) variables y_n and u_n correspond to $j(E_{kin})$ and I, while the unobservable investigated variable x_n and the systematic noise v_n correspond to $const \cdot DOS(E_{kin})$ and $N(E_{kin})$, respectively. Depending on even slight changes in the experimental conditions (such as temperature drift) from one measurement to another, the investigated $const \cdot DOS(E_{kin})$ parameter can randomly vary within a certain distribution function around a fixed mean value $x^\star$.

The *goal* of the entire measurement process *is to probe* this mean value $x^\star$.

For each experiment we can consider unknown x_n values to be randomly distributed:

$$x_n = x^\star + w_n, \quad n = 1, 2, \ldots,$$

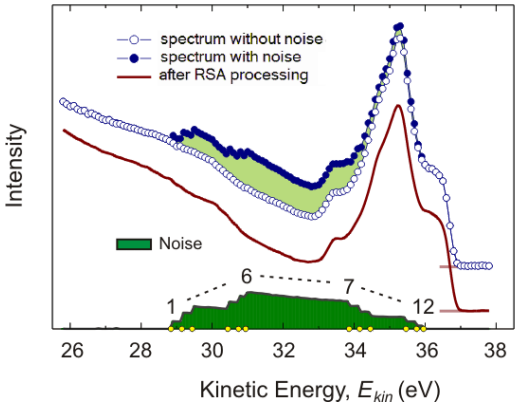

Fig. 4.2. Experimental PE spectra of the valence band of $W(110)$ obtained with $He\,II\alpha$ radiation without (open circles) and with (filled circles) systematic noise. Spectrum obtained after application of the RSA algorithm (4.13) to a series of 50 experimental single-scan spectra is shown by a thin line. The shaded area at the bottom is systematic noise measured separately. Twelve control points used for demonstration of the convergence dynamic of the algorithm are marked with labels (1-12) on the kinetic energy axis [108].

where $x^\star$ is the investigated mean value and w_n is a random zero-mean disturbance.

If v_n from (4.12) represents a systematic noise with a non-zero mean value, conventionally used algorithms fail to correctly estimate $x^\star$. To account for the non-zero mean contribution of SN to the estimated mean value of the investigated parameter, randomly distributed u_n were generated in the present experiment (see above).

Let us assume the observable variable (input) u_n is randomly distributed around its known mean value $\bar{I}$. Note, there is no dependence between u_n and v_n that is of high importance for the consistency of the following algorithm. This independence conditions is always true if u_n is selected to be random, while v_n is an unknown but bounded deterministic function. The main issue of algorithm (4.2) suggested in Subsection 4.1.1 is to the estimate mean value $\overline{x}$ of $[const \cdot DOS(E_{kin})]$ measuring y_n (experimental spectra) for randomly distributed u_n (intensity of $He\,II\alpha$ radiation).

Intuitively, the underlying mathematics can be understood on the basis of (4.12), which can be partially differentiated with respect to variable u_n. In a case where u_n and v_n are independent of each other, a partial derivative of y_n is equal to x_n.

Based on the above, the task of eliminating the systematic noise is reformulated to construct a sequence of estimations of the average value of the parameter of the linear regression model, as in Subsection 4.1.1. It was

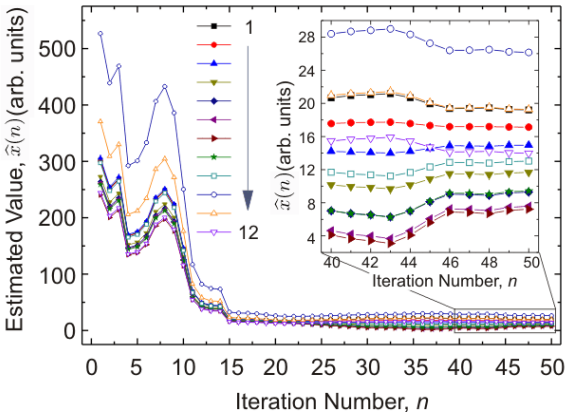

Fig. 4.3. Convergence of the RSA algorithm (4.13): The estimated values [$const \cdot DOS(E_{kin})$] of the twelve control points marked in Figure 4.2 [108]

shown that in this case one can effectively use a RSA type algorithm (4.2) or randomized least-square method (4.11).

In order to evaluate the experimental data we applied a modification of the RSA algorithm (4.8) (see Figure 4.1(b) for the scheme of RSA application). Everywhere in the following we use $\widehat{x}(n)$ instead of x_n to emphasize that the estimated, not real, values of the investigated variable are considered

$$\widehat{x}(n) = \widehat{x}(n-1) - \frac{\Delta_n}{\sigma_n^2 n}(u_n \widehat{x}(n-1) - y_n),\ n = 1, 2, \ldots, \widehat{x}(0) = 0, \quad (4.13)$$

where $\Delta_n = u_n - \overline{I}$ and σ_n^2 is positive bounded dispersion of Δ_n.

By virtue of Theorem 4.1, the estimation sequence $\{\widehat{x}(n)\}$ leads asymptotically to the real value of $x^\star$ under general conditions mentioned above. In the experiment, however, we can follow only a finite number of measurements. Supposing that after some measurements the sequence of $\{\widehat{x}(n)\}$ becomes stable, we can assume with high probability that this estimation value is in good agreement with the real one.

Figure 4.3 shows the evolution of the estimated values $\widehat{\mathbf{x}}(n)$ which must be proportional to the DOS value. In order to prove the RSA algorithm (4.13) in [108] authors chose twelve control points in PE spectra at particular kinetic energies (points are marked on the kinetic energy axis in Figure 4.2). The intensities at these points in PE spectra during step-by-step application of the algorithm are plotted in Figure 4.3 with corresponding zoom for the last 10 steps (see inset). In spite of the overestimated large initial "guess" values, the algorithm process is almost stabilized at around the 20-th iteration. The most stable estimations occur from the 46-th iteration. Figure 4.2 shows the result of the application of fifty steps of the RSA algorithm (bottom spectrum). The

evaluated spectrum is in good agreement with the one directly measured in the experiment, when the x-ray source was not activated (no SN noise).

4.1.3 Moving Average

Let the observation model be described by the equation

$$y_t = \sum_{i=1}^{d} u_{t-i} x_i^\star + v_t,\ t = 1, 2, \ldots, \tag{4.14}$$

where $y_t \in \mathbb{R}^1$ are outputs (observations), $v_t \in \mathbb{R}^1$ are observation noises, $u_t \in \mathbb{R}^1$ are inputs, $x_1^\star, x_2^\star, \ldots, x_d^\star \in \mathbb{R}^1$ are unknown parameters of the model (4.14).

Let $r \geq d$ be a natural: $r \in \mathbb{N}$. By the virtue of the observation model (4.14) for any $i \in 1..d$ we have

$$y_{rn+i} \ = \ u_{rn}\, x_i^\star + \bar{v}_{i,n},\ n = 1, 2, \ldots. \tag{4.15}$$

where

$$\bar{v}_{i,n} \ = \ u_{rn+i-1}x_1^\star + \ldots + u_{rn+1}x_{i-1}^\star + u_{rn-1}x_{i+1}^\star + \ldots + u_{rn+i-d}x_d^\star + v_{rn+i}.$$

Assume that inputs $\{u_t\}$ satisfy the following condition

As4.9: $\{u_t\}$ is an independent sequence, Assumptions *As4.1* and *As4.2* hold for the inputs $\{u_{rn}\}_{n=1,2,\ldots}$, and for any n the input u_{rn} does not depend on noises $v_1, v_2, \ldots, v_{rn+d-1}$ if they are random.

Note, for any $n = 1, 2, \ldots$ the input u_{rn} does not depend on $\bar{\mathbf{v}}_1, \ \ldots, \bar{\mathbf{v}}_n$ where $\bar{\mathbf{v}}_j = (\bar{v}_{1,j}, \bar{v}_{2,j}, \ldots, \bar{v}_{d,j})^{\mathrm{T}},\ j \in 0..n$, when Assumption *As4.9* holds. If Assumptions *As4.6* and *As4.8* also hold then

$$\forall n, i \ \ E\bar{v}_{i,n}^2 \leq \sigma_{\bar{v}}^2 = d\left(\sigma_v^2 + (\sigma_\Delta^2 + M_u^2)\|\mathbf{x}^\star\|^2\right) < \infty$$

where $\mathbf{x}^\star = (x_1^\star, x_2^\star, \ldots, x_d^\star)^{\mathrm{T}}$. Therefore, to estimate the vector $\mathbf{x}^\star$, we can use any of the three algorithms: (4.2), (4.10) or (4.11).

By the arbitrariness of $i \in 1..d$, to estimate all unknown parameters of the moving average model (4.14), we can use d parallel randomized algorithms which are based on any of the previously proposed for estimating the parameters of linear regression.

RSA algorithm (4.2) to estimate vector $\mathbf{x}^\star$ which consists of all unknown parameters of the moving average model (4.14) takes the form

$$\widehat{\mathbf{x}}(n) = \widehat{\mathbf{x}}(n-1) - \alpha_n \gamma \boldsymbol{\Delta}_n (u_{rn}\widehat{\mathbf{x}}(n-1) - \mathbf{y}_n), \quad n = 1, 2, \ldots, \tag{4.16}$$

where

$$\Delta_n = u_{rn} - Eu_{rn}, \quad \mathbf{y}_n = \begin{pmatrix} y_{rn} \\ y_{rn+1} \\ \vdots \\ y_{rn+d-1} \end{pmatrix} \in \mathbb{R}^d,$$

$\gamma > 0$ is a constant.

The next theorem follows immediately from Theorems 4.1 and 4.2.

Theorem 4.5. *Let Assumption As4.9 and conditions* (4.3) *be satisfied.*
If *Assumption As4.7 and condition* (4.4) *hold*
then *for estimates generated by the algorithm* (4.16) *we have* $\widehat{\mathbf{x}}(n) \to \mathbf{x}^\star$ *a.s. as* $n \to \infty$.
If *Assumption As4.8 holds*
then $E\|\widehat{\mathbf{x}}(n) - \mathbf{x}^\star\|^2 \to 0$ *as* $n \to \infty$.
If *additionally* $\alpha_n = n^{-1}$, $En^{-1}\Delta_n^4 \to 0$ *as* $n \to \infty$, $|E\Delta_n^2 - \sigma_\Delta^2| = \mathcal{O}(n^{-1})$

$$2\gamma\sigma_\Delta^2 > 1$$

then *for the components of estimates of the algorithm* (4.16) *we have*

$$E(\widehat{x}_i(n) - x_i{}^\star)^2 \le n^{-1}\,\frac{\gamma^2\sigma_\Delta^2\sigma_v^2}{2\gamma\sigma_\Delta^2 - 1}(1 + \nu M_u^2) \;+\; o(n^{-1}), \;\; i \in 1..d, \tag{4.17}$$

with any $\nu > 0$.

4.1.4 Filtering

We will confine the discussion to considering the following problem statement: a scalar signal is observed that satisfies the equation

$$y_t = \mathbf{u}_t^{\mathrm{T}}\mathbf{x}_t + v_t, \tag{4.18}$$

which is a mixture of the transformed vector process $\{\mathbf{x}_t\}$, $\mathbf{x}_t \in \mathbb{R}^d$, and the observation noise $\{v_t\}$. Here, $\{\mathbf{u}_t\}$ are d-vectors which are known at the time instant t.

The vector process $\{\mathbf{x}_t\}$ is generated by a linear filter

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_{t+1}, \tag{4.19}$$

in which $\mathbf{A}$ is the known matrix:$\|\mathbf{A}\| \le 1$, and $\{\mathbf{w}_t\}$ is a realization of a sequence of zero-mean independent random vectors.

The problem of *filtering with one-step prediction* consists of designing the estimate $\widehat{\mathbf{x}}(t+1)$ of the value of the process $\{\mathbf{x}_t\}$ at the time instant $t+1$ based on the observations $y_k, \mathbf{u}_k,\ k \le t$. The quality of filtering is determined by the mean value of the squared prediction errors

$$E \parallel \widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1} \parallel^2 .$$

It is usually supposed that the vectors $\{\mathbf{u}_t\}$ in the observation model are specified by a deterministic sequence. Here, we assume that the sequence of the vectors $\{\mathbf{u}_t\}$ is random and satisfies condition *As4.1,4.2'*. Under these assumptions procedure (4.18) of the measurement of the process $\mathbf{x}_t$ is, in fact, randomized since the sought signal is "weighed" with a randomly chosen set of coefficients $\mathbf{u}_t$ whose values are known at a current moment.

Consider the following randomized algorithm for the next estimate generation:

$$\widehat{\mathbf{x}}(t+1) = \mathbf{A}\widehat{\mathbf{x}}(t) - \alpha\mathbf{A}\mathbf{\Gamma}\mathbf{\Delta}_t(\mathbf{u}_t^{\mathrm{T}}\widehat{\mathbf{x}}(t) - y_t),\ \mathbf{\Delta}_t = \mathbf{u}_t - \bar{\mathbf{u}}_t,\ \bar{\mathbf{u}}_t = E\mathbf{u}_t,\ (4.20)$$

where $t = 0, 1, \ldots,\ \alpha > 0$ is the step-size value (gain coefficient) and $\mathbf{\Gamma}$ is a positive-definite symmetrical matrix.

The initial guess $\widehat{\mathbf{x}}(0)$ is assumed to be given by an arbitrary nonrandom vector from $\mathbb{R}^d$. Algorithm (4.20) is called randomized because the current measurement is carried out with a randomized input (a set of weights), and the current estimate is changed in a randomized direction $\mathbf{A}\mathbf{\Gamma}\mathbf{\Delta}_t$.

Substituting (4.18) and (4.19) into (4.20), for the prediction error we get

$$\widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1} = \mathbf{A}(\mathbf{I} - \alpha\mathbf{\Gamma}\mathbf{\Delta}_t\mathbf{\Delta}_t^{\mathrm{T}})(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - \alpha\mathbf{A}\mathbf{\Gamma}\mathbf{\Delta}_t(\bar{\mathbf{u}}_t^{\mathrm{T}}(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - v_t) - \mathbf{w}_{t+1}.$$

Let us denote $D_t := \|\widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1}\|^2$. If Assumption *As4.3,4.5,4.6,4.8* hold then averaging conditionally with respect to the prehistory of all random processes up to the time instant t except $\mathbf{w}_t$ we deduce that $E_{\mathcal{F}_{t-1},\mathbf{\Delta}_t} D_t =$

$$= E_{\mathcal{F}_{t-1},\mathbf{\Delta}_t}\|\mathbf{A}(\mathbf{I} - \alpha\mathbf{\Gamma}\mathbf{\Delta}_t\mathbf{\Delta}_t^{\mathrm{T}})(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - \alpha\mathbf{A}\mathbf{\Gamma}\mathbf{\Delta}_t(\bar{\mathbf{u}}_t^{\mathrm{T}}(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - v_t)\|^2 + d\sigma_{\mathbf{w}}^2.$$

Hence, taking the conditional expectation over $\mathcal{F}_{t-1}$ by virtue of assumption *As4.2'* it may be concluded that

$$E_{\mathcal{F}_{t-1}} D_t \le (1 - 2\alpha\lambda_{\min}(\mathbf{B}\mathbf{\Gamma}) + \alpha^2\|\mathbf{\Gamma}\|^2 M_4)\|\mathbf{A}\|^2 D_{t-1} +$$
$$+ \alpha^2\|\mathbf{A}\mathbf{\Gamma}\|^2 \mathrm{Tr}[\mathbf{B}] E_{\mathcal{F}_{t-1}}(\bar{\mathbf{u}}_t^{\mathrm{T}}(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - v_t)^2 + d\sigma_{\mathbf{w}}^2,$$

since the symmetry of the distribution for $\mathrm{P}(\cdot)$ leads to

$$(\widehat{\mathbf{x}}(t) - \mathbf{x}_t)^{\mathrm{T}} E_{\mathcal{F}_{t-1}}(\mathrm{I} - \alpha\mathbf{\Delta}_t\mathbf{\Delta}_t^{\mathrm{T}}\mathit{\Gamma})\mathrm{A}^{\mathrm{T}}\alpha\mathrm{A}\mathit{\Gamma}\mathbf{\Delta}_t(E_{\mathcal{F}_{t-1}}\{\bar{\mathbf{u}}_t^{\mathrm{T}}(\widehat{\mathbf{x}}(t) - \mathbf{x}_t) - v_t) = 0.$$

Further, after taking unconditional mathematical expectation from the both parts of the last formula, using assumptions *As4.3,4.5,4.6,4.8* and satisfying the inequality below for any $\nu > 0$

$$2E\mathbf{u}_t^{\mathrm{T}}(\widehat{\mathbf{x}}(t) - \mathbf{x}_t)v_t \ \le\ \nu M_{\mathbf{u}} v_t^2 + \frac{M_{\mathbf{u}}}{\nu} D_{t-1},$$

for the mean value of the prediction error, we derive the estimate

$$ED_t \ \le\ \psi(\alpha, \rho)ED_{t-1} \ +\ \alpha^2(1 + M_{\mathbf{u}}\nu)\|\mathbf{A}\mathbf{\Gamma}\|^2 \mathrm{Tr}[\mathbf{B}]\sigma_v^2 + d\sigma_{\mathbf{w}}^2,$$

where

$$\psi(\alpha, \nu) = (1 - 2\alpha(\lambda_{\min}(\mathbf{B}\mathbf{\Gamma}) - \alpha\rho(\nu)\|\mathbf{\Gamma}\|^2))\|\mathbf{A}\|^2, \qquad (4.21)$$
$$\rho(\nu) = (M_4 + (M_{\mathbf{u}} + 1/\nu)M_{\mathbf{u}}\mathrm{Tr}[\mathbf{B}])/2.$$

A direct conclusion of the next theorem follows from the last inequality.

Theorem 4.6. *Assume that the sequences* $\{y_t\}$,$\{\mathbf{u}_t\}$,$\{v_t\}$,$\{\mathbf{x}_t\}$ *and* $\{\mathbf{w}_t\}$ *are related by Equations* (4.18) *and* (4.19), $\alpha > 0$, $\mathbf{\Gamma}$ *is a positive-definite matrix, and* $\widehat{\mathbf{x}}(0)$ *is an arbitrary nonrandom vector from* $\mathbb{R}^d$.

If *the Assumptions As4.1,4.2',4.3,4.5,4.6,4.8 are satisfied*

then *for the prediction errors of the estimates* $\{\widehat{\mathbf{x}}(t)\}$ *generated by the algorithm* (4.20) *for any* $\nu > 0$ *and a sufficiently small* α *such that* $\psi(\alpha, \nu) < 1$, *the following inequality are satisfied*

$$E \parallel \widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1} \parallel^2 \leq \frac{d\sigma_{\mathbf{w}}^2 + \alpha^2(1 + M_{\mathbf{u}}\nu) \parallel \mathbf{A\Gamma} \parallel^2 \mathrm{Tr}[\mathbf{B}]\sigma_v^2}{1 - \psi(\alpha, \nu)} + \tag{4.22}$$

$$+\psi(\alpha, \nu)^t E \parallel \widehat{\mathbf{x}}(0) - \mathbf{x}_0 \parallel^2, \; t = 0, 1, \ldots,$$

where function $\psi(\alpha, \nu)$ *is defined by the Equation* (4.21).

Note, that the result of Theorem 4.6 is an accurate in the sense that in typical cases inequality (4.22) is transformed into the equation when inequalities are replaced with equalities in the Theorem 4.6 conditions.

The second term on the right side of inequality (4.22) shows the contribution of uncertainty about the initial data and tends to approach zero exponentially over time.

It is interesting to analyze the first term on the right side of inequality (4.22). The boundedness of disturbances, which is usually assumed in minimax filtering problems, leads to results whose accuracy is proportional to specific sizes of an uncertainty set. Inequality (4.22) shows the unexpected novel feature of a randomized algorithm (to be more exact, an algorithm with a randomized measurement process when the current estimate is changed in the chosen random direction). If the upper bound $\sigma_{\mathbf{w}}^2$ of the variance of the uncontrollable part of the process under study is sufficiently small, then it is possible to obtain small prediction errors as compared with the level of observation noise σ_v.

Theorem 4.6 facilitates studying the filtering performance dependence on the step-size α of algorithm (4.20). In the case $\|\mathrm{A}\| < 1$, for a sufficiently small α we have

$$E\|\widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1}\|^2 \leq \frac{d\sigma_{\mathbf{w}}^2}{1 - \|\mathrm{A}\|^2} + \|\mathrm{A}\|^{2t} E \parallel \widehat{\mathbf{x}}(0) - \mathbf{x}_0 \parallel^2 + \mathcal{O}(\alpha^2).$$

Let's suppose that $\|\mathrm{A}\| = 1$, $\mathbf{\Gamma} = \mathbf{B}^{-1}$, $\mathrm{E}\{\|\widehat{\mathbf{x}}_0 - \mathbf{x}_0\|^2\} = 0$, and α is a sufficiently small. Then from Theorem 4.6 we obtain

$$E\|\widehat{\mathbf{x}}(t+1) - \mathbf{x}_{t+1}\|^2 \leq c_{-1}\frac{1}{\alpha} + c_0(\nu) + c_1(\nu)\alpha + \mathcal{O}(\alpha^2), \tag{4.23}$$

where

$$c_{-1} = \frac{d\sigma_{\mathbf{w}}^2}{2}, \; c_0(\nu) = \frac{d\sigma_{\mathbf{w}}^2 r(\nu)}{2||\mathbf{B}||^2}, \; c_1(\nu) = \frac{d\sigma_{\mathbf{w}}^2 r(\nu)^2}{2||\mathbf{B}||^4} + \frac{(1 + M_{\mathbf{u}}\nu)\mathrm{Tr}[\mathbf{B}]\sigma_v^2}{||\mathbf{B}||^2}.$$

The expression (4.23) roughly indicates the tradeoff between filtering ability and noise sensitivity. In the case of $M_{\mathbf{u}} = 0$, a similar result can be obtained from the inequality (3.15) as a corollary to Theorem 3.4 for a tracking problem.

By minimizing first three items of (4.23) in α and ν when $\sigma_{\mathbf{w}} > 0$, we establish $\alpha^\star = \bar{D}(\nu^\star)$ where

$$\tilde{D}(\nu) = \left(\frac{\rho(\nu)^2}{||\mathbf{B}||^4} + \frac{(1 + M_{\mathbf{u}}\nu)\mathrm{Tr}[\mathbf{B}]\sigma_v^2}{d\sigma_{\mathbf{w}}^2 ||\mathbf{B}||^2} \right)^{-\frac{1}{2}},$$

and $\nu^\star$ is the minimum point of the function

$$\bar{D}(\nu) = d\sigma_{\mathbf{w}}^2 \left(\frac{\rho(\nu)}{2||\mathbf{B}||^2} + \bar{D}(\nu) \right). \tag{4.24}$$

If $M_{\mathbf{u}} = 0$, then the function $\bar{D}(\nu)$ is independent of ν. In this case, we have

$$\alpha^\star = 2||\mathbf{B}||^2 \sqrt{\frac{d\sigma_{\mathbf{w}}^2}{M_4^2 d\sigma_{\mathbf{w}}^2 + 2||\mathbf{B}||^2 \mathrm{Tr}[\mathbf{B}]\sigma_v^2}}$$

and

$$\bar{D}^\star = \frac{d\sigma_{\mathbf{w}}^2}{4||\mathbf{B}||^2} \left(M_4 + 2\sqrt{M_4^2 + 2||\mathbf{B}||^2 \mathrm{Tr}[\mathbf{B}]\sigma_v^2/(d\sigma_{\mathbf{w}}^2)} \right).$$

From this equation we easily conclude that it is better to use the probabilistic Bernoulli distribution (± 1) for simulation of the random vectors $\mathbf{u}_n$ when they can be chosen arbitrary from a d-dimensional cube $[-1, 1]^d$.

Let $d = 1$, $\sigma_w^2 << \sigma_v^2$, $\{u_t\}$ be a scalar Bernoulli independent process (equal to $\pm\tilde{u}$ with the same probability). Then, $\alpha^\star A\Gamma \approx \sigma_w/|\tilde{u}|\sigma_v$. Note that this value approximately coincides with the limiting value of the Kalman coefficient for the optimal Kalman filter when observation noises $\{v_t\}$ are independent and equal to $\pm\sigma_v$ with the identical probability.

Example. We now discuss the example of applying algorithm (4.20) to the prediction problem for the scalar ($d = 1$) signal $\{\mathbf{x}_n\}$ generated through a linear filter (4.19) with A = 1 and $x_0 = 0$ by an independent process $\{w_t\}$ which is uniformly distributed over the interval $[-\frac{1}{3}, \frac{1}{3}]$: $Ew_t = 0$, $Ew_t^2 = \frac{2}{81}$. The sequence $\{u_t\}$ consist of random variables generated by the uniform distribution over the interval $[0.5; 1.5]$. The quantities y_t and u_t are available at each time t. They are related to the signal x_t by the equation (4.18) with the immeasurable bounded noise (disturbance): $|v_t| \leq 2$.

Minimization of the function $\bar{D}(\nu)$ of (4.24) in ν provides $\nu^\star = 0.269$. Hence for algorithm (4.20) the optimal step–size is $\alpha^\star = 11.3808$, $\Gamma = 1/48$, and we can get that the assessment of filtering error is $\bar{D}^\star = 1.3699$. Note, this assessment sufficiently less than the squared observation noise level $\sigma_v^2 = 4$ which includes usually into correspondence bounds for any others algorithms of minimax filtering under unknown but bounded observation noises.

4.2 Random Projections and Sparsity

Information processing and control decision-making may be schematized as follows (see Figure 4.4).

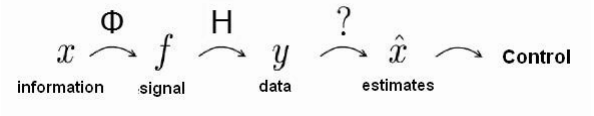

Fig. 4.4. Diagram of data acquisition and processing

We assume that information $\mathbf{x}$ about some variable of phenomenon (or control plant, or some element of the environment) that is important for the researcher mediated by some forms of communication $\mathbf{\Phi}$, manifests itself via the signal f.The signal f be observed using some recording instrumentation or special sensors. The process of observation (data acquisition) is usually sufficiently complicated and may be treated as application of some observation operator $\mathbf{H}$ to the signal f.The data $\mathbf{y}$ are obtained as the result of interaction with the recording instrumentation are used by the researcher in an attempt to restore (estimate) the information $\mathbf{x}$ by generating an estimate $\widehat{\mathbf{x}}$. The estimation results characterize to a certain extent the changes occurring in the phenomenon of interest to the researcher and are then used to make one or another control decision.

The stage of data acquisition plays a key part.

All processes and phenomena occur in time. In reality, all signals are analog in nature. The simplest example of the operator $\mathbf{H}$ is represented by acquiring a set of "instantaneous" values of the signal f at m points. In this case, the operator $\mathbf{H}$ is formally representable as a correspondence between a signal f and a vector of its convolutions with argument-shifted delta-functions.

4.2.1 Compressed (Spars) Signals

The modern information theory originated from the famous *Nyquist–Shannon theorem* [207, 249, 290], which states that if an analog signal $f : \mathbb{R} \to \mathbb{R}$ from $L_2(\mathbb{R})$ has a limited spectrum, it can be restored uniquely without losses to its discrete readings taken at a frequency greater than the doubled maximal frequency of the spectrum. This theorem in many cases makes it possible to identify the analog signal f with a bounded spectra through its set of discrete values $f_t,\ t \in 1..T$.

Instantaneous data are an idealization. The real registering instruments interact with the arriving input over some period of time, and only then they output the result, which in fact is a certain integral characteristic of the signal over the registration interval. In the general case, the result of measuring the

signal f from the space of functions $L_2(\mathbb{R})$ (data acquisition) may be treated as a scalar product in $L_2(\mathbb{R})$ with some observation function h (a convolution of f with h).

Compressed signals. Let us consider a real one-dimensional finite-length discrete signal $\mathbf{f}$. Its values make up an $T \times 1$ column vector in $\mathbb{R}^T$ with the elements $f_t,\ t \in 1..T$. (We consider a 2-D image or high-dimensionality signal as vectorized in a long one-dimensional vector.) Any signals in $\mathbb{R}^T$ may be expanded in some basis of the $T \times 1$-dimensional vectors $\{\varphi_j\}_{j=1}^T$. We assume for simplicity that this basis is orthonormalized. If we define the $T \times T$ matrix $\mathbf{\Phi} = (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_T)$ with columns of basis vectors $\{\boldsymbol{\varphi}_j\}$, a signal $\mathbf{f}$ is then representable as

$$\mathbf{f} = \mathbf{\Phi}\mathbf{x} = \sum_{j=1}^{T} x_j \boldsymbol{\varphi}_j, \tag{4.25}$$

where $\mathbf{x}$ is an $T \times 1$ column vector of weight coefficients $x_j = \langle \mathbf{x}, \boldsymbol{\varphi}_j \rangle = \boldsymbol{\varphi}_j^{\mathrm{T}} \mathbf{x}$. Obviously, $\mathbf{f}$ and $\mathbf{x}$ are equivalent representations of the signal. Usually, $\mathbf{f}$ is called the representation in the time (or space) domain, and $\mathbf{x}$ is the representation in the spectral or transformed $\mathbf{\Phi}$-domain.

The signal $\mathbf{f}$ is referred to as *s-sparse* if it is a linear combination of s basis vectors only, that is, only s components x_j in (4.25) are not equal to zero and the rest $(T - s)$ of them are zero.

The case of interest is $s << T$. For such s-sparse signals, one may assume that it is namely their representation in the corresponding $\mathbf{\Phi}$-domain that is the essential information they carry. This information is defined uniquely by two sets of s natural indices and real values.

Along with the definition of the s-sparse signals, we use a more general notion of the compressed signal. The signal $\mathbf{f}$ is called compressed if it has a representation similar to (4.25) where only several components x_j are large enough and the rest of them are small.

Remarks. It deserves noting that some sort of sparse signal representation is a frequent but not obligatory requirement. It was omitted and replaced by the requirement for existence of a manifold on which $\mathbf{f}$ lies. In some studies, the requirement on sparseness is relaxed to a sparse representation in some redundant dictionary. Apart from sparse representation, additional constraints such as generation of this representation by a random Markov field are sometimes applied to the signal. A case where vectors $\mathbf{y}$ and $\mathbf{x}$ are parameterized by an infinite countable or even noncountable set was discussed in [237].

4.2.2 Transforming Coding

The fact that the compressed signals are well approximated by the s-sparse representations underlies the transforming coding often encountered even in everyday life using JPEG, MP3, and MPEG formats for images, audio, and video. In data acquisition systems (for example, digital photo and video cameras), the transforming coding is pivotal: assembly of the transformation co-

efficients $\{x_1, x_2, \ldots, x_T\}$ is calculated from the obtained full T-sample of the signal $\mathbf{f}$ via $\mathbf{x} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{f}$, then s large components are localized in $\mathbf{x}$, with the remaining $T - s$ being rejected (see, for example, [311] for detail). As the result, the codes are themselves s values and the numbers of their positions in the vector $\mathbf{x}$.

One of the following orthogonal transforms is usually employed in image processing: discrete Fourier transform (DFT), discrete cosine transform (DCT) or discrete wavelet-transform (DWT).

We recall that DFT of a T-dimensional signal $\mathbf{f}$ considered as a vector is defined as a vector $\tilde{\mathbf{f}}$ with components

$$\tilde{f}_k = \sum_{j=1}^{T} f_j e^{-2\pi \mathrm{i}(j-1)k/T}$$

where i is the imaginary unit. It is an invertible linear transformation with a matrix:

$$\mathbf{\Upsilon} = \begin{pmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 1 & e^{-\frac{2\pi \mathrm{i}}{T}} & e^{-\frac{4\pi \mathrm{i}}{T}} & e^{-\frac{6\pi \mathrm{i}}{T}} & \ldots & e^{-\frac{2\pi \mathrm{i}}{T}(T-1)} \\ 1 & e^{-\frac{4\pi \mathrm{i}}{T}} & e^{-\frac{8\pi \mathrm{i}}{T}} & e^{-\frac{12\pi \mathrm{i}}{T}} & \ldots & e^{-\frac{2\pi \mathrm{i}}{N}2(N-1)} \\ 1 & e^{-\frac{6\pi \mathrm{i}}{T}} & e^{-\frac{12\pi \mathrm{i}}{T}} & e^{-\frac{18\pi \mathrm{i}}{T}} & \ldots & e^{-\frac{2\pi \mathrm{i}}{T}3(T-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi \mathrm{i}}{T}(T-1)} & e^{-\frac{2\pi \mathrm{i}}{T}2(T-1)} & e^{-\frac{2\pi \mathrm{i}}{T}3(T-1)} & \ldots & e^{-\frac{2\pi \mathrm{i}}{T}(T-1)^2} \end{pmatrix}.$$

DCT is an orthogonal transform closely related with the Fourier transform but operating with real numbers. It is defined by a similar matrix with the elements $\cos\ k(j + \frac{1}{2})\frac{\pi}{T}$.

DWT is based on the convolution of the signal with filters of high and low frequencies with successive double sparsing of the number of readings. At that, the sparsed result of convolution with the low-frequency filter may be regarded as a reduced copy of the original signal which was purged of noise and over which the given operation is repeated. DWT is also a linear operator.

The corresponding two-dimensional invertible linear transforms are defined similarly. For example, the two-dimensional Fourier transform of the vector $T_1 \times T_2$ is defined as follows:

$$\tilde{f}_{k_1,k_2} = \sum_{j_1=1}^{T_1} \sum_{j_2=1}^{T_2} f_{j_1,j_2} e^{-2\pi \mathrm{i}(j_1-1)k_1/T_1} e^{-2\pi \mathrm{i}(j_2-1)k_2/T_2}.$$

This is true for the majority of real-world images for which application of these transforms provides transformed signals with a substantially smaller number of components.

Unfortunately, transforming coding in which a full sample is first obtained and then compression is performed has three inherent disadvantages:

- First, the initial size of the sample T may be extremely large even if the size s of the resulting assembly of components is small.
- Second, all T transformation coefficients $\{x_j\}_{j\in 1..T}$ must be calculated even though the majority, except for s coefficients, will be rejected.
- Third, the positions of s large coefficients must be additionally coded.

4.2.3 Compressive Sensing

The aforementioned disadvantages of the transforming coding are eliminated by using the compressive sensing (CS) owing to the direct determination of the compressed representation of the signal without the intermediate determination of the T-sample [65, 97].

Let us consider the general linear measurement process calculating $m < T$ internal scalar products of $\mathbf{f}$ and vectors $\mathbf{h}_i,\ i \in 1..m$:

$$y_i = \langle \mathbf{h}_i, \mathbf{f} \rangle.$$

We collect the results y_i in an $m \times 1$ column vector $\mathbf{y}$ and generate the rows of the $m \times T$ matrix $\mathbf{H}$ from the transposed measurement vectors $\mathbf{h}_i^{\mathrm{T}}$. By substituting $\mathbf{\Phi}$ from (4.25), the expression of $\mathbf{y}$ may be rearranged in

$$\mathbf{y} = \mathbf{H}\mathbf{f} = \mathbf{H}\mathbf{\Phi}\mathbf{x} = \mathbf{U}\mathbf{x}, \tag{4.26}$$

where $\mathbf{U} = \mathbf{H}\mathbf{\Phi}$ is the $m \times T$ matrix.

Another possible approach to measurement is described in [50], where each sensor measures only one signal bit, the *sign* of projection of $\mathbf{f}$ on $\mathbf{h}_i$:

$$y_i = \mathrm{sign}\langle \mathbf{h}_i, \mathbf{f} \rangle.$$

The condition $\mathbf{y}^{\mathrm{T}}\mathbf{U}\mathbf{x} \geq 0$ is used in this case instead of condition (4.26).

We notice that the considered measurement process is nonadaptive in the sense that a matrix $\mathbf{H}$ is fixed and independent of a signal $\mathbf{f}$.

The problem of compressive sensing lies in designing

- a universal measurement matrix $\mathbf{H}$ such that essential information about any s-sparse (or compressed) signal is not damaged when reducing the dimensionality from $\mathbf{f} \in \mathbb{R}^T$ to $\mathbf{y} \in \mathbb{R}^m$, and
- a reconstruction algorithm of restoring $\mathbf{x}$ — and, consequently, $\mathbf{f}$ — only from $m \sim s$ measurements (or approximately the same number of measurements as a transmitted volume of data at the traditional transforming coding).

Expression (4.26) should not be regarded as a process of "multiplying" the signal $\mathbf{f}$ by the matrix $\mathbf{H}$. In the CS paradigm this measurement process is not usually carried out separately, but is part of the physical process of data acquisition, which facilitates doing without excessive computer power and saves power in the sensing devices. The particular form taken by matrix

$\mathbf{H}$ is important in the calculations of signal restoration from the acquired compressed data. For example, if a problem offers a technical opportunity for direct determination of the coefficients of the Fourier transform of $\mathbf{f}$, the corresponding vector $\mathbf{y}$ is then obtained directly from a selected assembly of m indices, and the rows $\mathbf{h}_i^{\mathrm{T}}$ of the matrix $\mathbf{H}$ from (4.26) are the corresponding rows of the matrix of the Fourier transform. Another illustrative example may be found at http://dsp.rice.edu/cscamera where consideration is given to a single-pixel compressing digital camera [98] that directly obtains (in an analog manner by using only lenses and mirrors) m random projections of the desired image without preliminary collection of the values (data acquisition) of all its T pixels. The light waves emerging from the picture $\mathbf{f}$ are reflected by a special device of digital micromirrors (DMD) consisting of an array of T tiny mirrors. Such DMD devices can be found in many computer projectors and projection TV sets. The reflected light is collected by the second lens and focused on a single-pixel photodiode (PD). Each micromirror can be oriented independently either in the direction of the photodiode (corresponds to one) or to the side (corresponds to zero). To collect the measurements, pseudorandom orientations of the micromirrors creating the measurement vector $\mathbf{h}_i$ of ones and zeros are established using a random number generator. The photodiode voltage is then equal to y_i which is the internal product of $\mathbf{h}_i$ and the image $\mathbf{f}$, that is, the reflected rays are summed optically. To obtain all components of $\mathbf{y}$, the process is repeated m times. In this way, we obtain the measurement vector $\mathbf{y}$ of (4.26), where multiplication by the matrix is part and parcel of a special physical process of measurement.

4.2.4 Universal Measurement Matrix: Random Projections

The measurement matrix $\mathbf{H}$ should enable reconstruction of the signal $\mathbf{f}$ of length T from a smaller number of measurements $m < T$ (from the vector $\mathbf{y}$). Since $m < T$, this problem is ill-conditioned. However, if $\mathbf{f}$ is s-sparse and one is aware of the locations of the s other-than-zero components $\mathbf{x}$, the problem may then be solvable for $m \geq s$. The necessary and sufficient condition for solvability of this simplified problem lies in that the inequalities

$$\lambda^{-1}||\mathbf{z}||_2 \leq ||\mathbf{U}\mathbf{z}||_2 \leq \lambda||\mathbf{z}||_2, \tag{4.27}$$

must be satisfied for some $0 < \lambda < \infty$ and any nonzero vector $\mathbf{z}$ where the same s nonzero components as in $\mathbf{x}$ are specified, that is, the matrix $\mathbf{U} = \mathbf{H}\boldsymbol{\Phi}$ must retain the lengths of such specific, in a sense s-sparse, vectors.

In the general case, of course, the positions of the s other-than-zero components of $\mathbf{x}$ are unknown. However, satisfaction of condition (4.27) for arbitrary $2s$-sparse vectors $\mathbf{z}$ suffices for stable solution of the problem for any s-sparse vector $\mathbf{x}$. One can ensure the validity of this fact using the proof by contradiction. The following unique decoding rule may be used: among all vectors $\mathbf{x}$ such that $\mathbf{y} = \mathbf{U}\mathbf{x}$, we select the one with the least number of nonzero coefficients. Let a problem have two different solutions $\mathbf{x}'$ and $\mathbf{x}''$.

Obviously, they have at most s nonzero components each. Since $\mathbf{x}'$ and $\mathbf{x}''$ are s-sparse vectors, $\bar{\mathbf{x}} = \mathbf{x}' - \mathbf{x}''$ is a $2s$-sparse vector. By virtue of linearity, $\mathbf{U}\bar{\mathbf{x}} = \mathbf{U}\mathbf{x}' - \mathbf{U}\mathbf{x}'' = 0$ and, therefore, we have $\bar{\mathbf{x}} = 0$ in virtue of (4.27), that is, $\mathbf{x}' = \mathbf{x}''$, which is a contradiction.

The literature on CS most frequently uses another condition allied to (4.27), known as the *restricted isometry property* (RIP): the $m \times T$ matrix $\mathbf{U}$ has RIP(δ, m) with the parameters $\delta \in (0, 1)$ and $m \in \mathbb{N}$ if inequalities

$$\sqrt{1-\delta} \le \frac{||\mathbf{U}\mathbf{z}||_2}{||\mathbf{z}||_2} \le \sqrt{1+\delta} \tag{4.28}$$

are satisfied for any nonzero m-sparse vector $\mathbf{z}$. In the literature, condition (4.27) is known as the *modified RIP (MRIP)*.

Along with RIP-like properties, CS makes use of the condition

$$\mu(\mathbf{H}, \mathbf{\Phi}) = \sqrt{T} \max_{i,j} \frac{|\langle \mathbf{h}_i, \boldsymbol{\varphi}_j \rangle|}{||\mathbf{h}_i||_2},$$

for smallness of mutual dependence $\mu(\mathbf{H}, \mathbf{\Phi})$ of the rows $\{\mathbf{h}_i^{\mathrm{T}}\}$ of the matrix $\mathbf{H}$ and the columns $\{\boldsymbol{\varphi}_j\}$ of the matrix $\mathbf{\Phi}$, which is called the *noncoherence* of $\mathbf{H}$ and $\mathbf{\Phi}$. To satisfy this condition, the rows of the matrix $\mathbf{H}$ should represent the columns of $\mathbf{\Phi}$ (and vice versa) as a linear combination with the majority of the coefficients being zeros. CS makes frequent use of the fact that the random matrices $\mathbf{H}$ with high probability are strongly noncoherent with any fixed basis $\mathbf{\Phi}$. For example, if the rows of the matrix $\mathbf{H}$ represent a random sample of an orthonormalized basis obtained by orthogonalization of T random vectors selected uniformly and independently from the unit sphere, then $\mu(\mathbf{H}, \mathbf{\Phi}) \approx \sqrt{2 \log T}$. A sufficient condition for the possibility of highly probable precise restoration from m observations with matrix $\mathbf{H}$ was obtained in [64] for the s-sparse vectors:

$$m \;\ge\; c\,\mu(\mathbf{H}, \mathbf{\Phi})^2\, s\, \log T, \tag{4.29}$$

where c is a positive constant. For the above random matrix $\mathbf{H}$, we have $m \ge 2cs(\log T)^2$.

As was indicated in [66], systems with $m \approx 4s$ have been proven to work well in practice.

The conditions RIP$(\delta, 2s)$ and MRIP$(\lambda, 2s)$ are non-robust in the sense that they are not sufficient for restoration of

- an arbitrary s-sparse signal under noisy observations $\mathbf{y}$ or
- a compressed signal with small nonzero $(T - s)$ components.

In these cases, the conditions RIP$(\delta, 3s)$ or MRIP$(\lambda, 3s)$ are sufficient in the sense (see [66]) that

$$||\mathbf{x} - \widehat{\mathbf{x}}||_1 \;\le\; const\; ||\mathbf{x} - \mathbf{x}^\star||_1,$$

where $\widehat{\mathbf{x}}$ is the result of restoration of $\mathbf{x}$ and $\mathbf{x}^{\star}$ is the vector obtained from $\mathbf{x}$ by zeroing all components, with the exception of s greatest-in-magnitude components. In this case, the condition for a high degree of noncoherence is overly conservative because the permissible value of $\mu(\mathbf{H}, \mathbf{\Phi})$ is related to RIP$(\delta, 3s)$ as $\delta = (m-1)\mu(\mathbf{H}, \mathbf{\Phi})$.

Direct construction of the measurement matrix $\mathbf{H}$ such that $\mathbf{U} = \mathbf{H\Phi}$ features RIP requires verification of condition (4.28) for each of $\binom{T}{s} = \frac{T!}{s!(T-s)!}$ possible subsets of the positions of s other-than-zero components in the vector $\mathbf{z}$ of length T. However, it turned out that RIP may be satisfied with high probability just by taking a random matrix as $\mathbf{H}$ (randomization of observations). At that, the vector of measurement results $\mathbf{y}$ is an assembly of m different linear combinations of the components of $\mathbf{f}$ with randomly selected weights.

The random $m \times T$ measurement matrix $\mathbf{H}$ with independent identically distributed (i.i.d.) elements $h_{i,j}$ and normal distribution density with zero means and variance $1/m$

$$h_{i,j} \sim \mathcal{N}\left(0, \frac{1}{m}\right)$$

has two interesting and useful properties [65]:

- if $0 < \delta < 1$ and
$$m \geq c_1\, s\, \log(T/s), \tag{4.30}$$
then, the matrix $\mathbf{H}$ satisfies RIP(δ, m) with the probability $\geq 1 - 2e^{-c_2 m}$, where $c_1,\ c_2 > 0$ are small constants depending only on δ (consequently, s-sparse and compressible signals of length T can be restored with high probability only from $m << T$ random measurements);
- the matrix $\mathbf{H}$ is *universal* in the sense that when dimensionality is reduced from $\mathbf{f} \in \mathbb{R}^T$ to $\mathbf{y} \in \mathbb{R}^m$ not only is essential information about any s-sparse (or compressed) signal not damaged, but also the matrix $\mathbf{U} = \mathbf{H\Phi}$ is random with normally distributed i.i.d. elements; therefore, $\mathbf{U}$ will feature RIP(δ, m) with the same high probability independent of the method of selecting the orthonormalized basis $\mathbf{\Phi}$.

The result regarding satisfying condition RIP(δ, m) with high probability from property (4.30) of the random matrix $\mathbf{H}$ relies largely on earlier publications [183].

Other random measurement matrices such as random sample of the i.i.d. elements $h_{i,j}$ from the symmetrical Bernoulli distribution

$$\mathrm{P}(h_{i,j} = \pm 1/\sqrt{m}) = \frac{1}{2}$$

or a similar distribution

$$h_{i,j} = \begin{cases} +\sqrt{3/m} & \text{with probability } \frac{1}{6} \\ 0 & \text{with probability} \frac{2}{3} \\ -\sqrt{3/m} & \text{with probability} \frac{1}{6} \end{cases}$$

may be used for CS.

It was shown in [25] that property (4.30) can be easily derived from the Johnson–Lindenstrauss lemma [178] according to which any set of T points in the d-dimensional Euclidean space may be embedded in the m-dimensional Euclidean space so that $m \sim \log T$, m is independent of d, and all mutual pairwise distances are approximately retained [25]. For the aforementioned last three methods of generating the random matrix $\mathbf{H}$, its universality was substantiated and particular conditions for selection of the constants c_1 and c_2 were given in [25]:

$$c_2 \le c_0(\delta/2) - c_1\left(1 + \frac{1 + \log(12/\delta}{\log(T/s)}\right),\ c_0(\delta/2) = \frac{\delta^2}{16} - \frac{\delta^3}{48}.$$

The proof of the property RIP(δ, m) for the random matrix $\mathbf{U}$ in [25] follows the lines of [1] in using the possibility of covering each of the C_T^m possible m-dimensional subsets of the set $\{\mathbf{z}:\ \|\mathbf{z}\|_2 = 1\}$ by at most $(12/\delta)^m$ spheres of the radius $\delta/4$ and relies on the proof that the expectation

$$E\|\mathbf{U}\mathbf{z}\|_2^2 = \|\mathbf{z}\|_2^2\ \ \forall \mathbf{z} \in \mathbb{R}^T$$

and the random variables $\|\mathbf{U}\mathbf{z}\|_2^2$ are strictly concentrated about $\|\mathbf{z}\|_2^2$:

$$\text{Prob}\{|\|\mathbf{U}\mathbf{z}\|_2^2 - \|\mathbf{z}\|_2^2| \ge \frac{\delta}{2}\|\mathbf{z}\|_2^2\} \le 2e^{-mc_0(\delta/2)}.$$

Other randomized methods of generation of the measurement matrix are also proposed in the literature (see, for example, [66, 97]):

- uniform random sample of T columns on the unit sphere in $\mathbb{R}^m$;
- random sample of the projector $\mathcal{P}$ and its normalization $\mathbf{H} = \sqrt{\frac{T}{m}}\mathcal{P}$ or
- choice by some other sub-Gaussian distribution.

Satisfaction of RIP(δ, m) is supported in each of these cases by condition (4.30) with an appropriate constant c_1 depending on the selected method of generating the random matrix $\mathbf{H}$.

Storage of $m \times T$ elements of the matrix $\mathbf{H}$ requiring $\mathcal{O}(mT)$ memory units becomes a challenge as T increases. For simple rules of generation of $\mathbf{H}$, its impact may be reduced by using, for example, a "repeated" generator of pseudo-random numbers and again recalculating $\mathbf{H}$ each time. The random "cuts" of m rows from the $T \times T$ matrices of the discrete Fourier transform (DFT) or the Vandermonde matrices corresponding to interpolation at different T points offer other examples of matrices satisfying the RIP and MRIP conditions. The size of memory for such random matrices may be reduced to $\mathcal{O}(m \log T)$. If the rows of matrix $\mathbf{H}$ are a random sample of transposed columns of an orthonormalized basis obtained by orthogonalization of T random vectors selected uniformly and independently from the unit sphere $\mathbb{R}^T$, then to satisfy the property RIP(δ, m) with high precision, it suffices to take m:

$$m\ \ge\ C\, s\, (\log T)^4.$$

The difficulty of verifying RIP for larger m is its most substantial disadvantage. Active attempts being made to replace it by more constructive properties. An interesting example of such a possibility is described in [180].

4.2.5 Reconstruction Algorithms through ℓ_1-Optimization and Others

Design of a signal reconstruction algorithm. The signal reconstruction algorithm must reconstruct the signal $\mathbf{f}$ of length T or its corresponding sparse vector of coefficients $\mathbf{x}$ from m measurements (vector $\mathbf{y}$), random measurement matrix $\mathbf{H}$ (or random law of its generation) and basis $\mathbf{\Phi}$. Since $m < T$ in (4.26), for the s-sparse signals there are infinitely many $\mathbf{x}'$ satisfying $\mathbf{U}\mathbf{x}' = \mathbf{y}$ due to the fact that if $\mathbf{U}\mathbf{x} = \mathbf{y}$ then $\mathbf{U}(\mathbf{x} + \mathbf{z}) = \mathbf{y}$ for any vector $\mathbf{z}$ from the zero subspace $\mathcal{N}ull(\mathbf{U})$ of the matrix $\mathbf{U}$. Therefore, the aim of the signal reconstruction algorithm is to determine a vector of the coefficients of a sparse representation of the signal in $(T - m)$-dimensional shifted zero subspace $\mathcal{H} = \mathcal{N}ull(\mathbf{U}) + \mathbf{x}$.

Reconstruction by minimization of the ℓ_2-norm. A classical approach to the inverse problems of the type at hand lies in determining a vector with the least norm (power) ℓ_2 in the shifted zero subspace $\mathcal{H}$

$$\widehat{\mathbf{x}} = \arg\min ||\mathbf{x}'||_2 \ : \qquad \mathbf{U}\mathbf{x}' = \mathbf{y}.$$

Solution of this optimization problem may be conveniently represented in the form of the least-squares method (LSM) as follows:

$$\widehat{\mathbf{x}} = \mathbf{U}^{\mathrm{T}}(\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}\mathbf{y}.$$

Unfortunately, the result of ℓ_2-minimization almost never is an s-sparse vector and has many other- than- zero elements.

Reconstruction by minimization of the ℓ_0-norm. Since the ℓ_2-norm measures the signal power and not its sparseness, it is possible to consider the problem of minimization of the ℓ_0-norm that calculates the number of nonzero components

$$||\mathbf{x}||_0 = |\{i | x_i \neq 0\}|.$$

(Strictly speaking, ℓ_0 is not a norm because it does not satisfy the property of uniformity, but the term "ℓ_0-norm" is widely used.)

Unfortunately, the problem of optimization of the ℓ_0-norm

$$\widehat{\mathbf{x}} = \arg\min ||\mathbf{x}'||_0 \quad : \quad \mathbf{U}\mathbf{x}' = \mathbf{y},$$

is nonconvex and combinatorial, the computational procedures of its solution are numerically unstable and NP-hard and require an enormous enumeration of all $\begin{pmatrix} T \\ s \end{pmatrix}$ possible variants of allocation of the nonzero elements in $\mathbf{x}$.

Reconstruction by minimization of the ℓ_1-norm. It may seem strange, but optimization based on the ℓ_1-norm

$$\widehat{\mathbf{x}} = \arg\min ||\mathbf{x}'||_1 \quad : \quad \mathbf{U}\mathbf{x}' = \mathbf{y},$$

makes it possible to reconstruct the s-sparse signals precisely and with high probability and to approximate the compressed signals using only

$$m \geq c_1\, s\, \log(T/s)$$

i.i.d. random measurements [65, 97]. This is a problem of convex optimization reducible to a problem of linear programming known as *the basis pursuit* problem [78]:

$$\langle \mathbf{c}, \tilde{\mathbf{x}} \rangle \to \min \quad : \quad \tilde{\mathbf{U}}\tilde{\mathbf{x}} = \mathbf{y},\ \tilde{\mathbf{x}} \geq 0,$$

where

$$\mathbf{c} = (1, 1, \ldots, 1)^{\mathrm{T}},\ \ \tilde{\mathbf{x}} = \begin{pmatrix} x_+ \\ x_- \end{pmatrix},\ \ \tilde{\mathbf{U}} = (\mathbf{U}, -\mathbf{U}).$$

The solution may be based on *the interior point method* (computational complexity of the order of $\mathcal{O}(T^3)$) or *the simplex method* (theoretical exponential complexity), which in practice proved to be relatively fast.

Some promising ways to find ℓ_1-optimizing or suboptimizing solutions are described in [136], but these are not yet used for compressive sensing.

Geometrical interpretation. Geometrical representation of the problem of compressive sensing in $\mathbb{R}^T$ illustrates why solving the problem of ℓ_2-optimization does not help in restoring the original sparse signal. Nevertheless, the signal can be reconstructed with the use of ℓ_1-optimization.

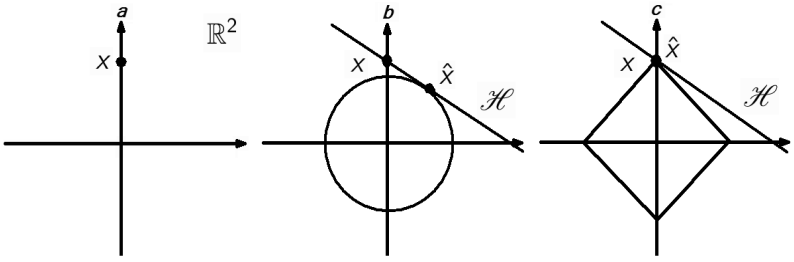

Fig. 4.5. (a) The space of 1-sparse vectors in $\mathbb{R}^2$ consists of two lines having one non-zero coordinate (axes); (b) ℓ_2-minimization determines the non-sparse vector $\widehat{\mathbf{x}}$ distant from $\mathbf{x}$; (c) ℓ_1-minimization determines the sparse point $\widehat{\mathbf{x}}$ of contact of the ℓ_1-sphere with the hyperplane $\mathcal{H}$ coinciding with $\mathbf{x}$ with high probability

The set of all s-sparse vectors $\mathbf{x}$ in $\mathbb{R}^T$ is a strongly nonlinear space consisting of all s-dimensional hyperplanes stretching along the coordinate axes, as

shown in Figure 4.5a for $T = 2$. The shifted zero subspace $\mathcal{H} = \mathcal{N}ull(\mathbf{U}) + \mathbf{x}$ is oriented at a random angle defined by randomization in the matrix $\mathbf{U}$ (see Figure 4.5b). The result of minimization of the ℓ_2-norm is a point on $\mathcal{H}$ which is nearest to the origin and can be determined by extending the hypersphere (ℓ_2-sphere) until touching $\mathcal{H}$. Owing to the random orientation of $\mathcal{H}$, this point $\widehat{x}$ which is nearest to the origin will, with high probability, be far from the coordinate axes and, consequently, neither s-sparse nor close to the correct answer $\mathbf{x}$. On the contrary, the ℓ_1-sphere in Figure 4.5c is a convex combination of points lying on the coordinate axes. Therefore, upon expansion of the ℓ_1-sphere in the two-dimensional space, the point lying on the coordinate axes and coinciding precisely with the point of location of the desired sparse vector $\mathbf{x}$ first comes into contact with the shifted one-dimensional zero subspace $\mathcal{H}$. (In practice, $T, m, s >> 2$ and the intuitive two-dimensional reasoning should not lead astray.)

In the practical problems, reduction of the problem of ℓ_1-optimization to that of linear programming is not satisfactory because of the great computational complexity.

Other methods using the ℓ_1 norm are considered in the literature along with this problem, such as ℓ_1 *regularized LSM* or the so-called *LASSO approach* proposed in [313] and, independently, in [78] as *the basis pursuit de-noising (BPDN)*

$$\widehat{\mathbf{x}}_\lambda = \arg\min ||\mathbf{U}\mathbf{x}' - \mathbf{y}||_2^2 + \lambda ||\mathbf{x}'||_1.$$

Another variant based on shifting to the dual problem was proposed in [80].

Smoothed ℓ_0-norm (*approximation of ℓ_0-norm*). A method based on approximating the delta-function $\delta(t)$ by a smooth function $e^{-\frac{t^2}{2\sigma^2}}$ as $\sigma \to 0$:

$$||\mathbf{x}||_0 = N - \sum_i \delta(x_i) \approx T - \sum_i e^{-\frac{(x_i)^2}{2\sigma^2}} = T - F(\mathbf{x}, \sigma)$$

was proposed in [238]. The nonconvex optimization problem

$$\widehat{\mathbf{x}}_\sigma = \arg\max F(\mathbf{x}', \sigma) \quad : \quad \mathbf{U}\mathbf{x}' = \mathbf{y},$$

is solved for each σ. Local optimization methods can determine the solution only in the case of a good initial approximation. It was proposed in [238] that instead of solving this problem, the result of several iterations of the algorithm of projection of the gradient [258] should be taken as $\widehat{\mathbf{x}}_\sigma$ for each value of σ. Solution of the problem of restoration is obtained as the limit of $\widehat{\mathbf{x}}_\sigma$ as $\sigma \to 0$.

Minimization-based reconstruction of the nonconvex regularizers. Other ℓ_ρ-norms that are used for $0 < \rho < 1$ in addition to the ℓ_1-norm are distinguished for better approximation of the ℓ_0-norm. In particular, an example where $\ell_{\frac{1}{2}}$ is substantially superior to ℓ_1 in terms of restoration performance may be found in [72]. Their weak point lies in that these are not convex functions,

which may result in hitting undesirable local minima. This disadvantage is suppressed by using methods like graduated non-convexity [73].

Iterative algorithms. Some publications suggest to use the iterative weighing algorithms for restoration of the original signal. Weighted ℓ_ρ-norms $(\sum_i \mathbf{w}_i(x_i)^\rho)^{\frac{1}{\rho}}$ with ρ equal to one or two were considered in [74] for the approximation of the ℓ_0-norm. For example, for $\rho = 2$ the weighted ℓ_2-norm is $(\mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{x})^{\frac{1}{\rho}}$ where $\mathbf{W} = \mathrm{diag}(\mathbf{w}_1, \ldots, \mathbf{w}_T)$, and the problem of minimization of the weighted ℓ_2-norm

$$\widehat{\mathbf{x}}(n) = \arg\min(\mathbf{x}')^{\mathrm{T}}\mathbf{W}_n\mathbf{x}' \quad : \quad \mathbf{U}\mathbf{x}' = \mathbf{y},$$

is solved in [74] analytically as follows:

$$\widehat{\mathbf{x}}_n = {\mathbf{W}_n}^{-1}\mathbf{U}^{\mathrm{T}}(\mathbf{U}{\mathbf{W}_n}^{-1}\mathbf{U}^{\mathrm{T}})^{-1}\mathbf{y}.$$

Then,

$$\mathbf{W}_{n+1} = \mathrm{diag}(\ldots, ((\widehat{x}_i(n)^2 + \varepsilon)^{\frac{\rho}{2}-1}, \ldots)$$

where $\varepsilon > 0$ is a small value, are selected iteratively as the following set of the weight coefficients. The solution $\widehat{\mathbf{x}}$ is established as a limit of $\widehat{\mathbf{x}}_n$ as $n \to \infty$.

To solve the problem of restoration of an original signal a series of *iterative shrinkage thresholding (IST)*

$$\widehat{\mathbf{x}}(n+1) = (1-\beta)\widehat{\mathbf{x}}(n) + \beta H(\widehat{\mathbf{x}}(n) + \mathbf{U}^{\mathrm{T}}(\mathbf{y} - \mathbf{U}\widehat{\mathbf{x}}_n))$$

algorithms are proposed in [31, 43, 54]. Here H is the so-called *denoising function* (Moreau proxi-map [43, 83]) as a rule taking the form

$$H(\mathbf{u}) = \arg\min_{\mathbf{w}}(\mathrm{dist}(\mathbf{w}, \mathbf{u}) + \lambda L(\mathbf{w})).$$

Usually, the distance $\mathrm{dist}(\mathbf{w}, \mathbf{u}) = ||\mathbf{w} - \mathbf{u}||_2^2$ is selected, and the second addend is directly proportional to the regularizer $L(\mathbf{q})$ (a penalty function for nonsatisfaction of some property). An example of a penalty function is the Huber function $||\mathbf{q}||_1$.

More efficient modifications of IST proposed in [31, 43] are represented by the *fast IST algorithm* and *the two steps IST (TwIST) algorithm* based on the method of accelerating the optimization algorithm proposed in [258] under the name of "heavy ball".

A similar algorithm was considered in [102] for image restoration from a set of Fourier transform coefficients. H was selected as $H(u) = \boldsymbol{\Upsilon}(\mathcal{R}(\boldsymbol{\Upsilon}^{-1}(u)))$ where $\boldsymbol{\Upsilon}$ is the discrete Fourier transform and $\mathcal{R}$ is an original image denoising algorithm. In contrast to IST, it was proposed in [102] to add an additional random disturbance (randomization) to $\widehat{\mathbf{x}}(n)$ at each iteration, whose irregular part is filtered by the $\mathcal{R}$ algorithm with parameters corresponding to the added disturbance.

"Greedy" algorithms are among the fastest tools for restoration of $\mathbf{x}$ from measurements $\mathbf{y} = \mathbf{U}\mathbf{x}$. The simplest of them is *a matching pursuit (MP)*

[78, 225]. At each step the algorithm specifies in $\mathbf{y}$ the most correlated row $\mathbf{U}$. This simple idea underlies more complicated greedy algorithms that as a rule have the following structure:

1. Initialize $\widehat{\mathbf{x}}(0) = 0$, $\mathbf{w}_0 = \mathbf{y}$, $n = 1$, $\mathbf{\Lambda}_0 = \emptyset$.
2. Estimate the residue $\mathbf{w}_n = \mathbf{y} - \mathbf{U}\widehat{\mathbf{x}}(n-1)$.
3. Establish a new estimate $\widehat{\mathbf{x}}_n$ based on $\mathbf{w}_n$, $\mathbf{U}$ and $\mathbf{y}$. Each algorithm has its own this step, but usually similar to the following:
   (*a*) use any method to calculate one or more current indices $\{\lambda_{n1}, \ldots, \lambda_{nk_n}\}$ of nonzero components of the vector $\mathbf{x}$ which were not yet included in $\mathbf{\Lambda}_{n-1}$, $\mathbf{\Lambda}_n = \mathbf{\Lambda}_{n-1} \cup \{\lambda_{n1}, \ldots, \lambda_{nk_n}\}$;
   (*b*) compile the matrix $\mathbf{U}_n$ of columns of $\mathbf{u}$ with corresponding indices from $\mathbf{\Lambda}_n$;
   (*c*) estimate the nonzero values of the vector $\mathbf{x}_n$ (we denote $\widehat{\mathbf{x}}(n) = \mathbf{x}_n[\mathbf{\Lambda}_n]$):
   $$\widehat{\mathbf{x}}(n) = \arg\min ||\mathbf{U}_n\widehat{\mathbf{x}}(n) - \mathbf{y}||_2.$$
   The components with indices not belonging to $\mathbf{\Lambda}_n$ are assumed to be zero.
4. $n := n + 1$, Return to Step 2 if the stop condition
   $$||\mathbf{w}_n|| < threshold$$
   is not met.

The orthogonal MP (OMP) is one of the most popular robust algorithms in the MP family [89, 321]. In OMP a set $\mathbf{\Lambda}_n$, which is taken at step 3a, consists of a single index
$$\arg\max_i |\langle \mathbf{w}_n, \mathbf{u}_i \rangle|$$
where $\mathbf{u_i}$ is the i-th column of the matrix $\mathbf{U}$.

The regularized OMP (ROMP) [243] selects a set of indices at step 3a using two additional steps:

- calculate a set J of s indices with the greatest scalar value $\langle \mathbf{w}_n, \mathbf{u}_i \rangle$;
- of all subsets I of the set J such that
$$|\langle \mathbf{w}_n, \mathbf{u}_i \rangle| \le 2|\langle \mathbf{w}_n, \mathbf{u}_j \rangle|, \ \forall i, j \in I,$$
take that for which $(\mathbf{U}^{\mathrm{T}}\mathbf{w})[I]$ has the greatest ℓ_2-norm.

The compressive sampling matching pursuit (CoSaMP) [244] was inspired by the ROMP algorithm and repeats it almost literally, except that it does not take all indices i for which $|\langle \mathbf{w}_n, \mathbf{u}_i \rangle|$ exceeds the threshold but rather exactly the $2s$ greatest indices. The vector resulting from the solution of
$$\arg\min ||\mathbf{U}_n\widehat{\mathbf{x}}(n) - \mathbf{y}||_2$$
by zeroing all components except for s components having the greatest magnitude is taken as the estimate $\widehat{\mathbf{x}}(n)$.

In *the DThresh method* [81], the condition for selection of indices is as follows:

$$|\langle \mathbf{w}_n, \mathbf{u}_i \rangle| \geq \frac{\delta ||\mathbf{w}_n||}{\sqrt{s}}.$$

The lattice matching pursuit (LaMP) algorithm was proposed in [69] for the case where the signal is generated by a random Markov field. Its scheme differs from the general scheme. The set of indices is completely reevaluated according to the most probable configuration of the given Markov field instead of accumulating the indices of the nonzero components in $\mathbf{\Lambda}_n$.

The algorithms of gradient pursuit and conjugate gradient pursuit were described in [48], and *sequentially sparse matching pursuit (SSAMP)* in [38].

A method of restoration based on *the belief propagation (BP) algorithm* used in the graphic models for preset statistical dependencies between the components x_j was proposed in [28]. In the presence of cycles in such models, the BP algorithm (called in this case *the cyclic BP*) may converge to a local minimum or not converge at all. However, for practical problems it usually gives a very good approximation.

The combinatorial restoration algorithms are described in [85].

Along with the iterative combinatorial CS algorithm based on using a connectivity matrix of a certain regular graph as the special-form measurement matrix and having the computational complexity of the order of $\mathcal{O}(T \log \frac{T}{s})$, [173] also compares different approaches in detail. Another table comparing the efficiencies of the main signal restoration algorithms is given in [39].

4.2.6 Some Applications

(1) Let's consider a spectrally sparse signal

$$f_t = \sum_{j=0}^{T} x_j \exp^{i2\pi jt/T}, \ t \in 1..T,$$

with a wide-spectrum where T is a very large number, but the number of nonzero components x_j is less than or equal to s which is assumed to be relatively small, $s << T$. Let the signal $f(t)$ be a sum of five sinusoidal functions and noise:

$$f(t) = 10.7 \sin(2\pi \cdot 50t) + 20 \sin(2\pi \cdot 120t) + 31.5 \sin(2\pi \cdot 200t) +$$

$$+23 \sin(2\pi \cdot 300t) + 25 \sin(2\pi \cdot 450t) + error$$

for $t \in [0, 1)$.

We consider a problem where five ($s = 5$) active frequencies from the interval $[0, 500]$ and the corresponding amplitudes are unknown. We notice that a set of unknown active frequencies is not necessarily a subset of $\mathbb{N}$. According to the Nyquist/Shannon theory, having defined the possible bandwidth

$[0, 500]$, one must take readings with the frequency $2 \cdot 500 = 1000$, that is, at least $T = 1000$ readings must be taken from the interval $(0, 1)$.

The CS paradigm with a high probability ensures that information about frequencies and amplitudes will be determined from a relatively small $m \sim s \log(T/s)$ sample of values of $f(t)$. For the considered example the amount of sample is bounded by $4s \log_2(T/s)$ and can be taken approximately equal to

$$m \approx 4 \cdot 5 \cdot \log_2(1000/5)(= 153).$$

Moreover, these values need not be selected in a special manner. The restoration algorithm will work effectively almost with any set of an appropriate size. For the sake of illustration, let us assume that the measurements follow the law

$$\mathbf{y} = \mathbf{U}\mathbf{x},$$

where a random matrix with elements ± 1 is selected as $\mathbf{U}$.

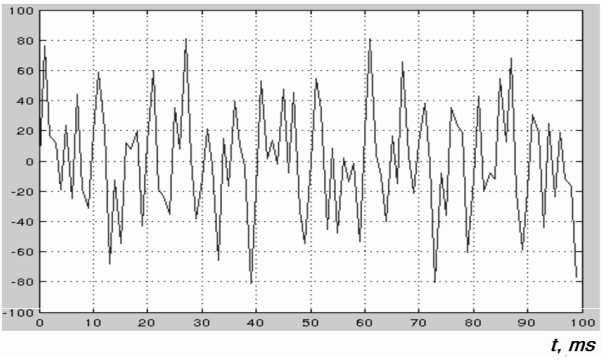

Fig. 4.6. 5-sparse noisy signal with noise

Figure 4.6 depicts an example of the signal $f(t)$ for $s = 5$ with additional random normally distributed noise having variance 0.1. Figure 4.7a shows the signal $f(t)$ in the spectral domain. The ℓ_1-regularized least squares method

$$\widehat{\mathbf{x}} = \min_{\mathbf{x}} ||\mathbf{y} - \mathbf{U}\mathbf{x}||_2^2 + \lambda ||\mathbf{x}||_1$$

was used for approximate restoration for $\lambda = 0.01$. The result of such restoration in the spectral domain is shown in Figure 4.7b. Despite the substantial reduction in the number of readings (data) and the adverse influence of noise together with the approximate nature of the algorithm, the highest-value frequencies correspond to their original counterparts.

(2) Following [98], as a second example we consider the aforementioned single-pixel compressing digital camera receiving directly m random linear measurements without preliminary acquisition of the values of all T pixels

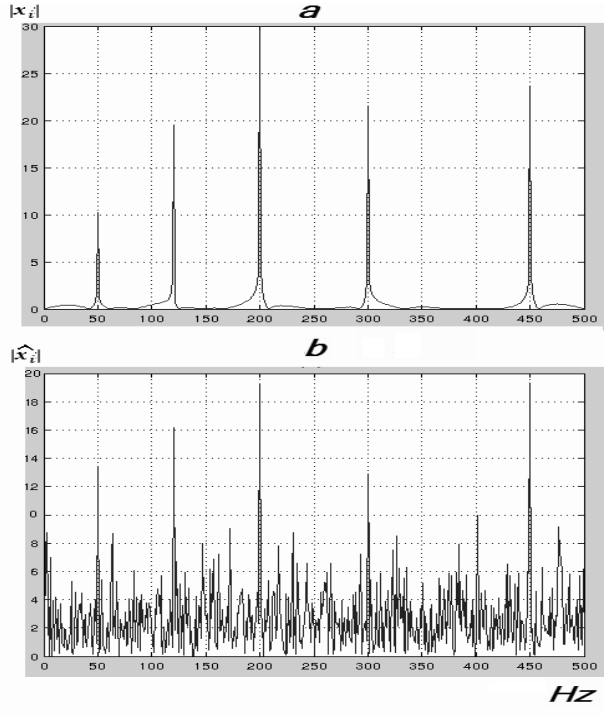

Fig. 4.7. Determination of frequencies and amplitudes of the 5-sparse signal: (a) the original and (b) restored signals in the spectral domain

of the original image (see http://dsp.rice.edu/cscamera). The light waves $\mathbf{f}$ emerging from the photograph of a person with a camera pass through the first lens and get into the array of two-position micromirrors reflecting light either in the direction of the second lens or to the side. The light flow arriving to the second lens is focused at one point of the PD (optically summed), and then this "sum" goes to the analog-digit converter (A/D) and is coded by a set of bits as one pixel, which is finally transmitted through the data channel. The process is repeated m times to obtain all components of $\mathbf{y}$,

$$\mathbf{y} = \mathbf{Hf},$$

with multiplication by matrix accomplished owing to a special physical measurement process rather than in a digital computing device. In the receiver (DSP), the measurement vector is decoded.

In addition to the requirement of lower numbers of measurements, the single-pixel camera may also reflect the waves of such lengths for which it would be difficult or expensive to construct a great array of sensors. This device may also acquire data through time intervals, thus facilitating video reconstruction.

(3) One of the first popular problems where the paradigm of compressive sensing was used to advantage was in the field of medical magnetic resonance tomography (MRT). Models for which only some of the coefficients of the two-dimensional Fourier transform that are on the rays emerging from the origin of the spectral domain are measurable were described in [220]. Obviously, such situation obeys the equation

$$\mathbf{y} = \mathbf{U}\mathbf{x} + \mathbf{v},$$

where $\mathbf{x}$ are all Fourier coefficients represented as vector and the matrix $\mathbf{U} = (\mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_m^{\mathrm{T}})^{\mathrm{T}}$ is constructed from the corresponding unit vectors $\mathbf{u}_i$. The problem lies in restoring the original image of the examined human organ.

(4) The fact that the compression algorithms are much less resource-intensive as compared to reconstruction algorithms proves to be useful when using compressive sensing for compression and restoration. This situation is exactly the opposite to the one existing, for example, in the compression of video. The operation of multiplication by a fixed matrix $\mathbf{H}$ in (4.26) is relatively simple (as compared, for example, with traditional methods of video coding) and may be used for fast coding prior to data transmission from a low-power device such as a cellular telephone to a high-performance device (server). In such cases, it is advisable to take the matrix $\mathbf{H}$ either random or problem oriented.

The existing standards of video compression define only the decoding algorithm and disregard the method of coding. The single constraint imposed on the compressing algorithm lies in that the result of compression must be decoded by a decoding algorithm. A high degree of compression with retention of good quality may require about ten runs of the coding algorithm, high computing power, and time.

In the case of transmission of video data in conference mode through communication channels, the computational complexity is not overly burdensome for the coder, but in traditional transmission compression algorithms even in the online mode the coder's computational requirements may exceed those of the decoder by a factor of five to ten.

The situation in compressive sensing is quite the opposite, allowing information to be transmitted between mobile devices of limited computing power by compressing at the mobile device, transmitting to the server, and using the server's powerful resources to restore information, compress it anew by classical algorithms, and then transmit it to another mobile device.

For example, the application of a compressive sensing concept to video compression is described in [346] through the simple multiplication by a new matrix $\mathbf{H}_k$ of each frame represented by a vector. If all frames are put one-to-one in a pile, we get a three-dimensional array $\mathbf{f}$ of numbers corresponding to the brightness of each pixel or of three-dimensional vectors denoting color. According to [346], this representation is well coded by three-dimensional wavelets because the difference between the neighbor frames is usually small.

The measurement matrix in this case will be block-diagonal:

$$\mathbf{H} = \begin{pmatrix} h_1 & 0 & \dots & 0 \\ 0 & h_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & h_l \end{pmatrix},$$

and we obtain

$$\mathbf{y} = \mathbf{H}\mathbf{f}.$$

As was already noted, in the case of video $\mathbf{f}$ admits sparse representation by the three-dimensional wavelets $\mathbf{f} = \mathbf{\Phi}\mathbf{x}$. By denoting $\mathbf{U} = \mathbf{H}\mathbf{\Phi}$, we obtain the classical problem of restoration of the sparse $\mathbf{x}$: $\mathbf{y} = \mathbf{U}\mathbf{x}$. A pile $\mathbf{f}$ of the same frames is obtained by multiplying the determined vector $\mathbf{x}$ by $\mathbf{H}$.

(5) The problem of deblurring arises, for example, in astronomy [222]. The blurred image may be regarded as the result of some blurring operator $\mathbf{H}$ (convolution of the original image with some window). It was assumed in [222] that this operator is known, but it may not be reversible. It is known, however, that the image admits a sparse representation in some basis $\mathbf{\Phi}$.

The case of an unknown blurring convolution operator and noisy measurements

$$\mathbf{y} = \mathbf{u} * \mathbf{x} + \mathbf{v}$$

was discussed in [186]. For a number of orthogonal transforms (DFT, DCT, wavelet-transform), most real-world images have sparse representation in this basis. This is one of the key ideas of using the property of sparseness in problems of high definition.

Another idea is also discussed in [186]. If some blocks are similar, it is advisable to process them jointly. If the corresponding coefficients of the orthogonal transformation of such blocks are piled together, then a 3D array results, to which the one-dimensional orthogonal transformation of the third dimension is applied. The solution is sought by minimizing the weighted sum of the error and ℓ_0-norm of the resulting vector using an IST type algorithm with $L(\mathbf{q}) = \|\mathbf{q}\|_1$.

(6) The potentialities of the CS paradigm in a machine learning are discussed in Section 6.3.

This chapter has focused mainly on discrete signals $\mathbf{f}$, but the paradigm of compressive sensing can also be applied to the s-sparse (or compressed) analog signals $f(t)$ that can be represented or approximated using only s of T possible elements from some continuous basis or dictionary $\{\mathbf{u}_j(t)\}_{j\in 1..T}$. Whereas each basic element $\mathbf{u}_j(t)$ may have a great scatter of frequencies (and, consequently, high Nyquist rate), the analog signal $f(t)$ has only s degrees of freedom and therefore may be measured at a substantially smaller number of points [330].

5

Randomized Control Strategies

For adaptive control the identification approach is often used. This approach constructs estimates for the possible values of the unknown parameters $\mathbf{x}^\star$ based on observation sequences, and these estimates are then used in a parameterized feedback loop that, if properly selected, normally assumed or established the quality of a closed-loop system that satisfies the user. In this context, we consider the problem of determining, based on the input and output data, collected at the time $t = 1, 2, \ldots, T$, a confidence region $\widehat{X}(T)$ which contains $\mathbf{x}^\star$ with a given probability, chosen by the user, and $\widehat{X}(T)$ to be built without any a priori knowledge about the distribution or correlation noise.

One of the major problems encountered in the synthesis of control laws is the insufficient *variation* of sequences of observations.

This chapter considers the possibilities of randomized controls for designing confidence regions for unknown parameters. Assumptions about external noise that affects a linear plant are reduced to a minimum: external noise can be virtually arbitrary, but independent of this noise the user must be able to add test perturbations through the input channel.

The identification method discussed below is based on the reparametrization of the mathematical ARMAX (AutoRegressive-Moving-Average model with eXogenous inputs) model of a plant. Instead of using the plant's coefficients as its initial parameters, some alternative parameters with one-to-one correspondence to the initial parameters can be conveniently used. This enables the plant to be written up in a form FIR (Finite Impulse Response) model that is not too different from a "linear observation scheme" [145].

The procedure suggested in [145, 148] works for any noise v_t and does not require a-priori knowledge of the noise characteristics. The noise can not be random or can be white or correlated, with zero-mean or bias, and the ratio signal/noise can be high or low. The recovery of unknown parameters values is provided by the properties of randomized test signals which are added together with an own adaptive control signal formed by closed-loop.

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_5,

5.1 Preliminary Example

For preliminary illustration of the main ideas we consider the example of a second-order scalar control plant described by the equation

$$y_t + a_1^\star y_{t-1} + a_2^\star y_{t-2} = b_1^\star u_{t-1} + b_2^\star u_{t-2} + v_t, \tag{5.1}$$

where $t = 1, \dots, T$ is time series, $T = 15$, $y_0 = y_{-1} = u_{-1} = 0$, $a_1^\star = -2$, $a_2^\star = 1$, $b_2^\star = 1.6$ and $b_1^\star$ is unknown. There is no information about external noises v_t, but if these noises are random they do not depend on u_t. The major difficulties of this problem lie in the non-stable properties of the considered plant (5.1).

The goal is to build, based on observations of inputs and output, the confidence interval $\widehat{X}_1(T)$, that contains $x_1^\star = b_1^\star$ with 80% probability. (Note, that such a problem with any probability up to 100% always has a trivial solution — the entire real axis.)

Moreover, the problem is to construct an algorithm of estimation $b_1^\star$ and generation of input data $u_0, \dots, u_{14}$, such that the confidence interval resulting $\widehat{X}_1(15)$ characterizes $b_1^\star$ more accurately. For example, this can means that corresponding sets $\widehat{X}_1(T)$ are gathering close to the point $b_1^\star$ as $T \to \infty$.

Let's consider the control action u_t generated by (1.12) with a deterministic part $\bar{u}_t$ and randomization Δ_t which does not depend on $v_1, v_2, \dots$.

For an asymptotic problem setting, one possible way is to choose an initial guess $\widehat{b}_1(0)$ and use an iterative algorithm like [145]

$$\widehat{b}_1(t) = \widehat{b}_1(t-1) - \frac{\gamma}{t}\Delta_t(u_t\widehat{b}_1(t-1) - \bar{y}_t), \tag{5.2}$$

where $\bar{y}_t = y_t - \psi_t$, $\psi_t = 1.6u_{t-2} + 2y_{t-1} - y_{t-2}$, $t = 1, 2, \dots$. This algorithm coincides with (1.13) when $\widehat{b}_1(0) = 0$, $\bar{u}_t \equiv 0$ and $\gamma = 1$. The confidence region can be built as early (1.14). We can formally rewrite the model (5.1) as $\bar{y}_t = b_1{}^\star u_{t-1} + v_t$. If we assume that noise v_t and the deterministic part of a control action $\bar{u}_t$ are uniformly bounded in the mean square sense $Ev_t^2 \le \sigma_v^2$, $E\bar{u}_t^2 \le c_{\bar{u}}^2$, then for any $\gamma > 0.5$ and $\mu > 0$ according to Theorem 4.2 we asymptotically have

$$E(\widehat{b}_1(t) - b_1^\star)^2 \le t^{-1}\,\frac{\gamma^2\sigma_v^2}{2\gamma - 1}(1 + \mu c_{\bar{u}}^2) \;+\; o(t^{-1}). \tag{5.3}$$

When we chose $\bar{u}_t \equiv 0$, the simulation with $b_1^\star = 1$ and i.i.d. normally distributed noise v_t with an average 0.5 (noise offset) and variance 0.1 shows that the output variable y_t increases considerably (see Figure 5.1, line 1). This is very bad from the practical point of view due to the instability of the control plant. In adaptive control problems with unknown but bounded noise the output variable can be stabilized by using the feedback with tweakable coefficients (see, for example, [145]). The same effect can be achieved in combination with some other identification methods.

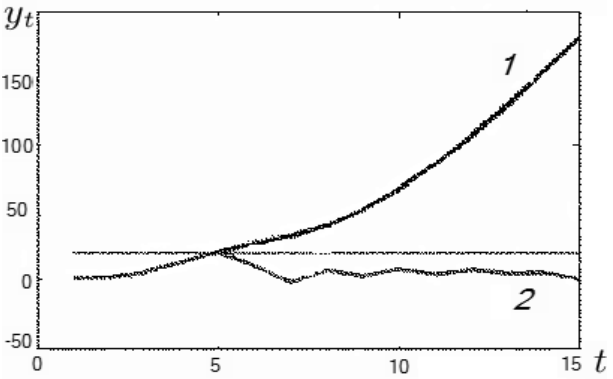

Fig. 5.1. Output variables y_t, $t \in 1..15$

For example, if we can choose sufficiently large c_v (e.g. $c_v = 10$) and suggest that $|v_t| < c_v$, the *Membership Set Approach* may be used. For each time instant t we have the bounds conditions for the possible values of an unknown parameter $b_1^\star$:

$$-c_v + \bar{y}_t < b_1^\star u_{t-1} < c_v + \bar{y}_t,$$

allowing for a sequence of estimates $\tilde{b}_1(t)$ by the "Modified Strip" algorithm [117]: $\tilde{b}_1(0) = b_1^{\pm}(0) = 0$ and for $t = 1, 2, \ldots,$

$$\tilde{b}_1(t) = \begin{cases} \tilde{b}_1(t-1), & \text{if } u_{t-1} = 0 \text{ or } \tilde{b}_1(t) \in [b_1^-(t), b_1^+(t)], \\ \frac{b_1^+(t)+b_1^-(t)}{2}, & \text{otherwise,} \end{cases}$$

$$b_1^{\pm}(t) = \begin{cases} b_1^{\pm}(t-1), & \text{if } u_{t-1} = 0 \\ \frac{\mp c_v \pm \bar{y}_t}{|u_{t-1}|}, & \text{otherwise} \end{cases}, \tag{5.4}$$

which will be used to build the stabilizing part $\bar{u}_t$ of control input added to the randomized control Δ_t (see ([145]). The stabilizing part $\bar{u}_t$ is obtained by the feedback law

$$\bar{u}_t = \begin{cases} 0, & \text{if } t = 0 \text{ or } \tilde{b}_1(t-1) = 0, \\ \bar{y}_t / \tilde{b}_1(t-1), & \text{if } |\tilde{b}_1(t-1)| > 1.6, \\ \tilde{\beta}_1(t) y_t + \tilde{\beta}_2(t) y_{t-1} - \tilde{\alpha}_1(t) u_{t-1}, & \text{othewise} \end{cases} \tag{5.5}$$

where for $t > 0$ and $|\tilde{b}_1(t-1)| < 1.6$ the adjusted coefficients $\tilde{\alpha}_1(t), \tilde{\beta}_1(t)$ and $\tilde{\beta}_2(t)$ are determined by the polynomial equation

$$(1 - 2\lambda + \lambda^2)(1 + \tilde{\alpha}_1(t)\lambda) - \lambda(\tilde{b}_1(t-1) + 1.6\lambda)(\tilde{\beta}_1(t) + \tilde{\beta}_2(t)\lambda) = 1.$$

The control plant with unknown parameter $b_1^\star$ may be non-minimal phase (non-stable upon input) when $|b_1^\star| < 1.6$, but the above feedback is stable

upon input for a plant with parameter $\tilde{b}_1(t)$ (see, e.g., [145]). Simulation with (5.5) shows the stabilized behavior of the inputs/outputs (see Figure 5.1, line 2). Figure 5.2 shows the iterative estimates $\widehat{b}_1(t)$ of the stochastic approximation algorithm (5.2) and $\tilde{b}_1(t)$ of the "Strip"-algorithm (5.4).

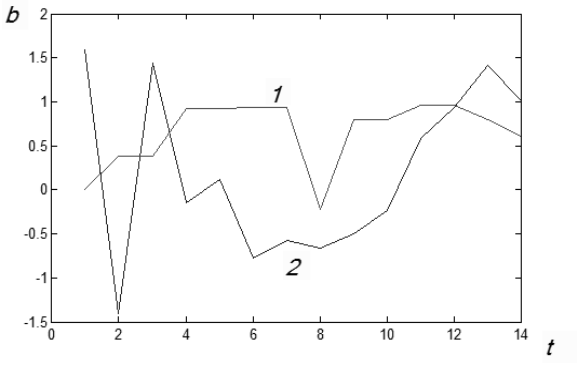

Fig. 5.2. Iterative estimates: 1 — stochastic approximation $\widehat{b}_1(t)$; 2 — "Strip" algorithm $\tilde{b}_1(t)$

The asymptotical bound (5.3) is not useful for the finite time horizon $T = 15$. But under considered control actions we can use LSCR (Leave-out Sign-dominant Correlation Regions, see [63]) for building the confidence interval $\widehat{X}_1(15)$ that contains the unknown parameter $b_1^\star$ with probability 80%.

Procedure

1. *Choose* $q = 2$ and $m = 20$ in the such way that $1 - 2q/m = 0.8$.
2. Using the observed data $u_0, \ldots, u_{14}, u_{-1}, y_{-1}, \ldots, y_{15}$, for $t \in 1..15$ we can consider the errors

$$\epsilon_t(b) = y_t - \psi_t - bu_{t-1}$$

as a linear function of b.
Denote

$$f_t(b) = \Delta_{t-1}\epsilon_t(b) = (\Delta_{t-1}(y_t - \psi_t)) - (\Delta_{t-1}u_{t-1})\, b.$$

Consider the empirical estimates of correlations $E[\Delta_{t-1}\epsilon_t(b)]$ that can be computed by using $f_t(b)$. Note that from definitions by virtue the properties $\Delta_{t-1}^2 = 1$ and $E\Delta_{t-1} = 0$ we can derive

$$E[\Delta_{t-1}\epsilon_t(b)] = (b_1^\star - b)E[\Delta_{t-1}^2] + E[\Delta_{t-1}(\bar{u}_t(b_1^\star - b) + v_t)] =$$

$$(b_1^\star - b) + E\Delta_{t-1}E(\bar{u}_t(b_1^\star - b) + v_t) = b_1^\star - b. \tag{5.6}$$

Based on this fact we *calculate* the series of empirical estimates of correlations

$$s_j(b) = \sum_{t=1}^{15} h_{j,t} \cdot f_t(b),\ j \in 1..19,$$

using different subsets of the available data. Here $h_{j,t}$ is i.i.d. Bernoulli variables, taking with equal probability $\frac{1}{2}$ values 0 and 1, i.e. $h_{j,t}$ determines whether $f_t(b)$ is used to calculate the j-th empirical correlation.

3. *Exclude* from consideration those areas of possible values of b in which empirical estimates take only positive (or only negative) values too often.

The simulation with $b_1^\star = 1$ and i.i.d. normally distributed noise v_t with an average 0.5 (noise offset) and variance 0.1 yields functions $s_j(b)$, $i = 1, 2, \ldots, 19$, and 80% confidence interval $\widehat{X}_1(15) = [0.6775, 1.1875]$, which are shown on Figure 5.3.

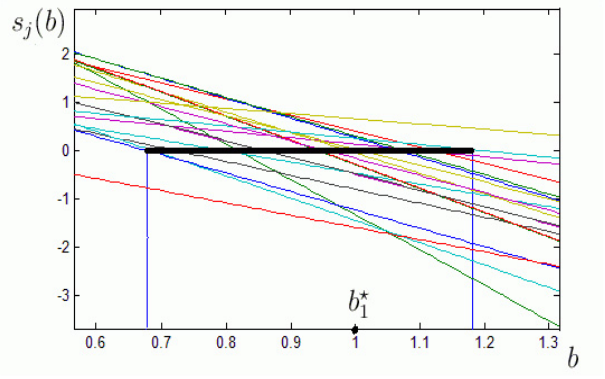

Fig. 5.3. Functions $s_j(b)$, $j \in 1..19$, and 80% confidence interval $\widehat{X}_1(15) = [0.6775, 1.1875]$ for $b^\star = 1.0$

Despite using the level of c_v in the syntheses of a stabilizing feedback, the design of confidence region obtained by the LSCR algorithm does not depend on c_v. Even with a choice of an understated level c_v the suggested algorithm still works, while the "membership set" identification methods would give the bad results.

5.2 Problem Statement

A further procedure for identifying the unknown parameters of a dynamic scalar linear control plant, which is described by the autoregressive moving average model, is based on reparameterization of the plant's mathematical model. Instead of the natural parameters of the plant — dynamic coefficients — it is convenient to use some other parameters in a one-to-one correspondence with them. Such reparameterization is the result of rewriting the plant's equation in the form of a moving average model. This facilitates using either the algorithm as a stochastic approximation or the LSCR procedure to build the confidence region even in cases when some adaptive algorithm is used in the feedback channel.

Consider a dynamic scalar linear control plant which is described in a discrete time by an ARMAX model

$$A^\star(z^{-1})y_t = B^\star(z^{-1})u_t + v_t \tag{5.7}$$

with scalar inputs u_t, outputs y_t, and noises v_t.

In Equation (5.7) z^{-1} is a delay operator: $z^{-1}y_t = y_{t-1},\ z^{-1}u_t = u_{t-1}$, the polynomials $A^\star(\lambda)$ and $B^\star(\lambda)$ take the forms

$$A^\star(\lambda) = 1 + a_1^\star\lambda + \cdots + a_{n_a}^\star\lambda^{n_a},$$

$$B^\star(\lambda) = b_l^\star\lambda^l + b_{l+1}^\star\lambda^{l+1} + \cdots + b_{n_b}^\star\lambda^{n_b},$$

where the naturals $n_a, n_b \in \mathbb{N}$ are output and input (control) model orders; l is a delay in control, $1 \le l \le n_b$, $a_1^\star, \ldots, a_{n_a}^\star, b_l^\star, \ldots, b_{n_b}^\star$ are the plant's parameters, some of which are unknown.

Noise v_t describes all other sources apart from $u_{t-l}, \ldots, u_{t-n_b}$, and y_{t-1}, $\ldots$, u_{t-n_a}, that cause variation in y_t.

Regions of reliability for unknown coefficients of the control plant (5.7) *must be defined*, with a given probability, by the observations of outputs $\{y_t\}$ and known inputs (controls) $\{u_t\}$ which can be chosen.

If we denote

$$\boldsymbol{\varphi}_t = \begin{pmatrix} -y_{t-1} \\ -y_{t-2} \\ \vdots \\ -y_{t-n_a} \\ u_{t-l} \\ u_{t-l-1} \\ \vdots \\ u_{t-n_b} \end{pmatrix}, \quad \boldsymbol{\tau}^\star = \begin{pmatrix} a_1^\star \\ a_2^\star \\ \vdots \\ a_{n_a}^\star \\ b_l^\star \\ b_{l+1}^\star \\ \vdots \\ b_{n_b}^\star \end{pmatrix}$$

then Equation (5.7) can be rewritten in the form

$$y_t = \boldsymbol{\varphi}_t^{\mathrm{T}}\boldsymbol{\tau}^\star + v_t$$

which is similar to the observation model for the linear regression case (4.1). But if $n_a > 1$ we cannot use the corresponding results from Subsection 4.1.1 since vectors $\{\boldsymbol{\varphi}_t\}$ are not independent.

5.3 Control Actions with Randomized Test Signals

Let $r \le n_a + n_b - l + 1$ be a natural, $r \in \mathbb{N}$, usually equal to the quantity of unknown plant (5.7) parameters. And let the number of observations T be equal $r \cdot K \cdot N_\Delta + r - 1$ with some naturals K and N_Δ.

Let us choose a sequence of independent random variables symmetrically distributed around zero (a randomized test perturbation) $\Delta_0, \Delta_1, \ldots, \Delta_{KN_\Delta - 1}$:

$$\text{Prob}\{\Delta_n = 0\} = 0,\ E\Delta_n^2 = \sigma_\Delta^2 < \infty,\ E\Delta_n^4 \le M_4 < \infty,$$

and add them to the input channel with some gain coefficients $\beta_k,\ k \in 1..K$ once per every r time instants (at the beginning of each time interval $rkn - l..rkn + r - l - 1$ where $n \in 1..N_\Delta$) in order to "enrich" the variety of observations.

More precisely, we will build controls $\{u_t\}$ by the rule

$$u_{rn+i-l} = \begin{cases} \bar{u}_{rn-l} + \beta_{n \div N_\Delta} \Delta_n, & i = 0, \\ \bar{u}_{rn+i-l}, & i \in 1..r-1 \text{ or } i \in -r+1..-1, \end{cases} \tag{5.8}$$

where $n \in 1..KN_\Delta$, and own (intrinsic) controls $\{\bar{u}_t\}$ are determined by the adjustable feedback

$$\bar{u}_t = \mathcal{U}_t(y_t, y_{t-1}, \ldots, \bar{u}_{t-1}, \ldots),\ \bar{u}_{-t} = 0,\ t \ge 0.$$

The type and characteristics of feedback depend on practical problems specifics. In particular, it is possible to use a trivial law of intrinsic feedback: $\bar{u}_t = 0,\ t \in 1..T - l$, or, as proposed in [145], to use the stabilized regulator

$$C(z^{-1}, \tilde{\boldsymbol{\tau}}_{t-r})\bar{u}_t = D(z^{-1}, \tilde{\boldsymbol{\tau}}_{t-r})y_t \tag{5.9}$$

with parameters $\tilde{\boldsymbol{\tau}}_t = \bar{\boldsymbol{\tau}}_{t-r}$ which are tuned by the "Strip"-algorithm

$$\bar{\boldsymbol{\tau}}_t = \bar{\boldsymbol{\tau}}_{t-1} - \begin{cases} 0, & \text{if } |\boldsymbol{\varphi}_t^{\mathrm{T}} \bar{\boldsymbol{\tau}}_{t-1} - y_t| > 2c_v + \delta\|\boldsymbol{\varphi}_t\|, \\ \frac{\boldsymbol{\varphi}_t^{\mathrm{T}} \bar{\boldsymbol{\tau}}_{t-1} - y_t}{\|\boldsymbol{\varphi}_t\|^2} \boldsymbol{\varphi}_t, & \text{otherwise}, \end{cases} \tag{5.10}$$

where c_v is an upper bound of noise level which is chosen sufficiently large, $\delta > 0$ is a small constant, and a feedback regulator (5.9) is determined by such polynomials as $C(\lambda, \tilde{\boldsymbol{\tau}})$ and $D(\lambda, \tilde{\boldsymbol{\tau}})$ so that $A(\lambda, \tilde{\boldsymbol{\tau}})C(\lambda, \tilde{\boldsymbol{\tau}}) - B(\lambda, \tilde{\boldsymbol{\tau}})D(\lambda, \tilde{\boldsymbol{\tau}})$ is a stable polynomial.

5.4 Assumptions and Model Reparameterization

The main assumption is similar to the *As1.1*:

As5.1: The user can choose Δ_n and this choice does not affect to external noises $v_1, v_2, \ldots, v_{r(n+1)-1}$. (In the mathematical sense Δ_n does not depend on $\{v_t\}_{t=1}^{r(n+1)-1}$.)

Note that we do not make any assumptions about the statistical structure of noise v_t. It can be non-random. If it is random, no assumptions are made about the zero-mean or any autocorrelation properties.

For time instants rn, $n \in 1..KN_\Delta$, we can denote $\bar{v}_n^1 = v_{rn} + (1 - A^\star(z^{-1}))y_{rn} + (B^\star(z^{-1}) - b_l^\star z^{-l})u_{rn} + b_l^\star \bar{u}_{rn-l}$ and rewrite Equation (5.7) in the following form:

$$y_{rn} = \beta_{n \div N_\Delta} \Delta_n x_1^\star + \bar{v}_n^1,$$

where $x_1^\star = b_l^\star$. This equation shows a direct relation between observation y_{rn} and test signal Δ_n which does not depend on the "new" noise $\bar{v}_n^1$.

Similarly, we can rewrite Equation (5.7) for the rest of time $rn+i-1$, $i \in 2..r$, $n \in 1..KN_\Delta$, sequentially excluding the variables $y_{rn+i-1}, \ldots, y_{rn}$ from the left-hand side of Equation (5.7) using the same equation for more early time instants

$$y_{rn+i-1} = \beta_{n \div N_\Delta} \Delta_n x_i^\star + \sum_{j=1}^{i-1} x_{i-j}^\star \bar{u}_{rn-l+j} + \bar{v}_n^i. \tag{5.11}$$

Here $x_{i-j}^\star$, $j \in 0..i-1$ are the corresponding coefficients of the remaining right-hand side terms with $\bar{u}_{rn-l+j}$, $\bar{v}_n^2 = v_{rn+1} - a_1^\star \bar{v}_n^1 + (1 - A^\star(z^{-1}) + a_1^\star z^{-1})y_{rn+1} + (B^\star(z^{-1}) - b_l^\star z^{-l} - b_{l+1}^\star z^{-l-1})u_{rn+1} + b_l^\star \bar{u}_{rn-l}$, and $\bar{v}_n^i$, $i \in 3..r$, are determined sequentially in a similar way. More precisely,

$$\bar{v}_n^i = F_i(z^{-1})v_{rn+i-1} - G_i(z^{-1})y_{rn} + F_i(z^{-1})B^\star(z^{-1})u_{rn+i-1} -$$

$$\sum_{j=1}^{i-1} x_{i-j}^\star \bar{u}_{rn-l+j} - \beta_{n \div N_\Delta} \Delta_n x_i^\star,$$

where polynomial $F_i(\lambda)$ of degree less then i and $G_i(\lambda)$ are the solution of the polynomial equation

$$F_i(\lambda)A^\star(\lambda) - \lambda^i G_i(\lambda) = 1$$

which is uniquely soluble [145]. The important feature of the rewritten Equation (5.11) is that test signal Δ_n is independent of the "new" noise $\bar{v}_n^i$, $i \in 1..r$ when Assumption *As5.1* holds.

In [145] the authors suggest forming new parameters as r-vector $\mathbf{x}^\star$ of coefficients $x_i^\star$ obtained in (5.11). They also give conditions for the invertibility of such a reparameterization procedure. The next formula immediately follows by the above definition: $\mathbf{x}^\star = \mathbf{A}^{-1}\mathbf{B}$, where $r \times r$ matrix $\mathbf{A}$ and r-vector $\mathbf{B}$:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ a_1^\star & 1 & \dots & 0 & 0 \\ a_2^\star & a_1^\star & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & a_{n_a}^\star & \dots & a_1^\star \; 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_l^\star \\ \vdots \\ b_{n_b}^\star \\ \vdots \\ 0 \end{pmatrix}.$$

How can we make the choice of an integer parameter r? The most natural way is to choose it in such a way that guarantees the existence of an inverse function corresponding to the new model parameters $\mathbf{x}^\star$.

Consider the conditions for the existence of a correspondence inverse function.

Assumption.

As5.2: Let r be a natural such that a set of control plant unknown parameters are uniquely determined by some function $\boldsymbol{\tau}(\mathbf{x})$ from the above defined vector $\mathbf{x}^\star$.

By Lemma 2.2 on p. 117 from [148] assumption *As5.2* holds for $r = n_a + n_b - l + 1$ when the plant's orders n_a, n_b are known and the following assumption is satisfied.

As5.3: The polynomials $z^{n_a} A^\star(z^{-1})$ and $z^{n_b} B^\star(z^{-1})$ are mutually prime.

In [148] there also includes an algorithm for the inverse function $\boldsymbol{\tau}(\mathbf{x})$ construction.

In practice, usually only some of plant parameters are unknown. Sometimes unknown parameters correspond to the low degrees of z^{-1} which are smaller than some $\bar{n}_a$ and $\bar{n}_b$ respectively. In such a case we can choose $r = \bar{n}_a + \bar{n}_b - l + 1$, which is significantly less than $n_a + n_b - l + 1$.

Example. Consider a second-order plant (5.1) with known coefficients $a_2^\star = 1$, $b_2^\star = 1.6$ and unknown ones $a_1^\star$ and $b_1^\star \neq 0$:

$$\boldsymbol{\tau}^\star = \begin{pmatrix} a_1^\star \\ b_1^\star \end{pmatrix}.$$

Let $r = 2$ and the "new" parameters vector $\mathbf{x}^\star$ is

$$\mathbf{x}^\star = \begin{pmatrix} b_1^\star \\ 1.6 - a_1^\star b_1^\star \end{pmatrix} \in \mathbb{R}^2.$$

In this case the inverse function $\boldsymbol{\tau}(\mathbf{x})$ is

$$\boldsymbol{\tau}(\mathbf{x}) = \begin{pmatrix} \frac{1.6 - x_2}{x_1} \\ x_1 \end{pmatrix}.$$

Equations (5.11) take the forms

$$\bar{y}_{2n} = \Delta_n x_1^\star + \tilde{v}_n^1,$$
$$\bar{y}_{2n+1} = \Delta_n x_2^\star + \tilde{v}_n^2,$$

where $\bar{y}_t = y_t - \psi_t$, $\psi_t = 1.6\bar{u}_{t-2} - y_{t-2}$, $\tilde{v}_n^1 = v_{2n} - a_1^\star y_{2n-1}$, $\tilde{v}_n^2 = v_{2n+1} + a_1^\star(a_1^\star y_{2n-1} + y_{2n-2} - 1.6\bar{u}_{2n-2} - v_{2n})$.

5.5 Stochastic Approximation Algorithm

Consider the estimation algorithm

$$\begin{cases} \widehat{\boldsymbol{\tau}}(t) = \boldsymbol{\tau}(\widehat{\mathbf{x}}(n-1)),\ t \in r(n-1)+1..rn,\ n \in 1..KN_\Delta, \\ \widehat{x}_i(n) = \widehat{x}_i(n-1) - \frac{\gamma}{n\beta_{n\div N_\Delta}}\Delta_n(u_{rn-l}\widehat{x}_i(n-1) + \\ \qquad + \sum_{j=1}^{i-1}\widehat{x}_{i-j}(n-1)\bar{u}_{rn-l+j} - y_{rn+i-1}),\ i \in 1..r. \end{cases} \tag{5.12}$$

Let the "own" control $\{\bar{u}_t\}$ be built by the feedback controller (5.9) with adjustable parameters $\widehat{\boldsymbol{\tau}}_t$ when inputs and outputs are bounded with a sufficiently large $\bar{R} > 0$, and otherwise with estimates $\bar{\boldsymbol{\tau}}_t$ which are formed by "Strip"-algorithm (5.10):

$$\tilde{\boldsymbol{\tau}}_t = \begin{cases} \widehat{\boldsymbol{\tau}}_t, & \text{if } |y_t| + |u_{t-1}| < \bar{R}, \\ \bar{\boldsymbol{\tau}}_{t-r}, & \text{otherwise.} \end{cases} \tag{5.13}$$

Theorem 5.1. If $2\gamma\sigma_\Delta^2 > 1$, $Ev_t^2 \le \sigma_v^2$, $\beta_{n\div N_\Delta} < \infty$, $\beta_{n\div N_\Delta}^2 n \to \infty$ *as* $n \to \infty$, $(\beta_{n\div N_\Delta}^2 n)^{-1} - (\beta_{(n-1)\div N_\Delta}^2(n-1))^{-1} = o(n^{-1})$, *and the conditions As5.1,5.2 hold*

then *for an arbitrary initial vector* $\widehat{\mathbf{x}}(0) \in \mathbb{R}^r$ *the algorithm* (5.12) *ensures the estimates* $\{\widehat{\mathbf{x}}(n)\}$ *such that the following limit bounds are valid in the mean square sense:*

$$E(\widehat{x}_i(n) - x_i^\star)^2 \le \frac{1}{n\beta_{n\div N_\Delta}^2}\frac{4\gamma^2\sigma_\Delta^2}{2\gamma\sigma_\Delta^2 - 1}\bar{c}_i + o(n^{-1}), \tag{5.14}$$

where $i \in 1..r$, $\bar{c}_i = |x_i^\star|^2 + if_i\sigma_v^2 + g_i n_a c_y^2 + e_i(n_b - l)c_{\bar{u}}^2$, $c_{\bar{u}} = \sup_t|\bar{u}_t|$, $c_y = \sup_t|y_t|$, f_i, g_i, e_i *are sums of squared values of coefficients of the correspondence polynomials* $F_i(\lambda)$, $G_i(\lambda)$ *and* $F_i(\lambda)B^\star(\lambda) - \sum_{j=0}^{i-1} x_{i-j}^\star\lambda^{i+l-1}$.

Remark. If $\beta_k \to 0$ as $k \to \infty$ and $c_{\bar{u}} < \infty$, it follows that a randomized test signal vanishes with time (e.g., we can choose $\beta_k = 1/\sqrt{1+\ln k}$). That is why an adaptive system can be synthesized with the described identification algorithm, since with time its output becomes indistinguishable from the output of an optimal system synthesized for known control plant parameters.

Proof. Given an arbitrary initial condition $\bar{x}(0)$ the algorithm (5.10) converges in the finite number of steps for any $\delta > 0$ (see [117], Theorem 2.1.8).

Together with stable properties of feedback (5.9) this implies that inputs $\bar{u}_t$ and outputs y_t variables are uniformly bounded.

By virtue of representation (5.11) for any $i \in 1..r$, algorithm (5.12) is a randomized stochastic approximation algorithm (4.2) for a linear regression with regressors $\beta_{n \div N_\Delta} \Delta_n$, unknown values $x_i^\star$ and noise $\bar{\bar{v}}_n^i = x_i^\star \bar{u}_{rn-l} + \sum_{j=1}^{i-1}(x_{i-j}^\star - \widehat{x}_{i-j}(n-1))\bar{u}_{rn-l+j} + \bar{v}_n^i$. For any $n = 1, 2, \ldots$ the input u_{rn-l} does not depend on $\bar{\bar{v}}_n^1, \bar{\bar{v}}_n^2, \ldots, \bar{\bar{v}}_n^r$ when Assumption *As5.1* holds.

For the considered case we have $E(\beta_{n \div N_\Delta} \Delta_n)^2 = \beta_{n \div N_\Delta}^2 \sigma_\Delta^2$, and

$$E(\bar{\bar{v}}_n^i)^2 \le |x_i^\star|^2(4 + \frac{1}{\mu^2}) + \mu^2(\sum_{j=1}^{i-1}(x_{i-j}^\star - \widehat{x}_{i-j}(n-1))\bar{u}_{rn-l+j})^2 + \nu_i \quad (5.15)$$

with arbitrary $\mu > 0$ and $\nu_i = 4(if_i\sigma_v^2 + g_i n_a c_y^2 + e_i(n_b - l)c_{\bar{u}}^2)$.

We will now utilize the method of mathematical induction to prove the inequalities (5.14) for any $i \in 1..r$.

Base of induction: $i = 1$. Inequality (5.15) takes the form $E(\bar{\bar{v}}_n^1)^2 \le 4\bar{c}_1$. This assessment allow for deriving the result of Theorem 5.1 for $i = 1$ directly from the Theorem 4.2 from Subsection 4.1.1 when $\beta_{n \div N_\Delta} = const$, which is based on Lemma 3 from [257]. For the case $\beta_{n \div N_\Delta} \to 0$ the proof is slightly different and follows by the same Lemma 3 from [257] as in the proof of Theorem 2.2 on page 146 in [148].

Inductive transition from $i-1$ to i. For the assessment $E(\bar{\bar{v}}_n^i)^2$ by virtue of (5.14) which holds for $j \in 1..i-1$ we derive

$$E(\bar{\bar{v}}_n^i)^2 \le |x_i^\star|^2(4 + 1/\mu^2) + \mu^2(\sum_{j=1}^{i-1}(x_{i-j}^\star - \widehat{x}_{i-j}(n-1))\bar{u}_{rn-l+j})^2 + \nu_i \le$$

$$\le 4|x_i^\star|^2 + \nu_i + (i-1)c_{\bar{u}}^2 \sum_{j=1}^{i-1}(x_{i-j}^\star - \widehat{x}_{i-j}(n-1))^2 =$$

$$= 4|x_i^\star|^2 + \nu_i + \mathcal{O}\left((\beta_{n \div N_\Delta}^2 n)^{-1}\right).$$

From this assessment we derive the result of Theorem 5.1 for $i > 1$ as above.

The diagram of the adaptive control system (5.7)–(5.10),(5.12) is shown in Figure 5.4 for $l = 1$.

Example. We return to the control plant (5.1) from Section 5.4 with unknown parameters $a^\star$ and $b^\star$. The algorithm (5.12) with $\beta_k \equiv 1$ takes the form

$$\begin{cases} \widehat{a}_1(t) = \frac{1.6 - \widehat{x}_2(n-1)}{\widehat{x}_1(n-1)}, \\ \widehat{b}_1(t) = \widehat{x}_1(n-1),\ t \in 2(n-1) + 1..2n, \\ \widehat{x}_1(n) = \widehat{x}_1(n-1) - \frac{\gamma}{n}\Delta_n(u_{2n-1}\widehat{x}_1(n-1) - \bar{y}_{2n}), \\ \widehat{x}_2(n) = \widehat{x}_2(n-1) - \frac{\gamma}{n}\Delta_n(u_{2n-1}\widehat{x}_2(n-1) + \widehat{x}_1(n-1)\bar{u}_{2n} - \bar{y}_{2n+1}). \end{cases}$$

with initials $\widehat{x}_1(0),\ \widehat{x}_2(0)$. The series of simulation results is presented in [145].

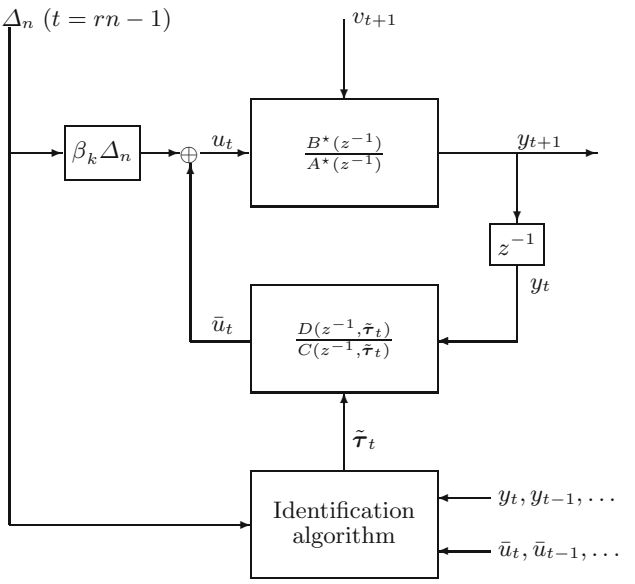

Fig. 5.4. Diagram of adaptive control system (5.7)–(5.10),(5.12) for $l = 1$

5.6 Procedure for Constructing Confidence Regions

The previous result has an asymptotic nature. For the finite number of observations we can use the following procedure.

1. For each $k \in 1..K$ consider the finite time interval $(k-1)rN_\Delta + 1..krN_\Delta$.
2. Using the observation data, for each $i \in 1..r$ we can write N_Δ predictors as a function of $\mathbf{x}_i = (x_1, x_2, \ldots, x_i)^{\mathrm{T}}$

$$\widehat{y}_{rn+i-1}(\mathbf{x}_i) = \beta_k \Delta_n x_i + \sum_{j=0}^{i-1} x_{i-j} \bar{u}_{rn+i-l-j}.$$

   Hereandafter $n \in (k-1)N_\Delta + 1..kN_\Delta$.
3. We can calculate prediction errors

$$\epsilon_n^i(\mathbf{x}_i) = y_{rn+i-1} - \widehat{y}_{rn+i-1}(\mathbf{x}_i).$$

4. According to the observed data, for each $i \in 1..r$ we form a set of N_Δ functions

$$f_n^i(\mathbf{x}_i) = \Delta_n \epsilon_n^i(\mathbf{x}_i).$$

5. Choose a natural $m > 2r$ and for $j \in 0..m-1$ construct m different binary stochastic strings (of zeros and ones) $(h_{j,1}, \ldots, h_{j,N_\Delta})$ as follows:

$h_{0,n} = 0$, $n \in 1..N_\Delta$, all the other elements $h_{j,n}$ take the values of zero or one with the equal probability $\frac{1}{2}$. Calculate

$$s^i_{k,j}(\mathbf{x}_i) = \sum_{n=(k-1)N_\Delta+1}^{kN_\Delta} h_{j,n-(k-1)N_\Delta} \cdot f^i_n(\mathbf{x}_i).$$

Note, that $s^i_{k,0}(\mathbf{x}_i) = 0$.

6. Choose q from the interval $[1, m/2r]$. Construct regions $\widehat{X}_i(k) \subset \mathbb{R}^i$ such that at least q of the $s^i_{k,j}(\mathbf{x}_i)$, $j \in 1..m-1$, functions are strictly higher than 0 and at least q functions are strictly lower than 0.
The confidence set we define by the formula

$$\widehat{X}(k) = \bigcap_{i=1}^{r} \left(\widehat{X}_i(k) \times \mathbb{R}^{r-i} \right). \tag{5.16}$$

Remarks 1. The procedure described above is similar to the one suggested in [63] but with two significant differences. First, we consider a confidence set $\widehat{X}(k)$ in state space $\mathbb{R}^r$ instead of $\mathbb{R}^{n_a+n_b}$ as in [63]. This is better since often we have $r < n_a + n_b$. Moreover, the considered confidence regions $\widehat{X}_i(k)$, $i \in 1..r$ are subsets of $\mathbb{R}^i$ instead the case of [63] when $\widehat{X}_i(k) \subset \mathbb{R}^{n_a+n_b}$. Second, randomized trial perturbations are included through the input channel only once per every r time instants instead of the permanent perturbations in [63]. This is better from the control point of view as we do not disturb the plant so often.

2. If we can separate the "new" noise $\bar{v}_{rn+i-1}$ in (5.11) into two parts — $\tilde{v}_{rn+k-1}$ and ψ_{rn+k-1} — where the first part is non-measurable but the second is determined by observable inputs and outputs with known coefficients — we can then use stronger predictors $\widehat{y}_{rn+i-1}(\mathbf{x}) = \beta_k \Delta_n x_i + \sum_{j=0}^{i-1} x_{i-j} \bar{u}_{rn+i-l-j} + \psi_{rn+i-1}$ in the above-described procedure.

The probability that $\mathbf{x}^\star$ belongs to each of $\widehat{X}_i(k)$, $i \in 1..r$, is given in the following theorem.

Theorem 5.2. *Let condition As5.1 be satisfied. Consider* $i \in 1..r$ *and assume that* $\mathrm{Prob}(s^i_{k,j}(\mathbf{x}^\star_i) = 0) = 0$. **Then**

$$\mathrm{Prob}\{\mathbf{x}^\star \in \widehat{X}_i(k)\} = 1 - 2q/m, \tag{5.17}$$

where m, q *and* $\widehat{X}_i(k)$ *from Steps 5 and 6 of the above described procedure.*

Proof. Fix $k \in 1..K$ and $i \in 1..r$.

The following preliminary Lemma 5.3, which is similar to Proposition 1 from [63], p. 2716, is instrumental to the proof of Theorem 5.2.

Lemma 5.3. *Let* $\mathbf{H}$ *be stochastic* $m \times N_\Delta$ *matrix with elements* $h_{j,n}$, $j \in 0..m-1$, $n \in 1..N_\Delta$, *from Step 5 of the above described procedure, and*

let $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{N_\Delta})^{\mathrm{T}}$ *be a vector independent of* $\mathbf{H}$ *consisting of mutually independent random variables symmetrically distributed around zero. For a given* $j \in [0, m-1]$, *let* $\mathbf{H}_j$ *be the* $m \times N_\Delta$ *matrix whose rows are equal to the* j*-th row of* $\mathbf{H}$. **Then** $\mathbf{H}\boldsymbol{\eta}$ *and* $(\mathbf{H} - \mathbf{H}_j)\boldsymbol{\eta}$ *have the same* m*-dimensional distribution provided that the* j*-th element of* $(\mathbf{H} - \mathbf{H}_j)\boldsymbol{\eta}$ *(which is* 0*) is repositioned as first element of the vector.*

The general scheme of the proof of Theorem 5.2 is the same as the corresponding proof in [63].

The difference is only on the stage when Lemma 5.3 is applied for the proof that each variable $s^i_{k,j}(\mathbf{x}^\star),\ j \in 0..m-1$ has the same probability $1/m$ to be in the generic ρ-th position (i.e., there are exactly $\rho - 1$ other variables smaller than the variable under consideration).

In our case denote $\eta_n := \Delta_n \epsilon^i_n(\mathbf{x}^\star_i)$. For the correlation between η_{n_1} and η_{n_2}, $n_1 > n_2$, we successively derive

$$E\eta_{n_1}\eta_{n_2} = E\Delta_{n_1}\epsilon^i_{n_1}(\mathbf{x}^\star_i)\Delta_{n_2}\epsilon^i_{n_2}(\mathbf{x}^\star_i) =$$

$$E\Delta_{n_1} E\epsilon^i_{n_1}(\mathbf{x}^\star_i)\Delta_{n_2}\epsilon^i_{n_2}(\mathbf{x}^\star_i) = 0$$

by virtue the assumption *As5.1* and the symmetry property of the test perturbation distribution: $E\Delta_{n_1} = 0$. Hence, variables $\eta_1, \ldots, \eta_{N_\Delta}$ are noncorrelated. The proof of their mutual independence is more complicated. As in [15, 62] it can be done carefully for the modified procedure when we use on Step 4

$$f^i_n(\mathbf{x}^i_1) = \mathrm{sign}[\Delta_n \epsilon^i_n(\mathbf{x}^i_1)],$$

where sign is defined as

$$\mathrm{sign}[x] = \begin{cases} 1, & \text{if } x > 0, \\ 1, & \text{with probability 0.5 if } x = 0, \\ -1, & \text{with probability 0.5 if } x = 0, \\ -1, & \text{if } x < 0. \end{cases}$$

Here we omit these technical details.

Pick a variable $s^i_{k,\bar{j}}(\mathbf{x}^\star_i)$ in the ρ-th position. The inequality

$$s^i_{k,j}(\mathbf{x}^\star_i) - s^i_{k,\bar{j}}(\mathbf{x}^\star_i) = \sum_{n=0}^{N_\Delta} (h_{j,n} - h_{\bar{j},n})\eta_n < 0$$

holds for $\rho - 1$ selection of $j \in [0, m-1]$. This is the same as requiring that $\rho - 1$ entries of $(\mathbf{H} - \mathbf{H}_{\bar{j}})\boldsymbol{\eta}$ be negative. From Lemma 5.3 we have $(\mathbf{H} - \mathbf{H}_{\bar{j}})\boldsymbol{\eta}$ with the same distribution as $\mathbf{H}\boldsymbol{\eta}$ and therefore Prob{"$\rho - 1$ entries of $(\mathbf{H} - \mathbf{H}_{\bar{j}})\boldsymbol{\eta}$ are negative"} = Prob{"$\rho - 1$ entries of $\mathbf{H}\boldsymbol{\eta}$ are negative"} and do not depend on $\bar{j}$.

The rest of the proof is the same as in [63].

The next corollary follows directly from Theorem 5.2.

Corollary 5.4. *Under the conditions of Theorem 5.2*

$$\mathrm{Prob}\{\mathbf{x}^\star \in \widehat{X}(k)\} \geq 1 - 2rq/m, \tag{5.18}$$

where $\widehat{X}(k)$ *from* (5.16).

Note that as pointed out in [63], the value of the probability in (5.17) is accurate, but is not the lower limit. The inequality in (5.18) is obtained due to the fact that events $\{\mathbf{x}^\star \notin \widehat{X}_i(k) \times \mathbb{R}^{r-i}\},\ i \in 1..r$ may overlap.

From the above, it is easy to derive the next theorem.

Theorem 5.5. *Let conditions As5.1,5.2 be satisfied and assume that*

$$\mathrm{Prob}(s^i_{k,j}(\mathbf{x}^\star) = 0) = 0.$$

Then *the set* $\boldsymbol{\tau}(\widehat{X}(k))$ *is the confidence set for unknown parameters of the plant* (5.7) *with a confidence level not less than* $1 - 2rq/m$.

Example. We return to the control plant (5.1) with $T = 960$ and unknown parameters $a_1^\star$ and $b_1^\star$.

Let $k = 1$, $\beta_k = 1$ and $N_\Delta = 480$. Define functions $f^i_n(\mathbf{x}_i),\ i = 1, 2,\ n \in 1..480$:

$$\begin{aligned} f^1_n(x_1) &= \Delta_n(y_{2n} - \Delta_n x_1 - x_1\bar{u}_{2n-1} - \psi_{2n}),\\ f^2_n(\mathbf{x}_2) &= \Delta_n(y_{2n+1} - \Delta_n x_2 - x_2\bar{u}_{2n-1} - x_1\bar{u}_{2n} - \psi_{2n+1}). \end{aligned}$$

Let us choose $m = 480$ and $q = 6$ and calculate empirical correlations

$$s^i_{1,j}(\mathbf{x}_i) = \sum_{n=0}^{499} h_{j,2n+i} \cdot f^i_n(\mathbf{x}_i),\ j = 1, \ldots, 479,\ i = 1, 2.$$

For $i = 1, 2$ we construct regions $\widehat{X}_i = \widehat{X}_i(1)$ which include only such values of $\mathbf{x}$ for which no less than six of the functions $s^i_{1,j}(\mathbf{x}_i),\ j = 1, \ldots, 479$ are strictly greater than zero and no less than six of them are strictly less than zero.

By Theorem 5.5, the vector of true parameters with a probability of more than $95\% = 1 - 2 \cdot 2 \cdot 6/480$ belongs to the confidence set $\boldsymbol{\tau}(\widehat{X}) = \boldsymbol{\tau}\left((\widehat{X}_1 \times \mathbb{R}) \bigcap \widehat{X}_2\right)$.

Figure 5.5 shows the confidence regions $\boldsymbol{\tau}(\widehat{X})$, $\boldsymbol{\tau}(\widehat{X}_1)$ and $\boldsymbol{\tau}(\widehat{X}_2)$ obtained from the simulation with the true values $a_1^\star = -2$ and $b_1^\star = 1$, i.i.d. Bernoulli randomization $\Delta_n = \pm 1$, and i.i.d. normally distributed noise v_t with an "unknown" average 0.5 (noise offset) and variance 0.1.

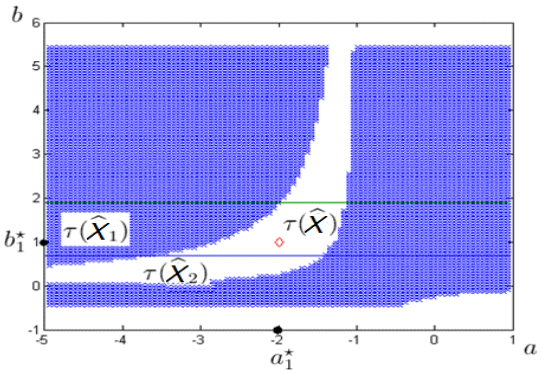

Fig. 5.5. Confidence regions $\boldsymbol{\tau}(\widehat{X})$, $\boldsymbol{\tau}(\widehat{X}_1)$ and $\boldsymbol{\tau}(\widehat{X}_2)$

5.7 Combined Procedure

Now we are ready to combine the two previous algorithms.

Algorithm:

1. For $k \in 1..K$.
2. Generate the sequence $\{\widehat{\mathbf{x}}(n)\}$ by the algorithm (5.12).
3. Choose $\varepsilon > 0$ and build the parallelepiped

$$\bar{X}(k) = \prod_{i=1}^{r} \{x_i : |\widehat{x}_i(kN_\Delta) - x_i| \le \varepsilon\}.$$

4. Choose q and m, and compute the region $\widehat{X}(k)$ using algorithm (5.16).
5. We define the confidence set as the intersection $\widetilde{X}(k) = \widehat{X}(k) \cap \bar{X}(k)$.

Theorem 5.6. *Let the conditions of Theorems 5.1 and 5.5 hold.*
Then

$$\text{Prob}\{x^\star \in \widetilde{X}(k)\} \ge (1 - 2rq/m)p_k + o\left(\frac{1}{kN_\Delta}\right), \tag{5.19}$$

$$p_k = \prod_{i=1}^{r} \left(1 - \frac{4\gamma^2\sigma_\Delta^2}{k\beta_k^2\varepsilon^2 N_\Delta(2\gamma\sigma_\Delta^2 - 1)}\bar{c}_i\right).$$

Proof. The result of Theorem 5.6 immediately follows by applying the Chebyshev's inequality to the result of Theorem 5.2 and by Corollary 5.4.

Remark. The complexity of the procedure does not increase with $t \to \infty$. Moreover, we can choose the decreasing to zero sequence β_t which allows vanishing the randomized part of control actions with time.

5.8 Randomized Control for Small UAV under Unknown Arbitrary Wind Disturbances

Consider a simplified model of a flying UAV. We assume that the model moves along a horizontal plane with a velocity of a and is pushed by a wind with a mean velocity b in direction x. Data from the GPS receiver enters the system through the time interval δ. That is, at time instant $T_t = T_0 + \delta t$ the system gets a $\bar{\mathbf{z}}_t \in \mathbf{R}^2$ (pair of coordinates) which measure the current position $\mathbf{z}_t$ with some error $\mathbf{err}_t$.

To control the UAV, i.e. to synthesize a sequence of control actions on the actuators $\{u_t\}$, it would be better to estimate the unknown parameter $x_t^\star \in \mathbb{R}$ (the wind direction) based on current measurements $\{\bar{\mathbf{z}}_i\}_{i \le t}$.

More precisely, let $\mathbf{z}_{fin} = (A, B)^{\mathrm{T}}$ be the goal point. At any time instant T_t UAV is in at point $\mathbf{z}_t$. We assume that the elevators, the ailerons, and the direction can change the UAV motion course to the angle u and keep it constant throughout the time from T_t until T_{t+1}. During this time interval the UAV motion in the direction u interferes with a wind with a constant average speed b and an angle x_{t+1}. Assuming a constant wind direction $x \equiv x_t$, a series of successive measurements makes it possible to accurately assess the wind direction. The optimal control is determined by the equation:

$$a \sin u \; - b \sin x = 0, \tag{5.20}$$

where UAV moves directly from point $\mathbf{z}_t$ It follows easy from (5.20) that

$$u = \arcsin\left(\frac{b}{a}\sin x\right). \tag{5.21}$$

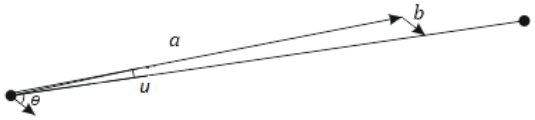

Fig. 5.6. Goal direction, course, wind shift

From a practical point of view the case $a \gg b$ is very interesting. For example, in the case of a glider discussed in [10], the cruising speed is $a \approx$ 20 m/s, and the range of wind velocities at which the glider is commonly used ranges from 0 to 7 m/s (usually 2-3 m/s). In this case we have

$$\sin u \; \approx u,$$

and the optimum angle of the course is

$$u = \frac{b}{a} \sin x = \mathcal{U}(x). \tag{5.22}$$

In the case of changes in wind directions, the wind angle variation is often assumed to be random:

$$x_{t+1} = x_t + w_{t+1}, \tag{5.23}$$

where $\{w_t\}$ are independent, zero-mean and identically distributed random variables:

$$Ew_t = 0,\ Ew_t^2 = \sigma_w^2 < \infty. \tag{5.24}$$

$$Ew_i w_j = 0 \text{ if } i \neq j.$$

In order to optimize moving to the goal point at time T_t, an estimation (prediction) algorithm *must be designed* for x_{t+1} based on observations $\bar{\mathbf{z}}_t$ and previously chosen control actions u_i, $i \in 0..t-1$, which minimizes the mean square deviation

$$E\left(x_{t+1} - \widehat{x}(t+1)\right)^2 \to \min.$$

The estimation algorithm design. Using the above theoretical results we consider the randomized approach to synthesis of control actions $\{u_t\}$ and compare it with the algorithm without randomization based on the traditional Kalman filter.

For the evaluation of emerging on-board navigation system errors, a small inertial system together with the data of GPS receiver is used.

At the beginning of each iteration, the course u_t is chosen and a zero value for the gyro is set in this direction. When the UAV is demolished in the chosen direction, the inertial system adjusts the control mechanisms on the interval $[T_t, T_{t+1}]$ so as to compensate for this effect. But at the end of the interval $[T_t, T_{t+1}]$ a discrepancy value ε is set when new GPS data enter and a new goal direction is recalculated. As a result the gyro is again adjusted, and the magnitude of this angle is taken as to be "residual" to clarify the current estimate $\widehat{x}(t+1)$ (see Figure 5.7).

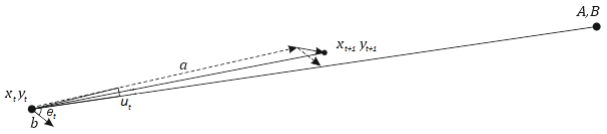

Fig. 5.7. "Residual" in the chosen course

It should be noted that if the UAV course is "adjusted" over the whole interval $[T_t, T_{t+1}]$ depending on the deviation from the desired trajectory of a gyroscope, it is not entirely correct to say that x_t is the angle of the wind direction. More precisely, x_t is the angle correcting the exposure determined by the wind.

The new observation data $\bar{\mathbf{z}}_t$ form the GPS system make it possible to calculate the "residual"

$$\varepsilon_t = a \sin u_t - b \sin x_t + v_t, \tag{5.25}$$

with some noise v_t which is determined an inaccuracy of observations $\bar{\mathbf{z}}_{t+1}, \bar{\mathbf{z}}_t$.

If we assume that $a \gg b$ and admissible u_t is small enough: $\sin u_t \approx u_t$, we then obtain the observation model

$$\varepsilon_n = a u_t - b \sin x_t + v_t, \tag{5.26}$$

suggesting the following **randomized algorithm**:

1. Choose a sequence (referred to, as earlier, randomized trial perturbation) Δ_n of independent, identically distributed (i.i.d.) random variables which are equal to $\pm\beta$ with same probabilities $\frac{1}{2}$.
2. Take control action u_t by the rule:

$$\begin{cases} u_t & = \bar{u}_{t-1} + \Delta_t, \\ \widehat{x}(t+1) & = \widehat{x}(t) - \alpha \Delta_t \varepsilon_t, \\ \bar{u}_t & = \mathcal{U}(\widehat{x}(t+1)), \end{cases} \tag{5.27}$$

where $\bar{u}_0 = 0$, $\widehat{x}(0) = 0$, $\alpha, \beta > 0$ are constants, $\mathcal{U}(\cdot)$ is determined by (5.22).

The prediction algorithm for $\widehat{x}(t+1)$ coincides with the randomized algorithm (4.20) with $\mathbf{A} = \mathbf{\Gamma} = 1$, and for the u_t syntheses, in fact, encouraged use of the strategy (5.8) with $r = l = N_\Delta = 1$ and $\mathcal{U}_t(\cdot) = \mathcal{U}(\cdot)$ from (5.22).

Simulation Results. In [10] the efficiency of algorithm (5.27) was tested by simulations in comparison with the estimates delivered by Kalman filter (see [56])

$$\begin{cases} u_t & = \bar{u}_{t-1}, \\ \widehat{x}(t+1) & = \widehat{x}(t) - K_t \varepsilon_t, \\ \bar{u}_t & = \mathcal{U}(\widehat{x}(t+1)), \end{cases} \tag{5.28}$$

$$K_t = \frac{\gamma_{t-1}}{\frac{1000}{3} + \gamma_{t-1} u_{t-1}^2},$$

$$\gamma_t = \gamma_{t-1} - \frac{u_{t-1}^2 \gamma_{t-1}^2}{\frac{16}{3} + \gamma_{t-1} u_{t-1}^2} + \frac{64}{3}, \ \gamma_0 = 0,$$

and by LMS algorithm (see [217])

$$\begin{cases} u_t & = \bar{u}_{t-1}, \\ \widehat{x}(t+1) & = \widehat{x}(t) - \alpha \varepsilon_t, \\ \bar{u}_t & = \mathcal{U}(\widehat{x}(t+1)). \end{cases} \tag{5.29}$$

Measurements in the computer simulation were carried out with an interval of time $\delta = 2$ c as an average delay time GPS module data update; the process x_t is observed in the time interval from 1 till 250, $\sigma_w = 8/\sqrt{3}$. The flight plan

is $u_0 = 0, y_0 = 0, A = 5000, B = 5000$. The initial wind angle is $x_0 = 3^0$ with respect to the initial flight course. The selected coefficients are $\alpha = 0.1$, $\beta = 0.01$.

Figures 5.8–5.10 show the comparative behavior of path trajectories corresponding to the three above-mentioned algorithms in typical cases for three different kinds of noises.

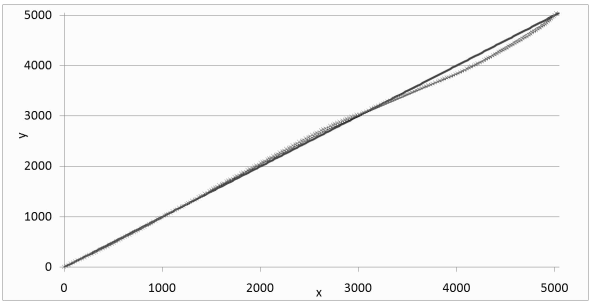

Fig. 5.8. Paths under random noises $v_t = 10 \cdot (rand() \cdot 4 - 2)$

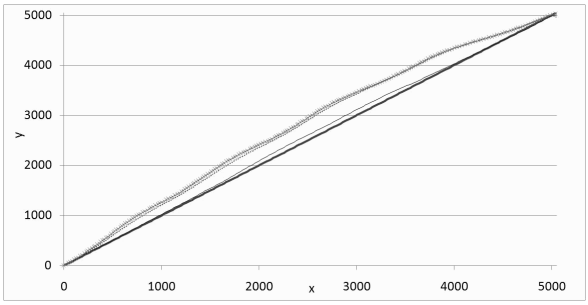

Fig. 5.9. Paths under irregular noises $v_t = 0.1 \cdot \sin(t) + 19 \cdot \text{sign}(50 - t \mod 100)$

The Kalman filter (5.28) is known to yield optimal estimates in the case of the Gaussian independent noise in the observation. Method (5.29) is sufficiently effective under zero-mean independent disturbances. This is why the behavior of the estimates generated by algorithms (5.28), (5.29) is good under zero-mean random disturbances despite a high level of noise in the observations (see Figure 5.8). In situations with an unknown constant noise or under a zero mean but not sufficiently "rich" the mean values of the errors of algorithms (5.28) and (5.29) are comparable with a squared noise level (see Figures 5.9, 5.10). At the same time, the average level of errors for estimates of the randomized algorithm (5.27) is almost the same in all three situations, and is several times better than the squared noise level. Table 5.1 shows the final results for the mean error values in typical computer experiments

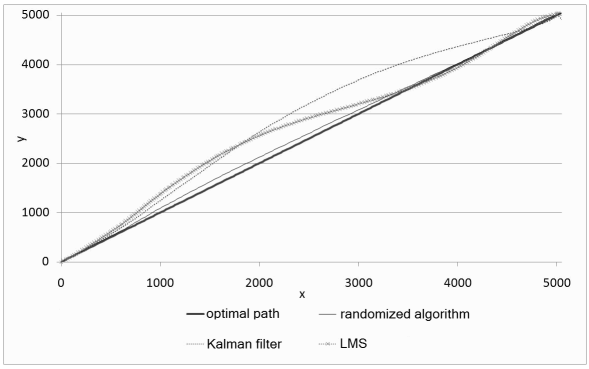

Fig. 5.10. Paths under a constant noise $v_t = 20$

Table 5.1. Results of simulations

Noise v_t	(5.27)	(5.28)	(5.29)
$10 \cdot (\text{rand}() \cdot 4 - 2)$	41.36	38.15	42.65
$0.1 \cdot \sin(\text{t}) + 19 \cdot \text{sign}(50 - \text{t} \mod 100)$	55.40	197.64	212.45
20	45.15	276.35	199.48

Part III

Data Mining

Huge amounts of data are being collected and gathered in the modern age. Data mining procedures are considered often as a phase of the well-known "Knowledge Discovery in Databases" (KDD) practice. This computational process consisting of uncovering unknown patterns in data is the issue of vigorous activity in numerous scientific areas such as artificial intelligence, machine learning and statistics. Apart of the analysis step itself, the general data mining methodology involves many additional aspects associated with databases management, data preprocessing and modeling, and visualization and so on. Clustering is a famous tool in data mining. The procedures of classifying objects consistent with their seeming resemblance are very widespread in science. Cluster Analysis encompasses algorithms and methods intended for grouping or classifying objects. Each item is presented as a set of measurements or as relationships between itself and other ones. The objective of cluster analysis or unsupervised classification, is to divide data (objects, instances, items) into groups (clusters) with the purpose that items belonging to the same group are more similar than items belonging to distinct groups. The crucial ingredient of the most of the clustering algorithms is the similarity measure constructed in order to reflect the inner data structure. Cluster analysis deals with difficult problems thus treatment of a suitable clustering technique for a given clustering task is needed. Furthermore, it is well known that no clustering method can effectively work with all kinds of cluster structures and data.

6

Clustering

The problem of classification of input signals (stimuli, objects) is a typical problem in learning theory. The classification process yields signs that characterize a group of objects as a data set or class (clusters). On these grounds, each signal can be attributed to a particular class. Clustering results in the partition of signals into groups in the environment when classes are not defined in advance. The resulting partition naturally describes the structure of a data set and can be used in the future to define it.

6.1 Partition into k Clusters

A partition (clustering)

$$\mathcal{X}^k(\mathbb{W}) = \{\mathbb{X}_1, \ldots, \mathbb{X}_k\}$$

of the set $\mathbb{W}$ is a set consisting of k non-empty clusters such that:

$$\mathbb{W} = \bigcup_{i=1}^{k} \mathbb{X}_i$$

and

$$\mathbb{X}_i \cap \mathbb{X}_j = \emptyset,\ i \neq j.$$

For a given partition $\mathcal{X}^k(\mathbb{W})$, the label function $\gamma_{\mathcal{X}^k} : \mathbb{W} \to 1..k$ assigning points to the clusters is defined as

$$\gamma_{\mathcal{X}^k}(\mathbf{w}) = i, \text{ if and only if } \mathbf{w} \in \mathbb{X}_i,\ i \in 1..k,$$

so that

$$\mathbb{X}_i = \{\mathbf{w} \in \mathbb{W} |\ \gamma_{\mathcal{X}^k}(\mathbf{w}) = i\}.$$

For any k a set $\mathbb{W}$ has many different partitions $\mathcal{X}^k(\mathbb{W})$.
What is the best partition?

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_6, © Springer-Verlag Berlin Heidelberg 2015

The following properties must be formulated mathematically: items belonging to the same group are more similar than items belonging to distinct groups. A standard approach is to associate some distortion (penalty, quality) functions q_i, defined as "closeness" to the cluster i, with each cluster (group) $i \in 1..k$, and to consider the problem of minimization

$$f(\mathcal{X}^k, \mathbf{w}) = \sum_{i=1}^{k} \gamma_{\mathcal{X}^k}(\mathbf{w}) q_i(\mathcal{X}^k, \mathbf{w}) \to \min_{\mathcal{X}^k}. \tag{6.1}$$

But the result of this function (6.1) minimization depends on $\mathbf{w}$. Let a probability distribution $\mathrm{P}(\cdot)$ be defined on a set $\mathbb{W}$. We can consider the problem of minimization of a common quality function

$$F(\mathcal{X}^k) = Ef(\mathcal{X}^k, \mathbf{w}) = \sum_{i=1}^{k} \int_{\mathbb{X}_i} q_i(\mathcal{X}^k, \mathbf{w}) \mathrm{P}(d\mathbf{w}) \to \min_{\mathcal{X}^k}. \tag{6.2}$$

In some cases, we can confine ourselves to a partition $\mathcal{X}^k$ that is fully defined by the set of k vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in \mathbb{R}^d$, which form a $d \times k$ matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$, and for $i \in 1..k$ and $\mathbf{w} \in \mathbb{W}$ each penalty functions $q_i(\cdot, \mathbf{w})$ depends on $\mathbf{x}_i$ only, i.e. $q_i(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{W} \to \mathbb{R}$. We can choose the partition rule

$$\begin{aligned} \mathbb{X}_i(\mathbf{X}) = \{\mathbf{w} \in \mathbb{W} :\ & q_i(\mathbf{x}_i, \mathbf{w}) < q_j(\mathbf{x}_j, \mathbf{w}),\ j = 1, 2, \ldots, i-1, \\ & q_i(\mathbf{x}_i, \mathbf{w}) \le q_j(\mathbf{x}_j, \mathbf{w}),\ j = i+1, i+2, \ldots, k\},\ i \in 1..k, \end{aligned}$$

which minimizes (6.1). Vectors $\mathbf{x}_i,\ i \in 1..k$, are conveniently interpreted as the *centers of clusters* when $\mathbb{W}$ is a subset of the Euclidean vector space $\mathbb{R}^d$. In this case, the partition quality functional (6.2) takes the form

$$F(\mathcal{X}^k) = \sum_{i=1}^{k} \int_{\mathbb{X}_i} q_i(\mathbf{x}_i, \mathbf{w}) \mathrm{P}(d\mathbf{w}) \to \min_{\mathcal{X}^k}, \tag{6.3}$$

and can be rewritten as follows

$$F(\mathbf{X}) = \int_{\mathbb{W}} \langle \mathbf{j}(\mathbf{X}, \mathbf{w}), \mathbf{q}(\mathbf{X}, \mathbf{w}) \rangle \, \mathrm{P}(d\mathbf{w}) \to \min_{\mathbf{X}}, \tag{6.4}$$

where $\mathbf{j}(\mathbf{X}, \mathbf{w})$ and $\mathbf{q}(\mathbf{X}, \mathbf{w})$ are k-vectors such that the first of them consists of zeros and ones values of characteristic functions $\mathbf{1}_{\mathbb{X}_i(\mathbf{X})}(\mathbf{X}, \mathbf{w})$ and the second one consists of $q_i(\mathbf{x}_i, \mathbf{w})$, $i \in 1..k$.

This formalization has a simple geometrical interpretation. Let distribution $\mathrm{P}(\cdot)$ be uniform on $\mathbb{W}$, and let the penalty functions

$$q_i(\mathbf{x}_i, \mathbf{w}) = \|\mathbf{x}_i - \mathbf{w}\|^2,\ i \in 1..k,$$

represent distances from the "centers" of clusters $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$. The integral

$$\int_{\mathbb{X}_i} \|\mathbf{x}_i - \mathbf{w}\|^2 d\mathbf{w}$$

defines the spread of the points $\mathbf{w}$ in the set $\mathbb{X}_i$. The functional (6.4) takes the form

$$F(\mathbf{X}) = \sum_{i=1}^{k} \int_{\mathbb{X}_i} \|\mathbf{x}_i - \mathbf{w}\|^2 d\mathbf{w} \to \min_{\mathbf{X}}. \tag{6.5}$$

Hence in this case, the clustering problem consists of finding a set of centers $\{\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star\}$ for which the total spread of the points is minimal.

It is easy to verify that the sets $\mathbb{X}_i$ in this example are polyhedrons and the set of centers $\{\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star\}$ minimizing the mean-risk functional (6.5) is the set of *centers of gravity* of these sets, i.e.,

$$\mathbf{x}_i^\star = \frac{\int_{\mathbb{X}_i} \mathbf{x} d\mathbf{w}}{\int_{\mathbb{X}_i} d\mathbf{w}},\ i \in 1..k.$$

These considerations are in agreement with the intuitive concept regarding the partition of a set $\mathbb{W}$ into k disjoint classes such that the centers of gravity of adjacent sets lie on a line orthogonal to the face separating the set [118].

In practice, we do not have complete information about the underlying probability distribution $\mathrm{P}(\cdot)$ on a set $\mathbb{W}$. But we usually do have a sequence of samples $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T$ that represent it. Other difficulties are that the values of a function $\bar{q}(\cdot, \cdot)$ are not always given analytically, but their values are measurable at the points $\mathbf{w}_1, \mathbf{w}_2, \ldots$ (may be with noise):

$$y_{i,n}(\mathbf{X}, \mathbf{w}_n) = q_i(\mathbf{x}_i, \mathbf{w}_n) + v_{i,n},\ i \in 1..k,\ n = 1, 2, \ldots.$$

Denote k-vectors of values $y_{i,n}(\mathbf{X})$ and $v_{i,n}$, $i \in 1..k$, $n = 1, 2, \ldots$, as $\mathbf{y}_n(\mathbf{X})$ and $\mathbf{v}_n$. We also use the notation $\mathbf{j}_n(\mathbf{X})$ for the k-vector of characteristic functions values found from noised measurements of $y_{i,n}(\mathbf{X})$, $i \in 1..k$, $n = 1, 2, \ldots$. This yields a problem of a mean-risk (6.4) minimization which is similar to (3.2) formulated in Section 3.1.

6.2 *k*-Means Clustering

k-means is one of the simplest and most widespread clustering methods. The algorithm provides a simple and understandable method for categorizing data into a fixed a priori number of groups. The key aim is to discover clusters' representatives (named centroids), one for each cluster, and to set the partition quality by means of the clusters scattered around these points. The k-means clustering seeks a partition $\mathcal{X}^k$ to minimize the within-cluster sum of squares (6.5).

An approximate solution provided by the *k-means algorithm* is outlined as follows [120]:

k-Means Algorithm:
Input:
$\mathbb{W}$ — set to be clustered;
k — number of clusters;
$\widehat{\mathcal{X}}_k(0)$ — an initial partition (optional).
Output: a partition $\mathcal{X}^k$ of the set $\mathbb{W}$ into k clusters.

1. *Initialization:* If $\widehat{\mathcal{X}}_k(0)$ is not given then construct an initial partition (items are usually randomly assigned to clusters).
2. *Minimization:* Calculate the mean (centroid) of each cluster's points.
3. *Classification:* Assign each element to the nearest current centroid.
4. Repeat Steps 2 and 3 until the partition is stable, that is, until the centroids no longer change.

The convergence proof of the k-means clustering algorithm requires proving the following two corollaries:

- Reassigning a point to a different group does not increase the error function.
- Updating the mean of a group does not increase the error function.

In practice the elements $\mathbf{w}$ are fed into the system sequentially: $\mathbf{w}_1, \mathbf{w}_2, \ldots$. For this reason iterative algorithms are much more preferable when centroids are recalculated simultaneously with the feeding of new elements. The following subsection proposes a randomized iterative algorithm and strong mathematical results regarding the convergence properties of the algorithm's estimates.

6.2.1 Randomized Iterative k-Means

Consider the problem of minimizing the mean-risk functional (6.4). To generate the estimate sequence $\left\{\widehat{\mathbf{X}}(n)\right\}$ for the optimal vectors set

$$\mathbf{X}^\star = \operatorname{argmin} F(\mathbf{X}),$$

we apply a randomized stochastic approximation type algorithm (see Section 3.2) that is based on using an observed series of random independent vectors $\boldsymbol{\Delta}_n \in \mathbb{R}^d,\ n = 1, 2, \ldots$, called the *test randomized perturbation*, consisting of independent Bernoulli random variables, equal to ± 1.

Let us take some initial set $\widehat{\mathbf{X}}(0) \in \mathbb{R}^{d\times k}$ and choose sequences of positive numbers $\{\alpha_n\}$ and $\{\beta_n\}$. Consider the following algorithm for constructing a sequence of estimates:

$$\begin{cases} \bar{\mathbf{y}}_n^{\pm} = \mathbf{y}_n(\widehat{\mathbf{X}}(n-1) \pm \beta_n \boldsymbol{\Delta}_n \mathbf{j}_n^{\mathrm{T}}(\widehat{\mathbf{X}}(n-1))), \\ \widehat{\mathbf{X}}(n) = \widehat{\mathbf{X}}(n-1) - \alpha_n \frac{\bar{\mathbf{y}}_n^{+} - \bar{\mathbf{y}}_n^{-}}{2\beta_n} \mathbf{j}_n^{\mathrm{T}}(\widehat{\mathbf{X}}(n-1)) \boldsymbol{\Delta}_n \mathbf{j}_n^{\mathrm{T}}(\widehat{\mathbf{X}}(n-1)). \end{cases} \tag{6.6}$$

Unfortunately, for $k > 1$, the properties of the estimate sequence $\{\widehat{\mathbf{X}}(n)\}$ cannot be studied directly with the general results of Section 3.3 regarding the convergence of stochastic approximation with input perturbation since the characteristic functions $\mathbf{1}_{\mathbb{X}_k(\mathbf{X}(n))}(\mathbf{w}_n)$ are not differentiable.

The following theorem is a slightly generalized version of Theorem 1 in [146].

Theorem 6.1. *Let the following conditions (assumptions) hold:*
(1) The learning sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n, \ldots$ *consists of identically distributed independent random vectors that take values in each of* k *classes in the attribute space* $\mathbb{W}$ *with a nonzero probability.*
(2) As3.1 from Section 3.1 for the functions

$$F_i(\mathbf{x}) = \int_{\mathbb{X}_i(\mathbf{X}^\star)} q_i(\mathbf{x}, \mathbf{w}) \mathrm{P}(d\mathbf{w}),\ i \in 1..k;$$

(3) As3.2,3.3 from Section 3.1 for the functions $q_i(\cdot,\cdot),\ i \in 1..k$*;*
(4) $\forall n \geq 1$ *the random vectors* $\mathbf{v}_1^{\pm}, \mathbf{v}_2^{\pm}, \ldots, \mathbf{v}_n^{\pm}$ *and* $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{n-1}$ *do not depend on* $\mathbf{w}_n$ *and* $\boldsymbol{\Delta}_n$*, and the random vector* $\mathbf{w}_n$ *does not depend on* $\boldsymbol{\Delta}_n$*;*
(5) $E\{\mathbf{v}_n\} < \infty,\ E\{\mathbf{v}_n^2\} \leq \sigma_n^2,\ |\mathbf{v}_n| \leq c_v,\ c_v > 0$*;*
(6) The clusters $\mathbb{X}_i(\mathbf{X}^\star)$*,* $i \in 1..k$*, are divided among themselves significantly in the sense that: if for some* i *from* $1..k$*,* $\mathbf{w} \in \mathbb{X}_i(\mathbf{X}^\star)$*, and for* $\mathbf{x}$ *and* $\mathbf{w}$ *the inequality*

$$|q_i(\mathbf{x}, \mathbf{w})| \leq d_{\max} = \max_{i \in 1..k} \max_{\mathbf{w} \in \mathbb{X}_i(\mathbf{X}^\star)} |q_i(\mathbf{x}_i^\star, \mathbf{w})|$$

holds, then the following inequity is satisfied:

$$|q_j(\mathbf{x}, \mathbf{w}')| > d_{\max} + 2c_v \ \ \forall \mathbf{w}' \in \mathbb{X}_j(\mathbf{X}^\star),\ j \neq i,\ j \in 1..k, \tag{6.7}$$

(7) $\sum_n \alpha_n = \infty$ *and* $\alpha_n \to 0,\ \beta_n \to 0,\ \alpha_n \beta_n^{-2} \to 0$ *as* $n \to \infty$*.*

If *the estimate sequence* $\{\widehat{\mathbf{X}}(n)\}$ *generated by algorithm* (6.6) *for some* $\widehat{\mathbf{X}}(0)$ *satisfies the relation*

$$\overline{\lim_{n\to\infty}} \left\langle \mathbf{j}_n(\widehat{\mathbf{X}}(n-1)), \mathbf{q}(\widehat{\mathbf{X}}(n-1), \mathbf{w}_n) \right\rangle \leq d_{\max} + c_v, \tag{6.8}$$

then *the estimate sequence* $\{\widehat{\mathbf{X}}(n)\}$ *converges in the mean-square sense:* $\lim_{n\to\infty} E\{\|\widehat{\mathbf{X}}(n) - \widehat{\mathbf{X}}^\star\|^2\} = 0$*, as* $n \to \infty$*, to one of the matrixes* $\widehat{\mathbf{X}}^\star$ *consisting of the vectors* $\mathbf{x}_1^\star, \mathbf{x}_2^\star, \ldots, \mathbf{x}_k^\star$*.*
Furthermore, **if** $\sum_n \alpha_n \beta_n^2 + \alpha_n^2 \beta_n^{-2} < \infty$*,*
then $\widehat{\mathbf{X}}(n) \to \widehat{\mathbf{X}}^\star$ *as* $n \to \infty$ *with probability* 1*.*

Theorem 6.1 deals with a more general case than Theorem 1 in [146] where all penalty functions $q_i(\cdot,\cdot)$ are considered to take same form. The proof of Theorem 6.1 is similar to the proof of corresponding results in [146].

The proof shows that the test perturbations need not necessarily be taken to be Bernoulli random variables. It is sufficient that their distributions are symmetric and have a bounded support. Other considerations show that, other conditions being the same, Bernoulli random variables are more effective.

Condition (6.8) in general is not preliminarily verified. If the condition is not satisfied for some estimate sequence $\{\widehat{\mathbf{X}}(n)\}$, thus does not mean that algorithm (6.6) cannot generate consistent estimates. Taking other initial values $\widehat{\mathbf{X}}(0)$, we can attempt to apply this algorithm once again.

It is easy to verify that the second and third conditions hold for functions $q_i(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|^2$. In this case, if condition (6.8) holds, then the distance between different classes must be greater than the largest radius of classes.

The observation noises $\mathbf{v}_n^{\pm}$ in Theorem 6.1 can be taken to be "almost arbitrary" since they can be nonrandom but unknown and bounded, or can be the realization of some stochastic process with arbitrary dependence structure.

We can also study subsequences of $\{\alpha_n^i\}$ and $\{\beta_n^i\}$ that vary separately for estimates of the respective class. This is convenient, particularly if representatives of different classes appear in a non-uniform manner in a learning sequence.

6.2.2 Example

We now study the application of the randomized stochastic approximation algorithm (6.6) to clustering. Let the input space contains three clusters of images which are mapped with a set of attributes into the corresponding cluster in the attribute space $\mathbb{R}^2$. Clustering in this case consists of partitioning the attribute space into three subspaces, i.e., finding three centers of these sets. For simulation, "true" classes in the attribute space were defined by the sets

$$\mathbb{X}_1 = [-1.0 \pm 1.8] \times [-1.0 \pm 1.8],\ \mathbb{X}_2 = [2.0 \pm 1, 1] \times [0.5 \pm 2.8],$$

$$\mathbb{X}_3 = [-0.5 \pm 1.3] \times [3.0 \pm 2.0].$$

Penalty functions were defined by $q_i(\mathbf{x}_i, \mathbf{w}) = \|\mathbf{x}_i - \mathbf{w}\|^2,\ i = 1, 2, 3$. Figure 6.1 shows the results obtained from a computer-aided simulation of the application of clustering algorithm (6.6) with observations recorded in the time interval $[1, 300]$. The test perturbation was defined by uniformly distributed random variables in the square $[-1, 1] \times [-1, 1]$. Estimates of the centers were equal to which are clearly close to the "true" centers. Figure 6.1 shows the rate and nature of convergence of current estimates to "true" centers.

Note that the sufficient condition (6.7) is not satisfied for this example. This numerical modeling example shows that algorithm (6.6) is quite effective even in a more general situation than that described in Theorem 6.1.

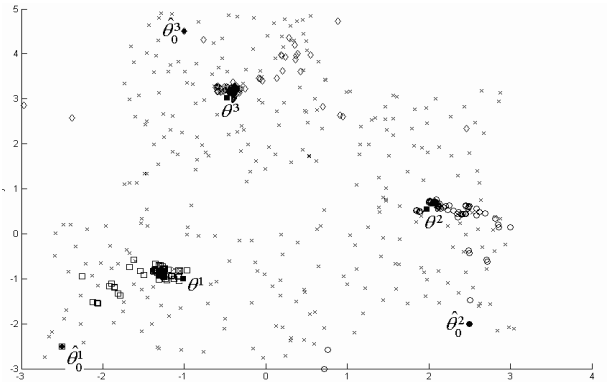

Fig. 6.1. Dataset and estimate sequences

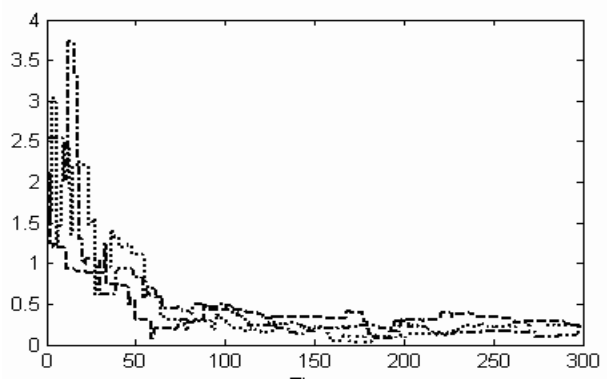

Fig. 6.2. Convergence of estimates to "true" centers

6.3 Clustering and Compressive Sensing

In this section we discuss the potentialities for machine learning of the compressive sensing paradigm described in Section 4.2.

The input data may be very large sized vectors. This may hinder the construction of a good classifier because of the great computational burden. On the one hand, this huge input data has a greater probability of leading to defects such as over fitting in cases where $\mathbf{w} \in \mathbb{W}$ classifies the examples well, but classifies all other vectors poorly, even allied vectors. Different methods can be used to reduce dimensionality. Methods based on compressive sensing turn out that in deciding to what class to assign the input signal, in the case of a priori information about signal sparseness, it is possible to do without signal restoration and confine ourselves to solving a similar problem in a space of substantially smaller dimensions.

Let us consider the problem of classification of large sized vectors $\mathbf{w} \in \mathbb{W} \subset \mathbb{R}^N$ having a sparse representation on some basis. Let us assume that these

vectors may reasonably be divided into two classes denoted by $\mathbb{X}_1$ and $\mathbb{X}_2$. This notion relies on the fact that projections of two sparse elements $\mathbf{w}', \mathbf{w}'' \in \mathbb{W}$ using a random $m \times N$ matrix $\mathbf{A}$ retain the distances between $\mathbf{w}'$ and $\mathbf{w}''$ with a higher probability. Linear separability is an important property of classification, which is retained if the distances are retained. Therefore, if there exists a hyperplane separating the classes then will be also a hyperplane separating their projections $\mathbf{y}' = \mathbf{A}\mathbf{w}'$ and $\mathbf{y}'' = \mathbf{A}\mathbf{w}''$ with a high probability. As a result, it is suggested to find a partition of the set $\mathbb{Y} \subset \mathbb{R}^m$ instead the set $\mathbb{W} \subset \mathbb{R}^N$.

An important distinction of this variant of using compressive sensing should be noted: the compressed vector *needs no restoration*, its processing takes place immediately in the compressed form.

6.4 Gaussian Mixtures and Clustering

Furthermore, for the sake of simplicity we assume that $\mathbb{W}$ is a subset of the Euclidean space $\mathbb{R}^d$ and $|\mathbb{W}| = T < \infty$. Each partition leads to the mixed decomposition of the underlying distribution $\mathrm{P}(\mathbb{W})$ of the set $\mathbb{W}$

$$\mathrm{P}(\mathbb{W}) = \sum_{i=1}^{k} p_i \mathrm{P}(\mathbb{X}_i), \tag{6.9}$$

where p_i are the cluster probabilities and $\mathrm{P}(\mathbb{X}_i)$ are the cluster distributions for all $i \in 1..k$. In order to avoid unnecessary additional conditions we furthermore assume that $p_i > 0,\ i \in 1..k$.

Clustering within the mixture model is carried out via an estimation procedure of the mixture components relying upon the given data. Conventional analytical methods are often not feasible to provide a suitable solution. Latent techniques such as the Expectation Maximization algorithm (EM) and the Gibbs sampling is traditionally used here. The cluster distributions may or may not belong to the same type. Hypothetically, they can be a distribution of the Bernoulli, the Poisson, the Normal and etc. type. The most widespread clustering version of the algorithm suggests the *Gaussian Mixture Model* (GMM) of data fitting (see, for example [23, 67, 122]):

$$f(\mathcal{X}^k, \mathbf{w}) = \sum_{i=1}^{k} p_i G(\mathbf{w}|\mathbf{x}_i, \mathbf{\Gamma}_i), \tag{6.10}$$

where $G(\mathbf{w}|\mathbf{x}, \mathbf{\Gamma})$ is the Gaussian density with the mean $\mathbf{x}$ and covariance matrix $\mathbf{\Gamma}$. Mixture models can be built-in using the EM algorithm [90], which yields maximum likelihood estimates when the data can be viewed as incomplete [105, 229, 230]. The covariance matrixes can be geometrically interpreted by means of a version of the standard spectral decomposition:

$$\mathbf{\Gamma}_i = \beta_i \mathbf{U}_i \mathbf{D}_i \mathbf{U}_i^{\mathrm{T}},\ i \in 1..k,$$

where β_i is a scalar constant representing the volume of the *i-th* covariance matrix; $\mathbf{U}_i$ is the matrix of eigenvectors determining the orientation of the *i-th* cluster; $\mathbf{D}_i = \text{diag}(\lambda_{1,i}, \ldots, \lambda_{d,i})$ is a diagonal matrix representing the shape of the the *i-th* covariance matrix such that $\lambda_{d,i} \leq \lambda_{d-1,i} \leq \ldots \leq \lambda_{1,i} = 1$. Finding the global minimum by evaluating the criteria for all possible partitions for the most part appears to be not realistic. Hence, many algorithms have been developed for determination of local minima or good suboptimal solutions. Eight basic cluster configurations have been proposed in [23] using geometrical interpretation of this decomposition.

Table 6.1. Constraints imposed on clusters by different criteria

	$\mathbf{\Gamma}_i$	Shape	Orientation	Volume
1	$\beta\mathbf{I}$	Spherical	N/A	Same
2	$\beta_i\mathbf{I}$	Spherical	N/A	Different
3	$\mathit{\Gamma}$	Same	Same	Same
4	$\beta_i\mathbf{\Gamma}$	Same	Same	Different
5	$\beta\mathbf{U}_i\mathbf{D}\mathbf{U}_i^{\mathrm{T}}$	Same	Different	Same
6	$\beta_i\mathbf{U}_i\mathbf{D}\mathbf{U}_i^{\mathrm{T}}$	Same	Different	Different
7	$\beta_i\mathbf{U}\mathbf{D}_i\mathbf{U}^{\mathrm{T}}$	Different	Same	Different
8	$\mathbf{\Gamma}_i$	Different	Different	Different

Many algorithmic approaches have been offered based on various methodologies. Of these, the maximum likelihood estimation technique is the most widespread approach to mixture recovery with the likelihood function:

$$L(\{p_i, \mathbf{x}_i, \mathbf{\Gamma}_i\}_{i\in 1..k}) = \prod_{\mathbf{w}\in\mathbb{W}} \left(\sum_{i=1}^{k} p_i G\left(\mathbf{w}|\mathbf{x}_i, \mathbf{\Gamma}_i\right) \right).$$

Dealing with the minus log-likelihood function is more appropriate:

$$F(\mathcal{X}^k) = -l(\{p_i, \mathbf{x}_i, \mathbf{\Gamma}_i\}_{i\in 1..k}) = -\sum_{\mathbf{w}\in\mathbb{W}} \ln \left(\sum_{i=1}^{k} p_j G\left(\mathbf{w}|\mathbf{x}_i, \mathbf{\Gamma}_i\right) \right), \quad (6.11)$$

This functional takes the form of (6.2).

Let $\bar{\mathbf{x}}_i$, $i \in 1..k$, be a $(1 + d + d^2)$-vectors consisting of p_i, d-vector $\mathbf{x}_i$ and elements of $d \times d$ matrix $\mathbf{\Gamma}_i$,

$$\bar{q}(\bar{\mathbf{x}}, \mathbf{w}) = -\ln p + (\mathbf{w} - \mathbf{x})^{\mathrm{T}} \mathbf{\Gamma}^{-1} (\mathbf{w} - \mathbf{x}),$$

and let $\bar{\mathbf{X}}$ be $(1 + d + d^2) \times k$ matrix consisting of vectors $\bar{\mathbf{x}}_i$, $i \in 1..k$. Based on this notation, if we choose the partition rule

$$\mathbb{X}_i(\bar{\mathbf{X}}) = \{\mathbf{w} \in \mathbb{W} : \bar{q}(\bar{\mathbf{x}}_i, \mathbf{w}) < \bar{q}(\bar{\mathbf{x}}_j, \mathbf{w}),\ j = 1, 2, \ldots, i-1,$$
$$\bar{q}(\bar{\mathbf{x}}_i, \mathbf{w}) \le \bar{q}(\bar{\mathbf{x}}_j, \mathbf{w}),\ j = i+1, i+2, \ldots, k\},\ i \in 1..k,$$

the functional (6.11) is rewritten as follows

$$F(\bar{\mathbf{X}}) = \sum_{i=1}^{k} \sum_{\mathbf{w} \in \mathbb{X}_i(\bar{\mathbf{X}})} \bar{q}(\bar{\mathbf{x}}_i, \mathbf{w}) \to \min_{\bar{\mathbf{X}}}. \tag{6.12}$$

Actually, this is a "k-means" standpoint regarding the problem. As mentioned earlier, there is no known method that provides a closed-form analytical solution maximizing log likelihood function (6.11). The most popular approach to this problem is the EM algorithm, first proposed by Dempster, Laird and Rubin [90]. The EM algorithm is an iterative optimization procedure such that for each EM-step the likelihood does not decrease, thus assuring convergence to no less than a local maximum of the likelihood. In EM clustering, a $k \times T$ matrix $\mathbf{Z}$ is introduced, such that, for each item $\mathbf{w}_t \in \mathbb{W},\ t \in 1..T$ the *t-th* vector of the latent variables $\mathbf{z}_n = (z_{1,t}, \ldots, z_{k,t})^{\mathrm{T}}$ presents the probabilities that $\mathbf{w}_t$ is found in each of the k clusters.

EM **Algorithm:**

Input:

$\mathbb{W}$ — set to be clustered (with cardinality T);

k — number of clusters;

$\widehat{\mathcal{X}}_k(0)$ — initial partition (optional).

Output: a partition $\mathcal{X}^k$ of the set $\mathbb{W}$ into k clusters.

1. *Initialization.* Initialize the mixture model parameters, thus creating a current model. The matrix $\mathbf{Z}_0$ can be initialized by randomly assigning the value 1 to one of elements in each row. In the case where $\widehat{\mathcal{X}}_k(0)$ is given, a current model is defined by this partition.
2. **M-STEP.** Recalculate model posterior parameters, obtained from the initialization step or from the $E - STEP$, as follows:

   2a. the sample clusters sizes

$$\widehat{S}_i = \sum_{t=1}^{T} z_{i,t}, \quad i = 1, \ldots, k.$$

   2b. the sample clusters probabilities

$$\widehat{p}_i = \frac{\widehat{S}_i}{T}, \quad i = 1, \ldots, k.$$

   2c. the sample clusters means

$$\widehat{\mathbf{x}}_i = \frac{1}{\widehat{S}_i} \sum_{t=1}^{T} z_{i,t} \mathbf{w}_t, \quad i = 1, \ldots, k.$$

   2d. The sample covariance matrix $\widehat{\mathbf{\Gamma}}_i$ is calculated for each cluster.

3. **E-STEP.** Calculate posterior probabilities by the Bayesian rule:

$$z_{i,t} = \frac{\widehat{p}_i G\left(\mathbf{w}_t | \widehat{\mathbf{x}}_i, \widehat{\mathbf{\Gamma}}_i\right)}{\sum_{i=1}^{k} \widehat{p}_i G\left(\mathbf{w}_t | \widehat{\mathbf{x}}_i, \widehat{\mathbf{\Gamma}}_i\right)}, \quad i = 1, \ldots, k.$$

4. *Convergence criteria testing.* Terminate, if current and new models are sufficiently close; otherwise, repeat from Step 2.

Hence, the algorithm is executed by iteratively alternating between the E-step and the M-step. The E-step calculates the conditional expectation of the latent matrix $\mathbf{Z}$ estimated conditionally on the recent configuration. The M-step maximizes this conditional expectation of the complete data log-likelihood function over the entire parameter space. The *Classification EM (CEM) algorithm* is an iterative clustering algorithm constructed to concurrently provide the parameter estimation and the item classification. The classification criterion (minus likelihood) is:

$$cl(\bar{\mathbf{X}}) = \sum_{t=1}^{T} \sum_{i=1}^{k} z_{j,t} \bar{q}(\bar{\mathbf{x}}_i, \mathbf{w}_t) \tag{6.13}$$

with respect to the latent component membership $\mathbf{Z}$ and the parameters $\bar{\mathbf{x}}_i,\ i \in 1..k$, consisting of $p_i, \mathbf{x}_i, \mathbf{\Gamma}_i$. As was shown by Celeux and Govaert [67], this algorithm is a classification version of the EM algorithm. A classification step between the E-step and the M-step is included so that the classification likelihood criterion value increases at each iteration, and the process converges in a finite number of iterations. In the case of identical proportions and covariance matrices equal to the identity matrix (spherical covariance matrices, $\mathbf{\Gamma}_i = \sigma^2 \mathbf{I},\ i \in 1..k$), the CEM algorithm matches up with the k-means algorithm from Section 6.2. As a result, the CEM algorithm appears to be a generalization of the k-means algorithm capable of operating with non-spherical clusters and non-uniform proportions. Clusters having the same size and geometrical structure are rarely presented in real datasets. However, this assumption is widespread and does not seem to deteriorate the analysis results.

As mentioned earlier the k-means algorithm is a special case of the general GMM clustering algorithm. The advantages and disadvantages of this model can be summarized as follows.

Advantages of GMM:

- Flexibility on cluster covariance structure.
- Rigorous statistical inference with full model.
- k-means is often fast.

Disadvantages of GMM:

- Model selection is usually difficult. Data may not fit Gaussian model.
- Too many parameters to estimate in complex covariance structure.
- Local minimum problem.
- Works poorly on non-convex clusters.

6.5 Information Methods

6.5.1 Mixture Clustering Model — An Information Standpoint

In this subsection we discuss the mixture clustering model mentioned above from the information theory point of view (see, [276, 306]). The main assumption of the model is that each item in $\mathbb{W}$ belongs to each cluster with a certain probability. Here a clustering solution is given by the set of probability distributions for associating points to the clusters. This association is termed "fuzzy membership in clusters". A particular case in which each point belongs to one cluster, corresponds to hard clustering. Determining an optimal association distribution is the aim of the clustering approaches. From the information theory point of view, clustering is the basic strategy for data loss compression. According to this methodology, the data is divided into groups described in terms of the bit rate using a representative for each group.

For two discrete random variables X and Y taking the values $\{x_i\}$, $i = 1, 2, ...$ and $\{y_j\}$, $j = 1, 2, ...$, correspondingly, we introduce the mutual information:

$$I(X,Y) = \sum_{i,j} p(x_i, y_j) \log_2 \frac{p(x_i, y_j)}{p(x_i)p(y_j)} = \sum_{i,j} p(x_i, y_j) \log_2 \frac{p(y_i|x_i)}{p(y_j)},$$

where

- $p(x_i, y_j)$ is the joint probability function of X and Y respectively;
- $p(x_i)$ and $p(y_j)$ are the marginal probability functions of X and Y;
- $p(y_i|x_i)$ is the conditional probability function of Y on X.

A clustering procedure is intended to compress the initial data by discarding the insignificant information. The relevant information is identified with the help of a distortion function, which usually measures a similarity among the data items. Let us suppose that the distortion function in each cluster is $q_i(\mathbf{w}', \mathbf{w}'')$, $i \in 1..k$. A lossy compression can be generated by assigning the data to clusters so that the mutual information

$$I(\mathcal{X}^k, \mathbb{W}) = \sum_{\mathbf{w},i} P(\mathbb{X}_i|\mathbf{w})P(\mathbf{w}) \log_2 \frac{P(\mathbb{X}_i|\mathbf{w})}{P(\mathbb{X}_i)} \tag{6.14}$$

is minimized. The minimization is constrained by fixing the expected distortion

$$F(\mathcal{X}^k, \mathbb{W}) = \sum_{\mathbf{w},i} P(\mathbb{X}_i|\mathbf{w})P(\mathbf{w})q_i(\mathbf{x}_i, \mathbf{w}),$$

where $\mathbf{x}_i$ is the representative (centroid) of the cluster $\mathbb{X}_i$. The formal solution of this task is provided by the Boltzmann distribution

$$P(\mathbb{X}_i|\mathbf{w}) = \frac{P(\mathbb{X}_i)}{Z(\mathbf{w}, C)} \exp\left(-\frac{q_i(\mathbf{x}_i, \mathbf{w})}{C}\right),$$

where

$$Z(\mathbf{w}, C) = \sum_i P(\mathbb{X}_i) \exp\left(-\frac{q_i(\mathbf{x}_i, \mathbf{w})}{C}\right)$$

is the normalization constant and C is the Lagrangian multiplier. In the hard clustering case, a partition $\mathcal{X}^k$ is defined by a set of the associations

$$\nu(\mathbf{w}, \mathbb{X}_i) = \begin{cases} 1, & \text{if } \mathbf{w} \in \mathbb{X}_i \\ 0, & \text{otherwise} \end{cases}.$$

Thus, the underlying distribution of $\mathbb{W}$ is given by (6.9) and the marginal distribution of cluster centroids is equal to:

$$P(\mathbf{X}) = \frac{e^{-\frac{F}{C}}}{\sum\limits_Y e^{-\frac{F}{C}}}$$

where

$$F = -C \sum_{\mathbf{w}} \ln\left(\sum_i \exp\left(-\frac{q_i(\mathbf{x}_i, \mathbf{w})}{C}\right)\right)$$

is the so called "free energy" of the partition. It is evident that the most probable values of the centroids minimize F. Therefore, optimal estimates of the cluster parameters can indeed be achieved by minimizing the free energy. An important example of this construction is a partition corresponding to the distortions

$$q_i(\mathbf{x}_i, \mathbf{w}) = (\mathbf{w} - \mathbf{x}_i)\mathbf{\Gamma}_i^{-1}(\mathbf{w} - \mathbf{x}_i),$$

where $\mathbf{x}_i$ is the centroid of the cluster $\mathbb{X}_i$ and $\mathbf{\Gamma}_i$ is its covariance matrix. In this case, the expression for F can be rewritten as

$$F = -C \sum_{\mathbf{w}} \ln\left(\sum_i \exp\left(-\frac{(\mathbf{w} - \mathbf{x}_i)\mathbf{\Gamma}_i^{-1}(\mathbf{w} - \mathbf{x}_i)}{C}\right)\right).$$

If we fix the matrices $\mathbf{\Gamma}_i$ then from the equations

$$\frac{\partial F}{\partial \mathbf{x}_i} = 0,\ i \in 1..k,$$

we obtain the result

$$\mathbf{x}_i = \frac{\sum\limits_{\mathbf{w}} \mathbf{w} P(\mathbb{X}_j|\mathbf{w})}{\sum\limits_{\mathbf{w}} P(\mathbb{X}_j|\mathbf{w})},\ i \in 1..k.$$

Note the similarity between the current result and the maximum-likelihood estimation of the normal mixtures parameters provided by means of the *EM* algorithm (see, Section 6.4). The essential point is that in the free energy optimization approach no prior information about the data distribution is required. The cluster distributions are derived directly from the corresponding Boltzmann and Gibbs distributions and the suitable optimization task.

6.5.2 Information Bottleneck

The *Information bottleneck method* [317] is a clustering technique according to which clusters have to reflect only the "relevant" information within the data. According to this approach, each element of $\mathbb{W} \subset \mathbb{R}^d$ is identified with a distribution over its features, and the partitions are created based on the features' empirical conditional distributions:

$$p(w_i|\mathbf{w}) = \frac{w_i}{\sum_{i=1}^{d} w_i}.$$

Co-occurrence data such as occurrences of verbs and direct objects in sentences [317], words and documents ([22, 167] and [295]), and tissues and gene expression patterns ([318]) may be organized according this principle. For instance, in the text clustering, $\mathbb{W}$ can be a set of documents and $\mathbb{Y}$ can be a set of words. Each element of $\mathbb{W}$ is represented by conditional distributions of words:

$$p(y|\mathbf{w}) = \frac{n(y|\mathbf{w})}{\sum\limits_{y_i \in \mathbb{Y}} n(y_i|\mathbf{w})},$$

where $n(y|\mathbf{w})$ is the total occurrence of the word y in the document $\mathbf{w}$. To finalize the model a uniform prior distribution of the documents is assumed:

$$p(\mathbf{w}) = \frac{1}{|\mathbb{W}|}.$$

It is natural to suppose that documents that have close term conditional distributions belong to the same cluster. This notion can lead to a cluster hierarchy structure based on the likeness of conditional distributions of words. Generally, any cluster method applying this probability interpretation would have to use a probability metric between distributions. A cluster hierarchy based on the similarity of conditional distributions was first introduced in [254].

Tishby, Pereira, and Bialek [317] proposed a method that avoids the arbitrary choice of a distortion or a distance measure. Given the joint distribution $p(X;Y)$, the method seeks a compact representation of X, which retains as much information as possible about the applicable variable Y. The mutual information between the random variables X and Y (see, (6.14)) is known to represent the natural statistical measure of the information that variable X holds about variable Y. A compressed representation S of X (a clustering of X in our case) is defined by $p(S|X)$. Hence, the quality of the clusters is calculated by the information covered by Y, namely, by $I(S;Y)/I(X;Y)$.

The information bottleneck principle determines the distortion measure between the points by means of the known Kullback-Leibler divergence $D_{KL}(p(y|x)||p(y|s))$ among the conditional distributions $p(y|x)$ and $p(y|s)$. Generally, the membership probabilities, $p(s|x)$, are "soft", i.e., each element can be assigned to every cluster with some (normalized) probability. In the hard clustering case the agglomerative information bottleneck algorithm proposed in [294, 295] uses the merging criterion based on the distortion:

$$DJS(x,s) = (p(x) + p(s)) * D_{JS}(p(y,x), p(y,s)),$$

where $D_{JS}(p,q)$ is the Jensen-Shannon divergence have identical weights.

6.6 Spectral Clustering

Let us suppose that there is a set $\mathbb{W}$ provided with a similarity function $q(\cdot,\cdot)$ defining the points' likeness. As discussed earlier, the purpose of clustering is to partition the data into groups such that points within the same group are more similar that are points belonging to different groups. In a case where this similarity is not quantified, we can represent the data in the form of a similarity graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where each vertex in the graph symbolizes a data point and two vertices are connected if and only if they are similar to one another. The source clustering task is thus transformed into a problem connected to graph clustering with the aim of dividing a graph into groups such that the edges between different groups have very low weights and the edges within a group have high weights. One of the most acceptable methods used for this purpose is spectral clustering methodology.

The basic stages of this approach include:

- Pre-processing: Construct a matrix representation of the dataset.
- Decomposition: Compute eigenvalues and eigenvectors of the matrix.
- Mapping: Map each point to a lower-dimensional representation based on one or more eigenvectors.
- Classification: Assign points to clusters, based on the new representation.

Spectral clustering skills commonly leverage the spectrum of a given similarity matrix in order to carry out dimensionality reduction for clustering in fewer dimensions. Note, that there is a large family of possible algorithms

based on spectral clustering methodology (see, for example [112, 221, 242]) that employ the eigenvectors differently. Researchers disagree regarding which eigenvectors to use and how to derive clusters from these eigenvectors.

Graph Laplacian matrices are the most important instrument used in the spectral clustering field. The Laplacian matrix, sometimes called the admittance matrix or the Kirchhoff matrix, is defined for a simple graph $\mathbb{G}$ (an undirected graph that has no loops and no more than one edge between any two different vertices) with T vertices as (see, e.g. [233]):

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

where $\mathbf{A}$ is the adjacency matrix of the graph and $\mathbf{D}$ is the degree matrix which is a diagonal matrix with the degrees of all the graph vertices on the diagonal. The main properties of this Laplacian matrix can be summarized as follows:

- $\mathbf{L}$ is symmetric positive-semidefinite.
- The smallest eigenvalue λ_0 of $\mathbf{L}$ is 0, corresponding to the eigenvector $\mathbf{v}_0 = (1, 1, \ldots, 1)^{\mathrm{T}}$.
- The multiplicity of λ_0 equals the number of connected components in the graph;

The symmetric normalized Laplacian defined as

$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \tag{6.15}$$

has spectral properties similar to those of $\mathbf{L}$. Spectral clustering methods are closely related to the relaxation of several graph clustering problems. The graph partitioning problem generally seeks to divide the graph into disjoint sub-graphs such that the edges between the different groups have a very low weight and the edges within a group have high weight. Solving the mincut problem appears to be the most natural way to cluster a graph.

Let us denote $links(\mathbb{B}, \mathbb{C})$ to be the sum of the edge weights between nodes in two subsets $\mathbb{B}$, $\mathbb{C}$ in $\mathbb{G}$. In other words,

$$links(\mathbb{B}, \mathbb{C}) = \sum_{i \in \mathbb{B}, j \in \mathbb{C}} a_{ij}.$$

The mincut approach seeks a partition $\mathcal{X}^k(\mathbb{G})$ that minimizes

$$cut(\mathcal{X}^k(\mathbb{G})) \ : \ = \frac{1}{2} \sum_{i=1}^{k} links(\mathbb{X}_i(\mathbb{G}), \overline{\mathbb{X}_i(\mathbb{G})}).$$

The factor $1/2$ is used for notational consistency in order to circumvent double counting of the edges. For $k = 2$, mincut is a relatively simple and efficiently solved problem (see, [307]). However, a minimization of this function can lead to unbalanced clusters. To avoid such a situation requires "forcing" the clusters to be "practically large" using an appropriate normalization.

Ratio Association. The ratio association (also called average association) objective [291] makes it possible to maximize within-cluster association normalized by the size of the cluster:

$$RAssoc(\mathbb{G}) = \underset{\mathcal{X}^k(\mathbb{G})}{max} \frac{1}{2} \sum_{i=1}^{k} \frac{links(\mathbb{X}_i(\mathbb{G}), \overline{\mathbb{X}_i(\mathbb{G})})}{|\mathbb{X}_i(\mathbb{G})|}.$$

Ratio Cut. Here, (see, [71]) the cut between clusters and the remaining vertices is minimized:

$$Rcut(\mathbb{G}) = \min_{\mathcal{X}^k(\mathbb{G})} \frac{1}{2} \sum_{i=1}^{k} \frac{links(\mathbb{X}_i(\mathbb{G}), \overline{\mathbb{X}_i(\mathbb{G})})}{|\mathbb{X}_i(\mathbb{G})|}.$$

Normalized Cut. The normalized cut [291, 356] is intended to minimize the cut relative to the total degree of the inner cluster vertices:

$$Ncut(\mathbb{G}) = \min_{\mathcal{X}^k(\mathbb{G})} \frac{1}{2} \sum_{i=1}^{k} \frac{links(\mathbb{X}_i(\mathbb{G}), \overline{\mathbb{X}_i(\mathbb{G})})}{\deg(\mathbb{X}_i(\mathbb{G}))}.$$

Spectral clustering is inherently constructed to resolve relaxed versions of presented problems. The Ratio Cut minimization problem can relaxed as follows. Given a partition $\mathcal{X}^k(\mathbb{G})$, let us introduce the matrix $\mathbf{H} \in \mathbb{R}^{T \times k}$, $(T = |\mathbb{G}|)$:

$$h_{j,i} = \begin{cases} \frac{1}{\sqrt{|\mathbb{X}_i(\mathbb{G})|}}, & \text{if } \nu_j \in \mathbb{X}_i(\mathbb{G}) \\ 0, & \text{otherwise} \end{cases}. \tag{6.16}$$

It is easy to check that

$$\mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I} \tag{6.17}$$

and

$$h_{\cdot,i}^{\mathrm{T}} \mathbf{L} h_{\cdot,i} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{L}\mathbf{H}\right)_{i,i} = \frac{links(\mathbb{X}_i(\mathbb{G}), \overline{\mathbb{X}_i(\mathbb{G})})}{|\mathbb{X}_i(\mathbb{G})|}.$$

Thus

$$RAssoc(\mathbb{G}) = \mathrm{Tr}\left[\mathbf{H}^{\mathrm{T}}\mathbf{L}\mathbf{H}\right],$$

and the optimization problem is transformed to

$$\min_{\mathcal{X}^k(\mathbb{G})} \mathrm{Tr}\left[\mathbf{H}^{\mathrm{T}}\mathbf{L}\mathbf{H}\right], \quad s.t. \text{ (6.16) and (6.17)}.$$

By allowing the entries of the matrix $\mathbf{H}$ to take arbitrary real values we obtain the following relaxed problem

$$\min_{\mathcal{X}^k(\mathbb{G})} \mathrm{Tr}\left[\mathbf{H}^{T}\mathbf{L}\mathbf{H}\right], \quad s.t.\ \mathbf{H}^{\mathrm{T}}\mathbf{H} = \mathbf{I}.$$

According to the Rayleigh-Ritz theorem (see, e.g. [305]), $\mathbf{H}$ is the matrix that contains the first k eigenvectors of $\mathbf{L}$ as columns, which are actually involved in most spectral clustering methods.

Introducing

$$h_{j,i} = \begin{cases} \frac{1}{\sqrt{\deg(\mathbb{X}_i(\mathbb{G}))}}, & \text{if } \nu_j \in \mathbb{X}_i(\mathbb{G}) \\ 0 & \text{otherwise} \end{cases}$$

we similarly obtain that the $Ncut$ problem is altered to

$$\min_{\mathcal{X}^k(\mathbb{G})} \text{Tr}\left[\mathbf{H}^{\text{T}}\mathbf{L}\mathbf{H}\right], \ s.t. \ \mathbf{H}^{\text{T}}\mathbf{D}\mathbf{H} = \mathbf{I}.$$

Substituting $\mathbf{S} = \mathbf{D}^{\frac{1}{2}}\mathbf{H}$ and allowing the entries of the matrix $\mathbf{H}$ to take arbitrary real values yields the following relaxed problem

$$\min_{\mathcal{X}^k(\mathbb{G})} \text{Tr}\left[\mathbf{S}^{\text{T}}\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\right], \ s.t. \ \mathbf{S}^{\text{T}}\mathbf{S} = \mathbf{I}.$$

According to the Rayleigh-Ritz theorem mentioned earlier, the matrix $\mathbf{S}$ providing the solution contains the first k eigenvectors of $\mathbf{L}_{sym}$ (6.15) as columns $\mathbf{H}$ consists of the first k generalized eigenvectors of:

$$\mathbf{L}\mathbf{x} = \lambda\mathbf{D}\mathbf{x}.$$

The *Ng-Jordan-Weiss (NJW) Algorithm*, which is based on this relaxation approach, was proposed in [248].

6.6.1 The Ng-Jordan-Weiss (NJW) Algorithm

Algorithm 1. ***Spectral clustering*** $(\mathbb{W}, k, \sigma)$ (NJW):
Input:
$\mathbb{W}$ — the data to be clustered ($\mathbb{W} \subset \mathbb{R}^d$);
k — number of clusters;
σ — the scaling parameter.
Output: a partition $\mathcal{X}_\sigma^k$ of $\mathbb{W}$ into k clusters depending on σ.
Algorithm:

1. Construct the affinity matrix $\mathbf{A}(\sigma^2)$

$$a_{i,j}(\sigma^2) = \begin{cases} \exp\left(\frac{-\|w_i - w_j\|^2}{2\sigma^2}\right), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}.$$

2. Compute $\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}(\sigma^2)\mathbf{D}^{-\frac{1}{2}}$.
3. Compute $\lambda_1, \lambda_2, \ldots, \lambda_k$, the k largest eigenvectors of $\mathbf{L}_{sym}$ (chosen to be orthogonal to each other in the case of repeated eigenvalues).
4. Create the matrix $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_k\} \in \mathbb{R}^{T \times k}$ by joining the eigenvectors as consequent columns.
5. Compute the matrix $\mathbf{Y}$ from $\mathbf{\Lambda}$ by normalizing each of $\mathbf{\Lambda}$'s rows to have the unit length.

6. Cluster the set $\mathbb{Y}$ of rows of $\mathbf{Y}$ into k clusters via the k-means or any other algorithm (that attempts to minimize a distortion) to obtain a partition $\mathcal{X}_\sigma^k(\mathbb{Y})$.
7. Assign each point $\mathbf{w}_i$ according to the cluster that was tagged to the i-th row in the obtained partition.

Note that there is a one-to-one correspondence between the partitions $\mathcal{X}_\sigma^k(\mathbb{W})$ and $\mathcal{X}_\sigma^k(\mathbb{Y})$. The magnitude parameter σ represents the increasing rate of the affinity of the distance function. How does the choice of the value of σ affect the results? What would be the optimal choice for σ? This parameter plays a very important role in the clustering process and can be naturally achieved as the outcome of an optimization problem intended to find the best possible partition configuration. An appropriate meta algorithm could be presented in the following form:

Algorithm 2. ***Self-Learning Spectral Clustering*** $(\mathbb{W}, k, F)$:

Input:
$\mathbb{W}$ — the data to be clustered ($\mathbb{W} \subset \mathbb{R}^d$);
k — number of clusters;
F — cluster quality function to be minimized.
Output:
$\sigma^\star$ — an optimal value of the scaling parameter;
$\mathcal{X}_{\sigma^\star}^k(\mathbb{W})$ — a partition of $\mathbb{W}$ into k clusters corresponding to $\sigma^\star$.
Algorithm:

$$\sigma^\star = \arg\min_\sigma(\mathit{Spectral\ Clustering}(\mathbb{W}, k, \sigma)),$$

$$\mathcal{X}_{\sigma^\star}^k(\mathbb{W}) = \mathit{Spectral\ Clustering}(\mathbb{W}, k, \sigma^\star).$$

σ is described as a human-specified parameter that is selected to form the "tight" k clusters on the surface of the k-sphere. Consequently, it is recommended to search over σ and to take the value that gives the tightest (smallest distortion) clusters of the set $\mathbb{Y}$. Here

$$F_1(\mathcal{X}_\sigma^k(\mathbb{W})) = \frac{1}{|\mathbb{Y}|}\sum_{i=1}^{k}\sum_{\mathbf{y}\in\mathbb{X}_i}\|\mathbf{y}-\mathbf{x}_i\|^2, \tag{6.18}$$

where $\mathbf{x}_i$, $i \in 1..k$ are cluster centroids.

Other functions of this kind can be found within the framework of distance learning methodology. Let us suppose that the degree of similarity between pairs of elements of data collection is known:

$\mathbb{S}$: $\{(\mathbf{w}', \mathbf{w}'')$ *if* $\mathbf{w}'$ *and* $\mathbf{w}''$ *are similar* (*belong to the same cluster*)$\}$
and
$\mathbb{D}$: $\{(\mathbf{w}', \mathbf{w}'')$ *if* $\mathbf{w}'$ *and* $\mathbf{w}''$ *are not similar* (*belong to different clusters*)$\}$

The purpose is to learn a distance metric $q(\mathbf{w}', \mathbf{w}'')$ such that all "similar" data points are kept in the same cluster, (i.e., close to each other) while

distinguishing the "dissimilar" data points. To this end, we define a distance metric in the form:

$$q_{\mathbf{\Gamma}}(\mathbf{w}', \mathbf{w}'') = \|\mathbf{w}' - \mathbf{w}''\|_{\mathbf{\Gamma}}^2 = (\mathbf{w}' - \mathbf{w}'')^{\mathrm{T}} \mathbf{\Gamma} (\mathbf{w}' - \mathbf{w}''),$$

where $\mathbf{\Gamma}$ is a positive semi-definite matrix: $\mathbf{\Gamma} > 0$, which is learned.

We can formulate a constrained optimization problem aimed at minimizing the sum of similar distances concerning pairs in $\mathbb{S}$ while maximizing the sum of dissimilar distances related to pairs in $\mathbb{D}$ as follows:

$$\min_{\mathbf{\Gamma}} \sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{S}} \|\mathbf{w}' - \mathbf{w}''\|_{\mathbf{\Gamma}}^2 \qquad s.t. \qquad \sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{D}} \|\mathbf{w}' - \mathbf{w}''\|_{\mathbf{\Gamma}}^2 \geq 1, \ \mathbf{\Gamma} > 0.$$

If we suppose that the desired metric matrix is diagonal, minimizing the function is then equivalent to solving the stated optimization problem [354] up to a multiplication of $\mathbf{\Gamma}$ by a positive constant. In this way, the second quality function can be expressed as:

$$F_2(\mathcal{X}_\sigma^k(\mathbb{W})) = \sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{S}} \|\mathbf{w}' - \mathbf{w}''\|^2 - \log \left(\sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{D}} \|\mathbf{w}' - \mathbf{w}''\| \right). \quad (6.19)$$

Finally, in the spirit of Fisher's linear discriminant analysis we can consider the function:

$$F_3(\mathcal{X}_\sigma^k(\mathbb{W})) = \frac{\sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{S}} \|\mathbf{w}' - \mathbf{w}''\|^2}{\sum_{(\mathbf{w}', \mathbf{w}'') \in \mathbb{D}} \|\mathbf{w}' - \mathbf{w}''\|^2}. \quad (6.20)$$

As usual, σ is selected for a given k, although the number of clusters is influenced by σ and vice versa. To illustrate this, let us consider the following example of 3-ring data. Three clusters are obviously expected here. However, the results obtained for $\sigma = 1$ appear to be very artificial. The number of clusters is actually equal to 1. For $\sigma = 0.01$ the algorithm reveals the correct cluster structure.

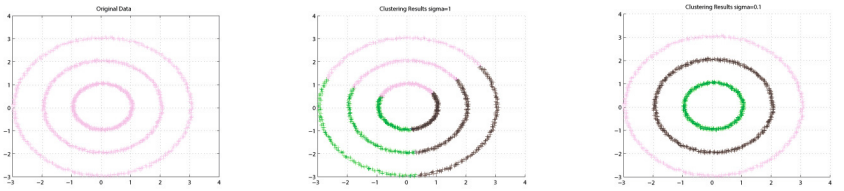

Fig. 6.3. Results of a 3-rings data clusterings for $\sigma = 1$ and $\sigma = 0.01$

6.6.2 σ-Learning and Model Selection

In this section we discuss the application of the σ learning methodology to the model selection problem. Actually, we consider a partial case of the cluster validation problem to be discussed in more detail in Chapter 7. We suggest that σ values should be learned from the samples clustered for several cluster quantities such that the most stable behavior of the parameter is exhibited when the cluster structure is the most stable. In our case, this means that the number of clusters is chosen in the best possible way.

The drawbacks of the used algorithm together with the complexity of the dataset structure add to the uncertainty of the process outcome. To overcome this ambiguity, a sufficient amount of data must be involved. This is achieved by drawing many samples and constructing an empirical distribution of the scaling parameter values. The most concentrated distribution corresponds to the appropriate number of clusters.

Algorithm 3. ***Spectral Clustering Validation*** $(\mathbb{W}, k_{\max}, F, J, m, Ind)$:

Input:
$\mathbb{W}$ — the data to be clustered ($\mathbb{W} \subset \mathbb{R}^d$);
$k_{\max}$ — maximal number of clusters to be tested;
F — cluster quality function to be minimized;
J — number of samples to be drawn;
m — size of samples to be to be drawn;
Ind — concentration index.
Output: $k^\star$ — estimated number of clusters in the dataset.
Algorithm:

- For $k = 2$ to $k_{\max}$ do
- For $j = 1$ to J do
- $\mathbb{S} = sample\,(\mathbb{W}, m)$;
- $\sigma_j =$*Self-Learning Spectral Clustering*$(\mathbb{W}, k, F)$;
- end For j;
- Compute $C_k = Ind\{\sigma_1, \ldots, \sigma_J\}$;
- end For k;

The "true" number of clusters $k^\star$ is chosen according to the most concentrated distribution indicated by an appropriate value of $C_k,\ k = 2, \ldots, k_{\max}$.

Remarks concerning the algorithm: Concentration indexes can be provided in several ways. The most widespread instrument used for the evaluation of a distribution's concentration is the standard deviation. However, this instrument is sensitive to outliers and can be principally dependent, in our situation, on the number of clusters examined. To counterbalance this dependence, the values have to be normalized. Unfortunately, the clustering literature has specified that a standard "correct" strategy, for normalization and scaling, does not exist (see, for example [209] and [314]). We use the coefficient of variation (CV) which is defined as the ratio of the sample standard

deviation to the sample mean. For comparison between arrays with different units, this value is preferable to the standard deviation because it is a dimensionless number.

6.6.3 Numerical Experiments

We exemplify the described approach by means of various numerical experiments on synthetic and real datasets provided for the three functions mentioned in (6.18)–(6.20). We choose $k_{\max} = 7$ in all tests and perform 10 trials for each experiment. The results are presented via the error-bar plots of the coefficient of variation within the trials.

Synthetic Data. The first example refers to the synthetic dataset described in Appendix A.1.1. Here we set $J = 100$ and $m = 400$.

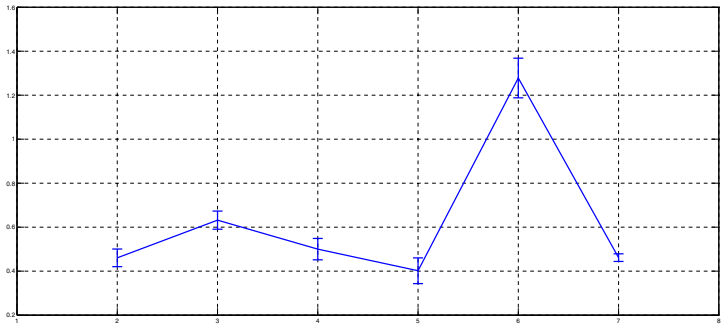

Fig. 6.4. *CV* for the $G5$ dataset using F_1 function

In this case, the *CV* index demonstrates approximately the same performance for all object functions, hinting at a 5- or 7-cluster structure. However, the bars do not overlap only in the first case, where a 5-cluster partition is properly indicated.

Real-World Data. *Three-Text Collection.* The first real dataset is the three-text collection described in Appendix A.2.1.

The results shown in Figures 6.7–6.9 for $m = J = 100$ show that the number of clusters was properly determined for all functions F_1, F_2, F_3.

Iris Flower Dataset. Another real dataset chosen is the well-known Iris flower dataset described in Appendix A.2.3. Here, we selected 100 samples of size 140 for each tested number of clusters. As can be seen, the "true" number of clusters has been successfully found for the F_2 and F_3 objective functions. The experiments with F_1 offer a two-cluster configuration.

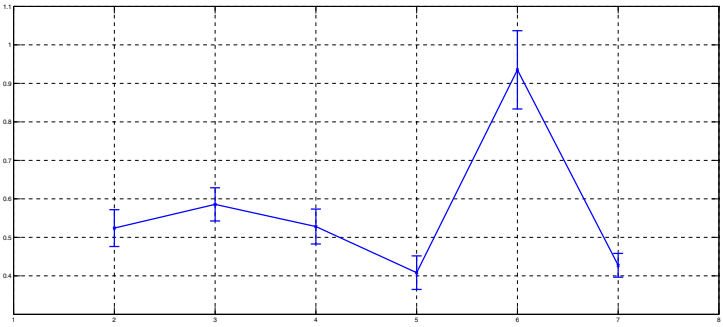

Fig. 6.5. *CV* for the $G5$ dataset using F_2 function

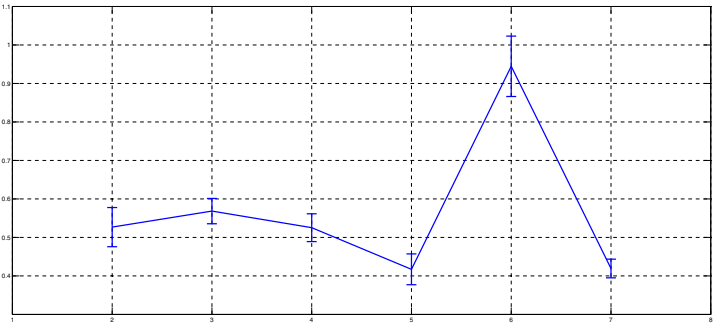

Fig. 6.6. *CV* for the $G5$ dataset using F_3 function

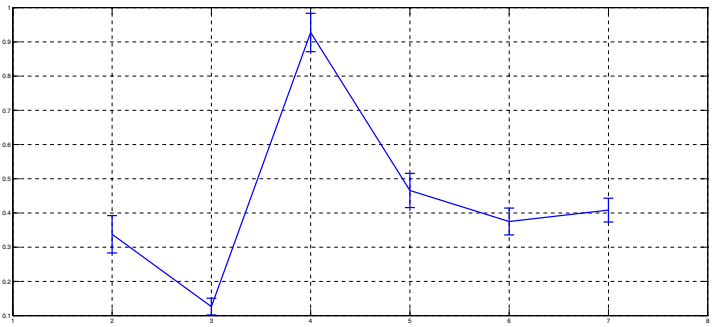

Fig. 6.7. *CV* for the $T3$, 600 terms, using F_1 function

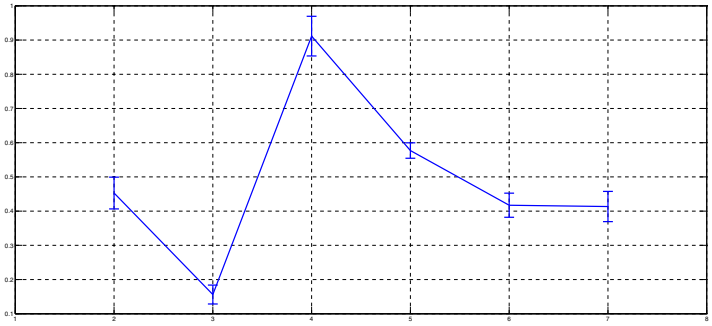

Fig. 6.8. *CV* for the $T3$, 600 terms, using F_2 function

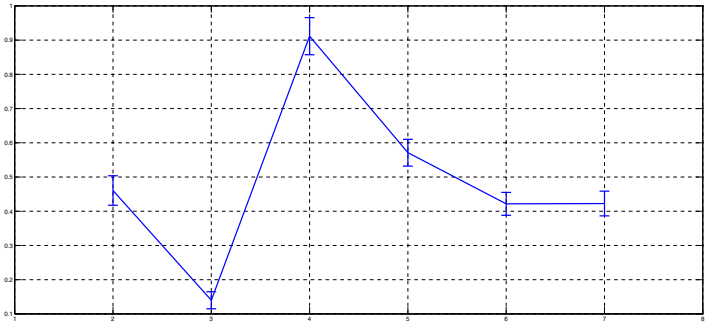

Fig. 6.9. *CV* for the $T3$, 600 terms, using F_3 function

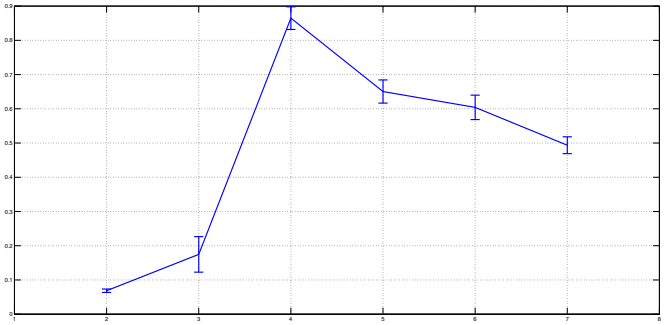

Fig. 6.10. *CV* for the Iris dataset using F_1 function

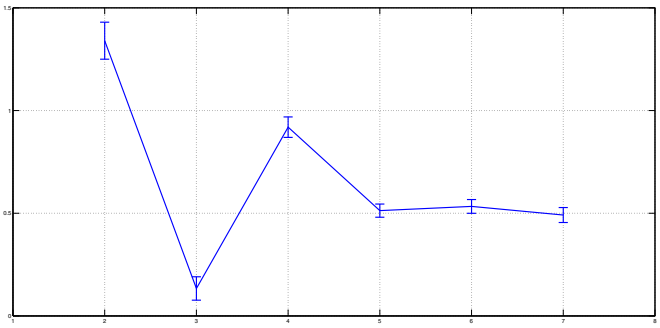

Fig. 6.11. *CV* for the Iris dataset using F_2 function

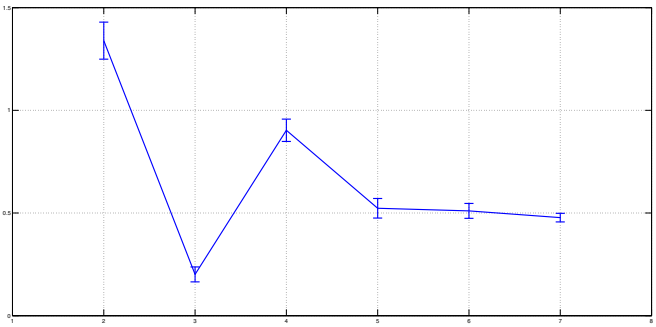

Fig. 6.12. *CV* for the Iris dataset using F_3 function

The Wine Recognition Dataset. This dataset defined in Appendix A.2.4. The parameters in use were $J = 100$ and $m = 100$. Figures 6.13–6.15 demonstrate undoubtedly that for F_2 and F_3 the true number of clusters is revealed; nevertheless, F_1 detects an incorrect structure.

The Glass Dataset. The last real dataset is the Glass dataset defined in Appendix A.2.5. Figures 6.16–6.18 demonstrate outcomes obtained with $J = 100$. Note, that this relatively small dataset possesses a comparatively large dimension and a significantly larger suggested number of clusters in comparison with the previous collection. To eliminate the influence of sample size on the clustering solutions, we draw samples with growing sizes $m = \max((k-1) * 40, 214)$. The minimal value depicted in the graph corresponding to the F_3 function is 6; nevertheless, the bars of 2 and 6 overlap. Since the index behavior is more stable once the number of clusters is 6, this value is accepted as the true number of clusters. Other functions do not manage to determine the true number of clusters.

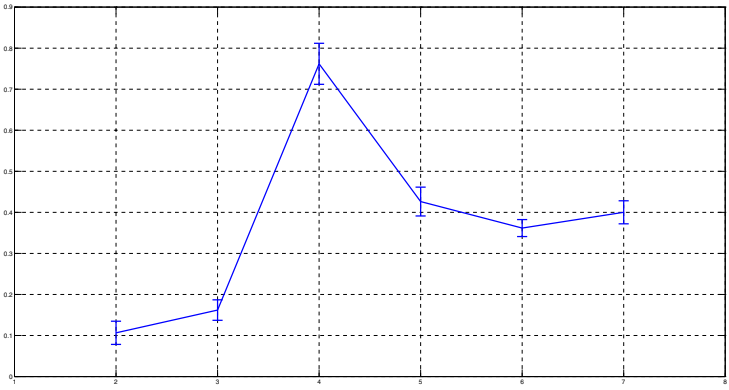

Fig. 6.13. *CV* for the Wine dataset using F_1 function

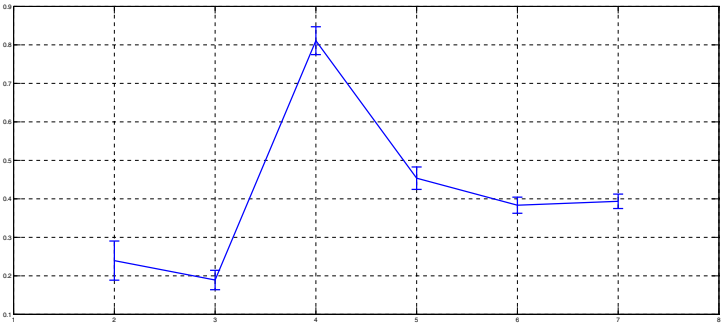

Fig. 6.14. *CV* for the Wine dataset using F_2 function

Comparison of the partition quality function used. Table 6.1 summarizes the results of the numerical experiments provided. As can be seen, the functions F_2 and F_3, introduced in this document subsume the previously offered function F_1.

6.6.4 Application to Writing Style Determination

To reinforce the value of our proposed clustering approach, we applied this enhanced clustering method in an attempt to address the writing style determination problem by means of spectral clustering. The results of our experiments have shown that this method performs very well in distinguishing between texts written by different authors, as well as in recognizing different

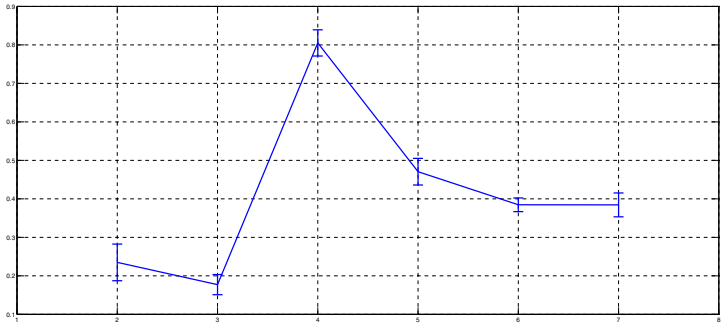

Fig. 6.15. *CV* for the Wine dataset using F_3 function

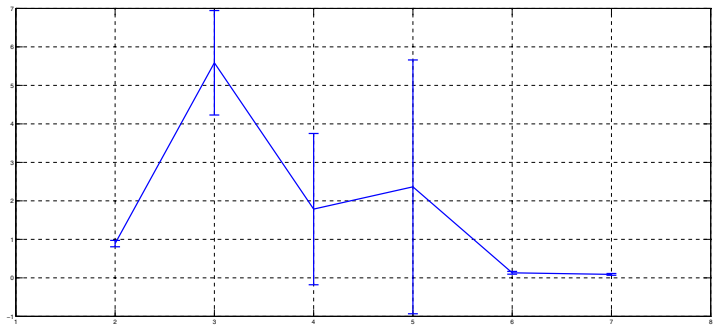

Fig. 6.16. *CV* for the Glass dataset using F_1 function

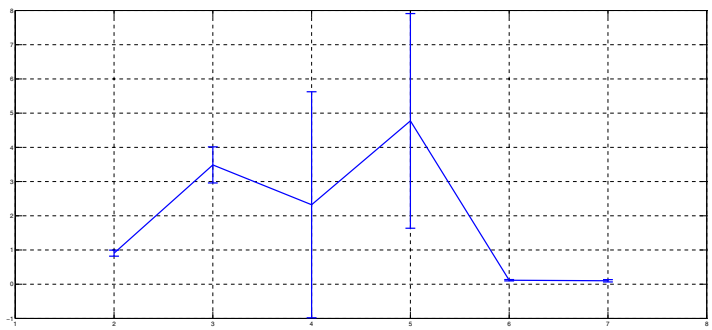

Fig. 6.17. *CV* for the Glass dataset using F_2 function

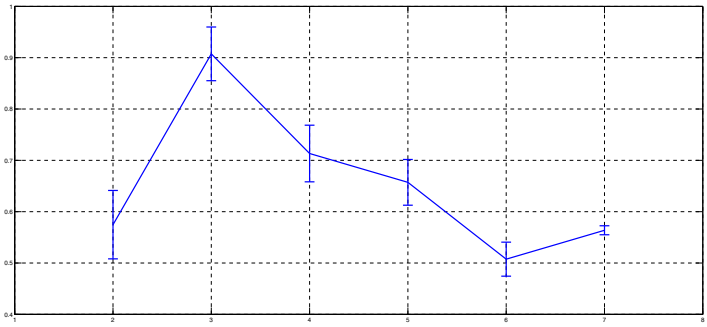

Fig. 6.18. *CV* for the Glass dataset using F_3 function

Table 6.2. Comparison of the partition quality function used

Dataset	F_1	F_2	F_3	TRUE
$G5$	5	5,7	5,7	5
$T3$	3	3	3	3
Iris	2	3	3	3
Wine	2	3	3	3
Glass	7	7	6	6

texts written by the same author. Writing style is a text property that reflects the way an author chooses to express himself. The style is affected by the author's perception of his audience and the way he wishes to convey his thoughts and messages to the audience. Attributes of writing style include the choice of words, syntactical structure (simple/complex), choice of prose (simple/formal wording), tone (casual/proper/formal), use of figurative language (e.g. use of metaphors), and overall organization of thought. Writing style is one of the fundamental characteristics of a text and thus it may be useful as a key to identifying an unknown author.

Numerical Experiments. The experiments in this category were carried out in two stages: First, we exemplified the algorithm by applying it to several texts written by known authors. The objective was to test how well the clustering algorithm is capable of distinguishing between various books and of grouping them according to their author, as determined by the writing style. In the second experiment we used the algorithm to analyze a set of books whose authors are unknown, with the goal of strengthening or weakening the hypothesis that some books were written by a certain author.

In the preprocessing step of each experiment we divided all the documents into equal sized chunks of 10Kbytes. Then, we applied the *Bag of Words method* to represent each chunk as a vector. As a result, each book was

represented by means of several vectors. The two experiments and their results are described in detail in the following sub-section.

1. Text classification by writing style determination. This experiment's data set (CAR) included five books by Arthur C. Clark, four books by Isaac. Asimov (Foundation series), and four books by J. K. Rowling (Harry Potter series). As a first step we tried to use our method to determine the "true" number of clusters in this collection. (Note that this number is evidently expected to be equal to 3.) We chose $k_{\max} = 7$ and repeated all tests 10 times for 100 samples of size 50 drawn from 67 sub-documents obtained here as vectors having 16694 coordinates. The results presented in Figure 6.19 clearly indicate a 3-cluster structure.

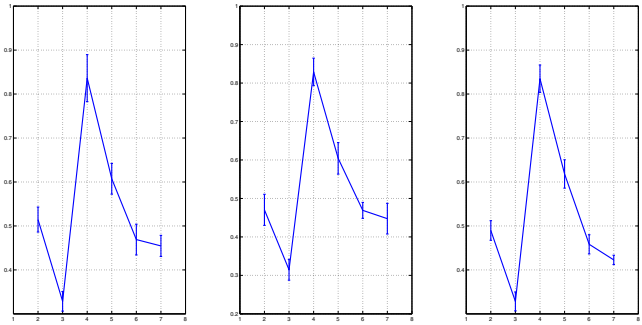

Fig. 6.19. *CV* for the CAR dataset using F_1, F_2, and F_3 functions

The clustering solution attained for $k = 3$ demonstrates that our method achieved perfect separation between the authors as can be seen in the following contingency Table 6.3.

Table 6.3. Contingency table of the authors' partition amid the clusters

cluster/author	Clark	Asimov	Rowling
1	0	18	0
2	18	0	0
3	0	0	27

2. Authorship of the Deutro-Pauline Epistles. In this subsection, we used our clustering method to examine question of authorship of the Deutero-Pauline Epistles. The Pauline Epistles are the fourteen letters in the New Testament traditionally attributed to Paul the Apostle, although the Epistle to the Hebrews does not bear his name. The Pauline Epistles are classified into several groups:

1. The Undisputed Epistles: Romans, 1 Corinthians, 2 Corinthians, 1 Thessalonians, Galatians, Philippians and Philemon, are considered by most scholars to be the genuine work of Paul [255].
2. The Pastoral Epistles: 1 Timothy, 2 Timothy and Titus are thought to be pseudepigraphic by most modern scholars that is written by an imposter of Paul in order to justify ecclesiastical innovations [189].
3. The Deutero-Pauline Epistles: Ephesians, Colossians and 2 Thessalonians, have no consensus on whether or not they are authentic letters of Paul. Thus, they are the focus of our experiments.
4. Almost all scholars agree that the Epistle to the Hebrews was not written by Paul [112].

The Pauline Epistles were among the first documents whose authorship was examined using statistical tools. In 1851, Augustus de Morgan conducted a study on word length statistics in these documents (see, [218]). In recent decades, various studies have been carried out on the Epistles, often reaching conflicting conclusions. We mention the comprehensive work of Kenny [189], who conducted a univariate study and then combined features in order to show the relationships between different epistles. He concluded: "I see no reason to reject the hypothesis that twelve of the Pauline Epistles are the work of a single, unusually versatile author". These twelve epistles included all of the letters traditionally ascribed to Paul, except Titus and Hebrews. Mealand [231] and Ledger [211] independently provided multivariate studies of 1000 word samples. Mealand concluded that Colossians and Ephesians were probably not written by Paul. Ledger further suggested that the authenticity of 1 Thessalonian and Galatians is in doubt. In our experiments, we used the 1550 Stephanus New Testament available at:
http://www.biblegateway.com/versions/index.php?action=getVersionInfo&vid=69.

We checked the number of clusters in the interval $2-15$, where 100 samples of 40 documents were drawn from 49 for each number of clusters. The subdocuments were been obtained as vectors with 7192 features.

Figure 6.20 shows the values of the functions F_1, F_2 and F_3, respectively, used for determining the number of clusters in the Pauline data set. As can be seen, the functions F_2 and F_3 clearly indicate that the optimal number of clusters is three, the function F_1 has a local minimum when the number of clusters equals three.

Application of the adaptive spectral clustering algorithm with $k = 3$ has resulted in the following partition:

1. Cluster 1: 1 Thessalonian, Philippians, 2 Corinthians, 2 Thessalonian, Colossians, Ephesians.
2. Cluster 2: 1 Corinthians, Galatians, Romans.
3. Cluster 3: 1 Timothy, 2 Timothy, Titus, Hebrews.

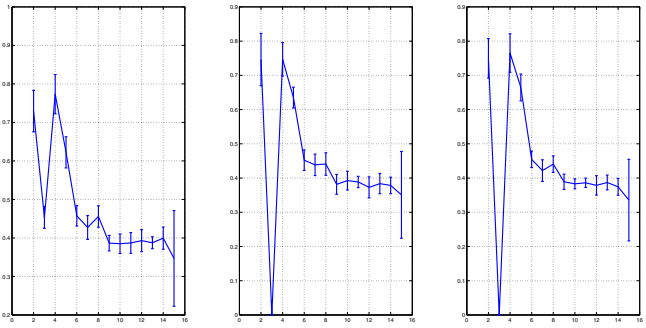

Fig. 6.20. *CV* for the Deutero-Pauline Epistles dataset using F_1, F_2, and F_3 functions

The Deutero-Pauline Epistles, marked in boldface, all belong to Cluster 1, together with the undisputed Epistles. Cluster 2 consists entirely of the undisputed Epistles, whereas Cluster 3 consists of the Pastoral Epistles and Hebrews. According to the above partition, some of the undisputed Epistles are more similar to the Deutero-Pauline Epistles than to other undisputed Epistles. Thus, our experiments seem to reinforce the hypothesis that the Disputed Epistles were in fact written by Paul.

7

Cluster Validation

A historical overview of the cluster validation methods is presented in Chapter 2. As was mentioned there, selecting the model to be used in cluster analysis, and in particular estimating the "true" number of clusters, is an ill-posed problem of essential relevance in cluster analysis. This chapter reviews some recently developed approaches to this problem. Several known stability-based techniques are presented together with geometrical and information approaches to the problem. The new modern probability metrics method is discussed, with its particular implementations such as *the kernel-based approach*, *the Minimal Spanning Tree (MST) approach* and *the Binomial test-based approach*.

7.1 Stability Criteria

Many authors have proposed resampling procedures based on the "cluster stability" concept as cluster validation approaches (see, for example, several recent contributions: [99, 123, 213] and [278].) This section describes the two most popular methods of this type: *the Clest algorithm* and *the general stability approach* stated in [278] and [208].

The Clest algorithm employs the external indexes of partition correlation. For the sake of consistency, we first provide several facts about these approaches and later on describe the algorithm itself.

7.1.1 External Indexes

The calculation of the external indexes is based on the so-called cross-tabulation or contingency table. We compare membership of elements in two partitions $\mathcal{X}^r$ and $\mathcal{X}^s$ of the same dataset. Let $T_{i,j}$ denote the number of items that are members in cluster i of $\mathcal{X}^r$ and in cluster j of $\mathcal{X}^s$ $\ i \in 1..r,\ \ j \in 1..s$. Note that a relationship between two partitions can be considered within the framework of the measures of associations for nominal data. The best known

O. Granichin et al., *Randomized Algorithms in Automatic Control and Data Mining*, Intelligent Systems Reference Library 67,
DOI: 10.1007/978-3-642-54786-7_7,

among these is the Cramer correlation coefficient, defined as follows. Let us introduce

$$T_i^r = \sum_{j=1}^{s} T_{i,j},\ i \in 1..r,\ T_j^s = \sum_{i=1}^{r} T_{i,j},\ j \in 1..s, \tag{7.1}$$

obviously

$$T = \sum_{i=1}^{r} T_i^r = \sum_{j=1}^{s} T_j^s,$$

The chi-square statistic equals

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{s} \frac{(T_{i,j} - e_{i,j})^2}{e_{i,j}},\ e_{i,j} = \frac{T_i^r T_j^s}{T}. \tag{7.2}$$

The Cramer coefficient is

$$V = \sqrt{\frac{\chi^2}{T \min(r-1, s-1)}}. \tag{7.3}$$

Several known external indexes employ the statistics

$$Z = \sum_{j=1}^{s}\sum_{i=1}^{r} T_{i,j}^2. \tag{7.4}$$

The Rand [270] introduced the index

$$R = 1 + \left(\frac{Z - 0.5\left(\sum_{j=1}^{s}\left(T_j^s\right)^2 + \sum_{i=1}^{r}\left(T_i^r\right)^2\right)}{\binom{T}{2}} \right).$$

The Jain and Dubes [175] considered the index

$$JD = \frac{(Z - T)}{\left(\sum_{j=1}^{s}\left(T_j^s\right)^2 + \sum_{i=1}^{r}\left(T_i^r\right)^2 - Z - T\right)},$$

and the Fowlkes and Mallows [121] provides the following expression

$$FM = \frac{(Z - T)}{2\sqrt{\sum_{j=1}^{s}\binom{T_j^s}{2}\sum_{i=1}^{r}\binom{T_i^r}{2}}}. \tag{7.5}$$

It is easy to see that the two indexes R and FM are linear functions of Z, and for that reason each one is a linear function of the other. An external index is frequently standardized in such a way that its expected value is 0, if

the partitions are random, and 1 when they correspond perfectly. The general formula offered to standardize an index is:

$$Ind' = \frac{Ind - E(ind)}{Ind_{\max} - E(ind)},$$

where $Ind_{\max}$ denotes the maximal value of the index Ind.

The most popular null model assumes that the contingency table is created from the generalized hyper-geometric distribution and that the two partitions are mutually independent. In this case the adjusted index has to be zero. Values close to zero specify that from each partition, nothing can be forecasted about the other.

In [34] and [35] the cluster stability problem was considered from the point of view of the concentration of an external index distribution built on samples drawn from the data. The Clest method [99], uses an external index as a stability magnitude.

The Jaccard index is known as a similarity coefficient which is a statistical indicator used for comparing the similarity and diversity of sample sets. The Jaccard similarity coefficient of two sets $\mathbb{A}$ and $\mathbb{B}$ measures similarity between sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets [172]:

$$J(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|}.$$

The Jaccard index $J(\mathbb{A}, \mathbb{B})$ simply indicates the number of unique elements common to both sets divided by the total number of unique elements in both sets. It is a number somewere between 0 and 1: 0 when the two sets are disjoint, 1 when they are equal, and between 0 and 1 otherwise. Two sets are more similar when their Jaccard index is closer to 1 and more dissimilar when their Jaccard index is closer to 0.

The Jaccard distance, which measures dissimilarity between sample sets, is complementary to the Jaccard coefficient and is obtained by subtracting the Jaccard coefficient from 1, or, equivalently, by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union:

$$J_\delta(\mathbb{A}, \mathbb{B}) = 1 - J(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cup \mathbb{B}| - |\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|}.$$

7.1.2 The Clest Algorithm

The Clest algorithm proposes assessing the "true" number of clusters via sequential splitting of the source data set $\mathbb{W}$ into two non-overlapping subsets of the same size. These sets $\mathbb{L}_n$ and $\mathbb{T}_n$ are named a learning set and a test set of a current iteration n, $n \in 1..N$. For each tested number of clusters k, a partition of the learning set is provided. This partition is applied to predict a clusterization of the test set, which simultaneously is divided into

clusters by direct application of the clustering procedure. The two partitions of the test set are evaluated using one of the earlier described external indices which for each k are compared to its expected value calculated within a suitable null distribution of the absent cluster structure. The "true" number of clusters corresponds to the largest significant piece of evidence against the null hypothesis. The idea of such an approach has been proposed by Breckenridge [55].

The **Clest algorithm** can be described in the following form:
For each number of clusters k, $2 \leq k \leq k_{\max}$, perform Steps 1-4.

1. Repeat the following N times ($n \in 1..N$):
   a) Randomly split the original learning set into two non-overlapping sets, a learning set $\mathbb{L}_n$ and a test set $\mathbb{T}_n$.
   b) Apply a clustering procedure to the learning set $\mathbb{L}_n$ to obtain a partition $\mathcal{X}^k(\mathbb{L}_n)$.
   c) Construct a classifier $C(\mathbb{L}_n)$ using $\mathcal{X}^k(\mathbb{L}_n)$.
   d) Apply the classifier $C(\mathbb{L}_n)$ to the test set $\mathbb{T}_n$.
   e) Apply the clustering procedure to the test set $\mathbb{T}_n$ to obtain a partition $\mathcal{X}^k(\mathbb{T}_n)$.
   f) Calculate an external index $Ind_{k,n}$ comparing the two sets of labels for $\mathbb{T}_n$.
2. Let
$$s_k = median(Ind_{k,1}, \ldots, Ind_{k,N})$$
be the observed median value of the external index for the k-cluster partition of the data.
3. Produce N_0 datasets under an appropriate null hypothesis. For each dataset, repeat the procedure outlined in Steps 1 and 2 above to obtain N_0 statistics $s_{k,1}, \ldots, s_{k,N_0}$.
4. Consider the average of these N_0 statistics:
$$\bar{s}_k^0 = \frac{1}{N_0} \sum_{n=1}^{N_0} t_{k,n},$$
and use p_k to denote the proportion of the $s_{k,n}$, $n \in 1..N_0$, that are at least as large as the observed statistic s_k, that is, the p-value for s_k.
Let $d_k = s_k - s_k^0$ denote the difference between the observed similarity statistic and its estimated expected value under the null hypothesis. Introduce the set K as
$$K = \{2 \leq k \leq k_{\max} : p_k \leq pmax, d_k \geq dmin\},$$
where $pmax$ and $dmin$ are predefined parameters. If this set is empty, no cluster structure has been detected. Otherwise, the number of clusters $k^\star$ corresponds to the largest significant difference statistic d_k:
$$k^\star = \arg\max_{k \in K} d_k.$$

The authors used the clustering procedure, named *Partitioning Around Medoids (PAM) algorithm*, described in [187], the naive Bayes classificator, the FM index (7.5), $N = N_0 = 20$, and $pmax = dmin = 0.05$.

The PAM algorithm takes as input parameters the wanted number of clusters k and the previously constructed $|\mathbb{W}| \times |\mathbb{W}|$ matrix of all pairwise distances between elements of $\mathbb{W}$. The output is a set of k cluster centers, called here "medoids", which are themselves items of $\mathbb{W}$. PAM admits clustering regarding any specified dissimilarity metric, and the medoids provide robust representations of the cluster centers. PAM is operational for small data sets but does not scale well for large data sets due to its quadratic complexity with respect to the dataset size.

7.1.3 The General Stability Approach

The idea of employing the semi-supervised learning methodology with external indexes as a partition goodness attribute has been generalized in [208] and [278]:

1. Split the dataset $\mathbb{W}$ into two sets of equal size: $\mathbb{W}_1$ and $\mathbb{W}_2$.
2. Enter the first dataset to a partitioning clustering algorithm. The result is the mapping $\gamma_1(\cdot)$ of each of the objects in $\mathbb{W}_1$ to one of the k clusters.
3. Enter the second dataset $\mathbb{W}_2$ to the algorithm. The result is the mapping $\gamma_2(\cdot)$ of each of the objects in $\mathbb{W}_2$ to one of the k clusters. Use $\gamma_2(\cdot)$ to predict the cluster membership of all objects contained in the first set $\mathbb{W}_1$.
4. The set $\mathbb{W}_1$ has two different clustering labeling. Find the correct permutation of labels by using the Hungarian method for the minimum weighted perfect bipartite matching (see, [202]).
5. Normalize the cost of the minimum weighted perfect bipartite matching with respect to the random stability.
6. Reiterate the whole procedure from Step 1 to Step 5, average over assignment costs and compute the expected stability value.
7. Reiterate the whole procedure for each k to be tested.

An application of the Hungarian method is required since cluster labels can be permuted within repeated rerunning of a clustering algorithm. A matching between the labels can be generated by solving the following:

$$\tau_k(\gamma_1, \gamma_2) = \min_{\psi \in \Psi_k} \left(mean \sum_{\mathbf{w}_j \in \mathbb{W}} \mathbf{1}_{\{\gamma_1(\mathbf{w}_j) \neq \psi(\gamma_2(\mathbf{w}_j))\}} \right). \tag{7.6}$$

(Recall, Ψ_k is the set of all possible permutations of the set $1..k$). The computational complexity for solving the problem by the Hungarian method (see [202]) is $\mathcal{O}(|\mathbb{W}| + k^3)$. The process needs time $\mathcal{O}(|\mathbb{W}|)$ for weight matrix creation and time $\mathcal{O}(k^3)$ to perform the matching itself. Such a technique has

been applied in [208, 278] and is also known in the cluster combining area (see, for example, [320]).

Normalization with respect to the random stability means

$$s_k = \frac{mean(\tau_k(\gamma_1, \gamma_2))}{mean(\tau_k(\rho_1, \rho_2))}, \tag{7.7}$$

where ρ_1, ρ_2 are the random predictors that uniformly assign cluster labels. The estimated number of clusters is given by k, providing the minimum of s_k.

Authors of [208] and [278] have noted that splitting a dataset into two disjoint subsets is recommended because the size of the individual sets should be large. Nevertheless, this approach can be applied formally for any sample size.

7.2 Geometrical Criteria

Given a partition $\mathcal{X}^k$, $2 \leq k$, the total dispersion matrix or total scatter matrix is given by

$$\mathbf{S}_k = \sum_{j=1}^{k} \sum_{\mathbf{w} \in \mathbb{X}_j} (\mathbf{w} - \overline{\mathbf{x}})(\mathbf{w} - \overline{\mathbf{x}})^{\mathrm{T}}, \tag{7.8}$$

where $\overline{\mathbf{x}}$ is the arithmetic mean of the set $\mathbb{W}$. Matrices $\mathbf{B}_k$ and $\mathbf{R}_k$ of between and within k-clusters sums of squares are defined by

$$\mathbf{B}_k = \sum_{j=1}^{k} |\mathbb{X}_j| \, (\overline{\mathbf{x}}_j - \overline{\mathbf{x}})(\overline{\mathbf{x}}_j - \overline{\mathbf{x}})^{\mathrm{T}}, \; \mathbf{R}_k = \sum_{j=1}^{k} \sum_{\mathbf{w} \in \mathbb{X}_j} (\mathbf{w} - \overline{\mathbf{x}}_j)(\mathbf{w} - \overline{\mathbf{x}}_j)^{\mathrm{T}}, \tag{7.9}$$

where $\overline{\mathbf{x}}_j$ is the arithmetic mean of $\mathbb{X}_j$, $j \in 1..k$. Note that, $\mathbf{S}_k = \mathbf{R}_k + \mathbf{B}_k$ (e.g., see [226]).

The following inner indices are frequently used to estimate the number of clusters in a dataset.

1. *The Calinski and Harabasz index* [60] is defined by

$$CH_k = \frac{\mathrm{Tr}[\mathbf{B}_k]/(k-1)}{\mathrm{Tr}[\mathbf{R}_k]/(T-k)},$$

   where T is the size of the considered dataset. The estimated number of clusters is given by k providing the maximum of CH_k.
2. *The Krzanowski and Lai index* [201] is defined by the following relationships

$$diff_k = (k-1)^{2/d}\mathrm{Tr}[\mathbf{R}_{k-1}] - k^{2/d}\mathrm{Tr}[\mathbf{R}_k],$$

   where d is the dimension of the dataset being considered, and

$$KL_k = |diff_k|/|diff_{k+1}|.$$

The estimated number of clusters matches the maximal value of the index KL_k.

3. *The Hartigan index* [159] is defined by

$$h_k = \left(\frac{\mathrm{Tr}[\mathbf{R}_k]}{\mathrm{Tr}[\mathbf{R}_{k+1}]} - 1 \right) (T - k - 1).$$

The estimated number of clusters is the smallest $k \geq 1$ such that $h_k \leq 10$, where, again, T is the size of the dataset being considered.

4. *The Sugar and James index* [309] proposes an information theoretic approach for finding the number of clusters in a dataset. Here, differences of the transformed distortions are calculated. Namely,

$$J_k = \mathrm{Tr}[\mathbf{R}_k]^{-\lambda} - \mathrm{Tr}[\mathbf{R}_{k-1}]^{-\lambda},$$

where λ is the transformation power. The estimated number of clusters matches the maximal value of the index J_k.

5. *The Gap index* [314] computes the values $\mathrm{Tr}[\mathbf{R}_k]$ for each $k \geq 1$. N reference datasets are generated under the null distribution assumption ($N = 10$ in the section). Each of the datasets are entered into a clustering procedure, and the values $\mathrm{Tr}[\mathbf{R}_k^1], \ldots, \mathrm{Tr}[\mathbf{R}_k^N]$ are evaluated. The estimated gap statistics are calculated by

$$gap_k = \frac{1}{N} \sum_n \log\left(\mathrm{Tr}[\mathbf{R}_k^n]\right) - \log\left(\mathrm{Tr}[\mathbf{R}_k]\right).$$

Let sd_k be the standard deviation of $\log\left(\mathrm{Tr}[\mathbf{R}_k^n]\right)$, $n \in 1..N$, and

$$\widehat{sd}_k = sd_k \sqrt{1 + \frac{1}{N}}.$$

The estimated number of clusters is the smallest $k \geq 1$ such that

$$gap_k \geq gap_{\bar{k}} - \widehat{sd}_{\bar{k}},$$

where $\bar{k} = \arg\max_{k \geq 1}(gap_k)$. Two approaches are considered for constructing the region of support for the distribution (for details see [314]).

6. *The Silhouette index* [279] provides a succinct graphical representation of how well each object lies within its cluster. Each cluster is represented by a so-called silhouette, which is based on the comparison of its tightness and separation. This silhouette shows which objects lie well within their cluster and which are merely somewhere between clusters. The entire clustering is displayed by combining the silhouettes into a single plot, providing an appreciation of the relative quality of the clusters and an overview of the data configuration. The average silhouette width provides an evaluation of clustering validity and may be used to select an "appropriate" number of clusters.

Assume the data have been clustered via any technique, such as k-means, into k clusters. For each element $\mathbf{w}_i$, let a_i be the average dissimilarity of $\mathbf{w}_i$ with all other data within the same cluster. Any measure of dissimilarity can be used, but distance measures are the most common. We can interpret a_i as how well matched $\mathbf{w}_i$ is to the cluster it is assigned (the smaller the value, the better the matching). Then find the average dissimilarity of $\mathbf{w}_i$ with the data of another single cluster. Repeat this for every cluster of which $\mathbf{w}_i$ is not a member. Denote the lowest average dissimilarity to $\mathbf{w}_i$ of any such cluster by b_i. The cluster with this lowest average dissimilarity is said to be the "neighboring cluster" of $\mathbf{w}_i$ as it is, aside from the cluster to which $\mathbf{w}_i$ is assigned, the cluster in which $\mathbf{w}_i$ fits best. We now define:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

which can be written as:

$$s_i = \begin{cases} 1 - a_i/b_i, & \text{if } a_i < b_i \\ 0, & \text{if } a_i = b_i \\ b_i/a_i - 1, & \text{if } a_i > b_i \end{cases}. \tag{7.10}$$

From the above definition it is clear that $-1 \leq s_i \leq 1$.

For s_i to be close to 1 we require $a_i \ll b_i$. As a_i is a measure of how dissimilar $\mathbf{w}_i$ is to its own cluster, a small value means it is well matched. Furthermore, a large b_i implies that $\mathbf{w}_i$ is badly matched to its neighboring cluster. Thus an s_i close to one means that the datum is appropriately clustered. If s_i is close to negative one, then by the same logic we see that $\mathbf{w}_i$ would be more appropriate if it was clustered in its neighboring cluster. An s_i near zero means that the datum is on the border of two natural clusters.

The average s_i of a cluster is a measure of how tightly grouped all the data in the cluster are. Thus the average s_i of the entire dataset is a measure of how appropriately the data have been clustered. If there are too many or too few clusters, as may occur when a poor choice of k is used in the k-means algorithm, some of the clusters will typically display much narrower silhouettes than the rest. Thus silhouette plots and averages may be used to determine the natural number of clusters within a dataset.

7.3 Information Criteria

One of the most appropriate statistical tools used in the field of model selection is *the likelihood ratio test* which has simple asymptotic behavior under natural regularity conditions [352]. In the mixture models framework this

approach appears to be tricky due to the fact that the null hypothesis being tested is applied on the parameter space's frontier. Thus, the Cramer regularity condition on the asymptotic properties is not formally applicable.

To overcome this difficulty it is possible to consider model clustering that relies on multiple information criteria, such as *the Akaike's Information Criteria (AIC)* or *the Bayesian Information Criterion (BIC)* in cases where a likelihood function for the clustering model is given, for example, in the context of the GMM presented in Section 6.4. Recall that the general finite mixture model is defined in a form similar to that in (6.9). The data $\mathbb{W}$ are considered to be drawn from a population having a probability distribution with the density

$$f(\mathbf{w}|\mathbf{X}) = \sum_{i=1}^{k} p_i f_i(\mathbf{w}|\mathbf{x}_i) \tag{7.11}$$

where

- $p_i,\ i \in 1..k$ — cluster proportions;
- $\mathbf{x}_i,\ i \in 1..k$ — vectors of unknown parameters of functions $f_i(\cdot)$;
- $\mathbf{X} = (p_1, \ldots, p_k, \mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_k^{\mathrm{T}})^{\mathrm{T}}$ — vector of all unknown parameters.

Accordingly, the minus log-likelihood function for the parameters is:

$$F(\mathbf{X}) = -\log\left(L(\mathbf{X}, \mathbb{W})\right) = -\left(\sum_{\mathbf{w}\in\mathbb{W}} \log\left(\sum_{i=1}^{k} p_i f_i(\mathbf{w}|\mathbf{x}_i)\right)\right).$$

The mixture model can be seen as an incomplete data structure model, where the complete data are given by

$$\mathbb{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\} = \{(\mathbf{w}_1, \mathbf{z}_1), \ldots, (\mathbf{w}_T, \mathbf{z}_T)\},$$

with matrix $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_T)$ representing the missing data such that in the case of hard clustering an element $z_{i,j}$ is the i-th coordinate of a vector $\mathbf{z}_j$and takes the value 1 if $\mathbf{w}_j$ arises from the component i, and takes the value 0 if not:

$$z_{i,j} = \begin{cases} 1, & \text{if } \mathbf{w}_j \text{ belongs to group } i \\ 0, & \text{otherwise} \end{cases}.$$

The appropriate assumptions are that the density of an observation $\mathbf{w}_j$ given $\mathbf{z}_j$ is

$$g(\mathbf{w}_j) = \prod_{i=1}^{k} \left(f_i(\mathbf{w}|\mathbf{x}_i)\right)^{z_{i,j}},$$

and that $\{\mathbf{z}_j\}$ are independent and identically distributed, each one according to a multinomial distribution of k categories with probabilities $\{p_i, i \in 1..k\}$. The complete-data minus log likelihood (e.g., see [230]) is

$$F_c(\mathbf{X}) = -\log\left(L_c(\mathbf{X}, \mathbb{W})\right) = -\left(\sum_{\mathbf{w}_j\in\mathbb{W}} \sum_{i=1}^{k} z_{i,j} \log\left(p_i f_i(\mathbf{w}_j|\mathbf{x}_i)\right)\right).$$

For fuzzy classification the matrix $\mathbf{Z}$ with elements

$$z_{i,j} = \frac{p_i f_i(\mathbf{w}_j|\mathbf{x}_i)}{\sum_{i=1}^{k} p_i f_i(\mathbf{w}|\mathbf{x}_i)}$$

measures the overlap of the mixture components. Accordingly, the entropy of such a partition is defined as

$$EN(k) = -\sum_{\mathbf{w}_j \in \mathbb{W}} \sum_{i=1}^{k} z_{i,j} \log\left(z_{i,j}\right).$$

AIC [4] measures the complexity of a data model for a given dataset with respect to its goodness of fit. Actually, the AIC is defined as

$$AIC = 2(\bar{d} - L),$$

where $\bar{d}$ is the number of parameters in the statistical model and L is the maximized value of the likelihood function. An additional commonly applied information criterion is the BIC. Unlike the AIC, this indicator is derived within the Bayesian framework as an estimate of the Bayes factor for two competing models [188, 289]

$$BIC = \bar{d}\ln(T) - L,$$

where T is the data size.

Several famous criteria have been proposed based on the stated fundamental notions (see [114]). The essential principle underlying these information criteria is the parsimony that makes it possible to choose the simplest model (with fewer parameters) among all others having the same log-likelihood.

Table 7.1 summarizes several popular criteria presented in the literature.

7.4 Probability Metric Method

In this section we describe the previously mentioned cluster stability model resulting from the notion of a well mixed sample. A suitable number of clusters is estimated via the stability of sample appearances within the clusters constructed by the clustering algorithm. The concept suggests statistical homogeneity of these simulated samples in the case of the "true" number of clusters. That is, we presume that in such a situation sample occurrences in a cluster appear to be independent realizations of the same random variable. Naturally, this leads to stability rate quantification via compound or simple probability metrics such that in the case of a stable partition these metrics are suggested to take small values. A probability metric should actually be chosen to reflect the structure of the considered clustering task.

Although we cannot directly choose samples from hidden clusters, we are able to rely on the fact that we may use clustering of i.i.d. samples drawn

Table 7.1. Information criteria

Criteria	Definition	Author
$AIC3$	$2(p-L)-p$	[52]
$AICc$	$AIC+2\bar{d}(\bar{d}+1)/(T-\bar{d}-1)$	[170]
$AICu$	$AICc+T\ln(T/(T-d-1))$	[227]
$CAIC$	$\bar{d}(1+\ln(T))-2L$	[51]
BIC/MDL	$\bar{d}\ln(T)-2L$	[272], [289]
CLC	$EN(k)-2L$	[40]
ICL_BIC	$BIC+2EN(k)$	[41]
NEC	$EN(k)/(L(k)-L(1))$	[42]
AWE	$-2L_c+2d(3/2+\ln(T))$	[23]
L	$-L+(\bar{d}/2)\sum_{i=1}^{k}\ln(Tp_i/12)+k/2\ln(T/12)+k(\bar{d}+1)/2$	[111]

from the entire dataset to simulate these values. Complicated sample configurations and algorithm weaknesses seriously amplify the noise level of the procedure. To overcome such obstacles, our conclusions should be established on a sufficient amount of data and on repetitive activation of the clustering algorithm for each one of the tested number of clusters. Due to the fact that the distance distribution may increase depending on the number of clusters (see, for example, [209] and [314]), we apply an appropriate normalization to balance this result. The true number of clusters is characterized by the normalized empirical distance distribution which is most concentrated at the origin.

Note that the interpretation of clustering in the form of random variables has been discussed in the literature. For example, the issue of the approach offered in [209] has actually been formulated in terms of random variables and their characteristics. A meta-algorithm implementing the offered methodology can be expressed as follows:

Algorithm:

1. Repeat for each tested number of clusters from k_{min} to k_{max}:
   a) Repeat the following steps for pairs randomly drawn without replacement samples:
      i. Simulate the occurrence of samples in the clusters with the help a clustering algorithm.
      ii. Calculate probability metric values between the occurrences of the samples within the clusters.
      iii. Calculate a summarizing value representing the distance between the samples in the partition.
      iv. Normalize the distances values.
2. The estimated number of clusters is the one whose normalized distribution is mostly concentrated at the origin.

A probability metric is a rule that sets dissimilarity amid random variables or vectors. Our intention in this subsection is to review the notion of probability metrics and to discuss several important examples. The probability metrics theory is stated in [360] and [269]. Let Ω denotes a measurable space with σ-algebra $\mathcal{B}$ and the probability measure P. Denote by $\mathcal{D}$ the set of all random variables defined on $(\Omega, \mathcal{B}, \mathrm{P})$. A mapping Dis: $\mathcal{D} \times \mathcal{D} \to \mathbb{R}_+$ is said to be a probability metric if for all $X, Y, Z \in \mathcal{D}$ the following conditions are fulfilled:

1. The identity property
   $Dis(X, Y) = 0$ if and only if $X = Y$ almost surely, i.e., $\mathrm{P}(X = Y) = 1$.
   Sometimes it is presented in more weak forms:
   a. $Dis(X, Y) = 0$ if $X = Y$ almost surely.

   If only this condition holds, the metric is called a semi-metric.
   b. $Dis(X, Y) = 0$ implies equality of the X and Y distributions.

   If only this condition holds, the metric is called simple; otherwise the metric is called compound.
2. The symmetry axiom:
   $Dis(X, Y) = Dis(Y, X)$.
3. The triangle inequality axiom:
   $Dis(X, Y) \leq Dis(X, Z) + Dis(Z, Y)$.

7.4.1 Compound Metrics

Compound distance can be applied as measures of uncertainty. Actually, let $Dis(\cdot, \cdot)$ be a compound distance. For a pair X, X_1 of i.i.d. variables the equation

$$Dis(X, X_1) = 0$$

implies $P(X = X_1) = 1$. Explicitly, X is almost surely a constant variable. By this reasoning, the value

$$\rho(X) = Dis(X, X_1)$$

is called an index concentration measure derived by the compound distance Dis.

In the clustering perception two measurable natural spaces become visible: $\Omega = 1..k$ and $\Omega = \mathbb{W} \subset \mathbb{R}^d$ endowed with their inherent σ-algebras. The number of clusters is typically suggested to be not too large. Consequently, the discrete space $\Omega = 1..k$ owns the small cardinality property, and all probability metrics defined here evidently are straightforwardly comparable. The most explicable from the clustering point of view metric is the indicator metric:

$$Dis_0(X, Y) = \mathrm{P}(X \neq Y).$$

The associated concentration measure

$$\rho(X) = \operatorname{ess\,sup} \mathrm{P}(X \neq X_1) \tag{7.12}$$

appears de facto in stability-based approaches (e.g., [99, 209, 213]) as a disagreement measure between repeated solutions given for the same clustered set. A normalization is provided here according to the "worst" distribution assuming the absence of the clustering structure, where X and X_1 are uniformly distributed on $1..k$. In this case the random variable

$$Z = 1_{\{X \neq X_1\}}$$

is obviously a Bernoulli random variable with a success probability of $(1 - \frac{1}{k})$, and a normalized version of $\rho(X)$ is constructed as

$$\widetilde{\rho}(X) = \frac{\rho(X)}{E(Z)} = \frac{\rho(X)}{(1 - \frac{1}{k})}. \tag{7.13}$$

For the sake of accuracy, it must be noted that the factor $E(Z)$ is often approximated by simulation according to finite sample sizes.

7.4.2 Simple Metrics

Simple metrics are a very acceptable tool in the so-called "two-sample problem". This tool tests the hypothesis whether two samples were drawn from the same population. Namely, let F and G be two unknown distributions and the null hypothesis

$$H_0 : F(\mathbf{w}) = G(\mathbf{w})$$

is tested against the general alternative

$$H_1 : F(\mathbf{w}) \neq G(\mathbf{w}).$$

A simple distance-statistic value is usually calculated on the drawn samples. The distribution of the distance is given under the null hypothesis. Within all possible values of the statistic, the most extreme evidence against the null hypothesis is chosen. It defines the critical region of the test. The null hypothesis is rejected if the actual distance value falls in the critical region.

The Kolmogorov-Smirnov test, the Cramer-von Mises test, the Friedman's nonparametric ANOVA test and the Wald-Wolfowitz test must be mentioned as the classical univariate procedures for this purpose. Information theoretic measures, such as the known Kullback-Leibler or Jensen-Shannon divergences quantifying the difference between two probability distributions, can actually be considered as simple probability metrics.

Many tests have been also derived for the multivariate case [26, 84, 101, 119, 155, 163, 277] and [357]. The multivariate two-sample test based kernel statistics introduced in [150, 151, 152, 153] has found many notable applications in machine learning theory. Closely connected to this methodology, a

kernel method for the two sample problem was earlier and independently proposed in [194]. This test is based on a characterization theorem stated in [359]. Applications of this approach have been discussed in [196] (see also [195]). *The two-sample energy test* described in [357] can also be interpreted within the framework of this methodology. We examine kernel-based distances in detail in Section 7.6.

As mentioned earlier, with the framework of our approach a two sample test statistic (simple probability metric) is intended to describe partition goodness by means of mingling the quality of items belonging to two disjoint i.i.d. samples, $\mathbb{S}_1$ and $\mathbb{S}_2$, within clusters: $n = |\mathbb{S}_1|$ and $m = |\mathbb{S}_2|$. To explain this concept let us consider the illustrative examples presented in Figure 7.1 and Figure 7.2. Sample items situated on the left of the picture are not

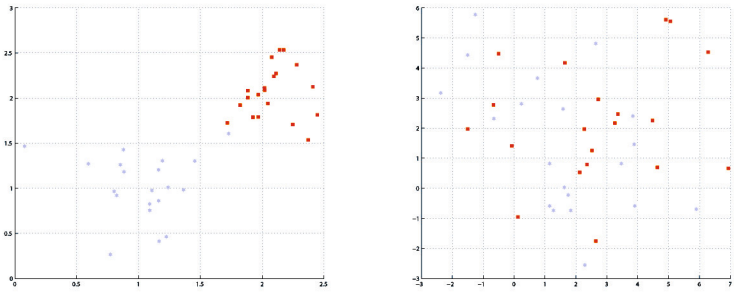

Fig. 7.1. Elements of two samples mixed in a cluster

well mixed amid a cluster; apparently, each sample produces its own cluster. In contrast, the samples on the right are well mingled and define the same stable cluster. We can measure the mixture merit by means of the K-nearest neighbor fractions of the samples quantified at each point. Obviously, these proportions are approximately equal if the samples are well mixed. Cluster validation from this point of view has been considered in [335]. The K-nearest neighbors type coincidences model applied to this problem in [339] deals with the statistic:

$$S_{n,m,K} = \sum_{\mathbf{x} \in \mathbb{S}_1 \cup \mathbb{S}_2} \sum_{r=1}^{K} 1_{\{\mathbf{w} \text{ and } r\text{-th neighbor belong to the same sample}\}},$$

which represents the number of all K-nearest neighbors type of coincidences. Asymptotic behavior of this statistic has been studied in [163], and an application to the cluster validation problem is considered in [341]. Other acceptable tools in the two sample test problems are based on *the Minimal*

Spanning Trees approaches. Figure 7.2 shows Minimal Spanning Trees constructed inside the clusters presented in Figure 7.1. The edges connecting points from different samples are marked by a solid red line. The left pic-

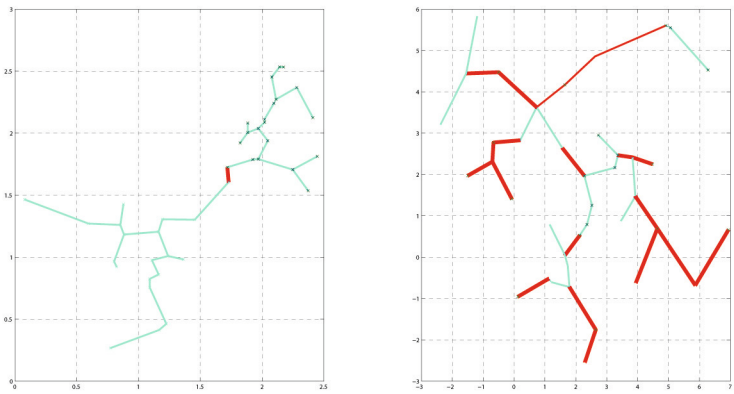

Fig. 7.2. Examples of Minimal Spanning Trees constructed inside a cluster

ture shows only one edge which connects between the items belonging to different clusters. The right picture demonstrates a consistent cluster where the number of "red" edges is sufficiently large. Thus, a large number of the "red" edges inside a cluster indicates its consistency. The Friedman-Rafsky test statistic [119], which counts the number of disjoint sub-trees resulting from removing all edges uniting vertices of different samples, quantifies this mixing feature. Cluster stability using this technique has been considered in [30] and [339].

7.4.3 Information Metrics

The Kullback-Leibler divergence (information divergence or relative entropy) is a non-symmetric measure of the difference between probability distributions. This measure is frequently intuited as a distance metric (KL -distance). Relative entropy is not a true metric because it is not symmetric and does not satisfy the triangle inequality. However, it is a natural distance function between a "true" probability distribution and a "target" probability distribution. The KL-distance between the probability distributions of random variables on a discrete measurable space is proposed to be

$$D_{KL}(\mu||\nu) = \sum_i \mu(x_i) \log_2 \left(\frac{\mu(x_i)}{\nu(x_i)} \right).$$

The usual convention, based on continuity arguments, is to take $0log\frac{0}{q} = 0$ for all real q and $plog\frac{p}{0} = \infty$ for all real non-zero value p. Thus, the relative entropy takes values in $[0, \infty]$.

The Jensen-Shannon divergence is a symmetrized and smoothed version of the Kullback-Leibler divergence and is defined as

$$D_{JS}(\mu||\nu) = \frac{1}{2}\left(D_{KL}(\mu||\eta) + D_{KL}(\nu||\eta)\right),$$

where $\eta = \frac{1}{2}(\mu + \nu)$. These measures can be generally employed in our methodology, although they are not actual probability metrics. Information stability measures are widely used in cluster analysis, since clustering can be viewed as a strategy for data compression by cleaning out irrelevant information via appointing a representative to each group. Lossy compression is provided by transmitting the data to clusters such that the mutual information between the data and the cluster representatives is minimized subject to fixing the expected distortion (see [276, 306]).

The approach aims in finding the "true" number of clusters by relying on information measures in [252]. The method is proposed in the framework of [208, 209] where partition quality is characterized by the similarity between two partitions: a partition constructed via transfer by prediction and a partition obtained directly. The distributions are estimated by means of two samples drawn from the data, and the distribution closeness is actually given by sufficiently mixing the samples in the clusters.

7.5 Simulation

In this section methods for simulating sample occurrences in clusters are introduced. Recall that these variables can take values in cluster applications in $1..k$ or $\mathbb{W}$. Samples elements in the clusters are created to represent independent random variables. Therefore, an independent clustering procedure must be applied for each of the samples. The intention is to avoid possible influences of the samples in the clustering process. This task addresses the problem of the comparison between two cluster solutions obtained from different (frequently disjoint) sets. A rendering tool must be developed for this purpose. Mechanisms presented in [34] and [213] propose to operate on datasets intersections. As pointed out in [208, 209] such an approach can be biased.

7.5.1 First Simulation Method

As mentioned earlier, this methodology was actually employed in [55, 208] and [278]. It suggests evaluating the likeness of clusterings via a classifier trained by means of a second (clustered) dataset. Apart from a basic clusterer $Cl_k(\mathbb{S})$ this approach uses an additional clusterer $\varepsilon_k(\mathcal{X}^k(\mathbb{S}_0))(\mathbb{S})$ proposed to extend

a partition of the set $\mathbb{S}_0$ to a partition of set $\mathbb{S}$. The cluster occurrences here are constructed on a set $\mathbb{S}_1 \subset \mathbb{W}$ using an auxiliary subset $\mathbb{S}_2 \subset \mathbb{W}$.

- Two solutions

$$\mathcal{X}_1^k(\mathbb{S}_1) = Cl_k(\mathbb{S}_1),$$
$$\mathcal{X}^k(\mathbb{S}_2) = Cl_k(\mathbb{S}_2)$$

  are determined.
- An additional cluster solution for the dataset $\mathbb{S}_1$ is obtained as

$$\mathcal{X}_2^k(\mathbb{S}_1) = \varepsilon_k(\mathcal{X}^k(\mathbb{S}_2))(\mathbb{S}_1).$$

  If there is a correspondence between the cluster labels in the two partitions $\mathcal{X}_1^k(\mathbb{S}_1)$ and $\mathcal{X}_2^k(\mathbb{S}_1)$, then this partition would be considered as the occurrences of $\mathbb{S}_1$ within clusters of the partition $\mathcal{X}^k(\mathbb{S}_2)$. Actually, the same cluster can be labeled differently in repeated data partitioning.
- This ambiguity is overcome by determining an optimal label permutation to provide minimal dissimilarity amid the cluster solutions. To solve this problem a discrepancy measure between the partitions is considered:

$$D_k(\gamma_{\mathcal{X}_1^k(\mathbb{S}_1)}, \gamma_{\mathcal{X}_2^k(\mathbb{S}_1)}, \psi) = \frac{1}{|\mathbb{S}_1|} \sum_{\mathbf{w} \in \mathbb{S}} 1_{\{\gamma_{\mathcal{X}_1^k(\mathbb{S}_1)}(\mathbf{w}) \neq \psi(\gamma_{\mathcal{X}_2^k(\mathbb{S}_1)}(\mathbf{w}))\}}, \quad (7.14)$$

  where $\psi \in \Psi_k$. The needed permutation is found as

$$\psi^\star = \arg\min_{\psi \in \Psi_k} D_k(\gamma_{\mathcal{X}_1^k(\mathbb{S}_1)}, \gamma_{\mathcal{X}_2^k(\mathbb{S}_1)}, \psi). \quad (7.15)$$

  This problem can be expressed as a partial case of the minimum weighed perfect bivariate matching problem and can be solved by means of the Hungarian method (see, Subsection 7.1.3). $\gamma_{\mathcal{X}_1^k(\mathbb{S}_1)}$ is the label function given by the partition $\mathcal{X}_1^k(\mathbb{S}_1)$.

Thus, random occurrences are defined in $1..k$ as:

$$Y_1(\mathbf{w}) = \gamma_{\mathcal{X}_1^k}(\mathbf{w}),\ \mathbf{w} \in \mathbb{S}_1,$$
$$Y_2(\mathbf{w}) = \psi^\star(\gamma_{\mathcal{X}_2^k}(\mathbf{w})),\ \mathbf{w} \in \mathbb{S}_1.$$

Accordingly, random occurrences in the clusters can be accepted as

$$Z_1^i = \{\mathbf{w} \in \mathbb{S}_1 |\ \gamma_{\mathcal{X}_1^k}(\mathbf{w}) = i\},\ i \in 1..k,$$
$$Z_2^i = \{\mathbf{w} \in \mathbb{S}_2 |\ \psi^\star(\gamma_{\mathcal{X}_2^k}(\mathbf{w})) = i\},\ i \in 1..k.$$

A formal disadvantage of this scheme is that predicting cluster membership requires the use of an additional prediction procedure ε_k. Typically, such a procedure is simply defined, for instance, in the case of centroid-based or spectral approaches. Nevertheless, this factor may lead to model indistinctness.

7.5.2 Second Simulation Method

This method has actually been proposed in [335] and [337]. Here, only one clusterer Cl_k is used. Let $\mathbb{S}_1$ and $\mathbb{S}_2$ be two disjoint samples drawn without replacement from the population $\mathbb{W}$.

- We consider
$$\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$$
and construct
$$\mathcal{X}^k(\mathbb{S}) = Cl_k(\mathbb{S}),\ \mathcal{X}_1^k(\mathbb{S}_1) = Cl_k(\mathbb{S}_1),\ \mathcal{X}_2^k(\mathbb{S}_2) = Cl_k(\mathbb{S}_2). \tag{7.16}$$
- Consider occurrences of the samples $\mathbb{S}_1$ and $\mathbb{S}_2$ in the clusters:
$$\mathbb{S}_l^i = \{\mathbf{w} \in \mathbb{S}_l |\ \gamma_l^k(\mathbf{w}) = i\},\ \gamma_l^k(\cdot) = \gamma_{\mathcal{X}_l^k}(\cdot),\ l = 1, 2,\ i \in 1..k. \tag{7.17}$$
- Now we need to coordinate the partitions $\mathcal{X}_l^k(\mathbb{S}_l)$, $l = 1, 2$. Note, that for each one of the samples $\mathbb{S}_l$, $l = 1, 2$, two cluster solutions $\mathcal{X}^k(\mathbb{S}_l)$ and $\mathcal{X}_l^k(\mathbb{S}_l)$ are defined. A matching of $\mathcal{X}_l^k(\mathbb{S}_l)$ with $\mathcal{X}^k(\mathbb{S})$ indirectly provides a matching between $\mathcal{X}_l^k(\mathbb{S}_l)$, $l = 1, 2$. Analogous to (7.14) and (7.15), for $l = 1, 2$ we determine the permutations:
$$\psi_l^\star = \arg\min_{\psi \in \Psi_k} D_k(\gamma_{\mathcal{X}^k(\mathbb{S}_l)}, \gamma_l^k, \psi),\ l = 1, 2. \tag{7.18}$$
Random occurrences defined in $1..k$ can be introduced here on $\mathbb{S}$ as:
$$Y_1(\mathbf{w}) = \gamma_{\mathcal{X}^k(\mathbb{S})}(\mathbf{w}),\ \mathbf{w} \in \mathbb{S},$$
$$Y_2(\mathbf{w}) = \psi_l^\star(\gamma_j^k(\mathbf{w}))),\ \mathbf{w} \in \mathbb{S}_l,\ l = 1, 2.$$
In turn, random occurrences in the clusters are defined as:
$$Z_1^i = \{\mathbf{w} \in \mathbb{S}_1 |\ \psi_1^\star(\gamma_1^k(\mathbf{w})) = i\},\ i \in 1..k,$$
$$Z_2^i = \{\mathbf{w} \in \mathbb{S}_2 |\ \psi_2^\star(\gamma_2^k(\mathbf{w})) = i\},\ i \in 1..k.$$

7.5.3 Third Simulation Method

In the third simulation method occurrences are constructed based upon two different disjoint samples $\mathbb{S}_1$ and $\mathbb{S}_2$ drawn without replacement from the population $\mathbb{W}$ using an auxiliary subset $\mathbb{S}_3 \subset \mathbb{W}$.

- We introduce
$$\mathbb{S}_{1,3} = \mathbb{S}_1 \cup \mathbb{S}_3,\ \mathbb{S}_{2,3} = \mathbb{S}_2 \cup \mathbb{S}_3$$
and construct
$$\mathcal{X}_{1,3}^k(\mathbb{S}_{1,3}) = Cl_k(\mathbb{S}_{1,3}),\ \mathcal{X}_{2,3}^k(\mathbb{S}_{2,3}) = Cl_k(\mathbb{S}_{2,3}).$$

- Now there are two partitions of the set $\mathbb{S}_3$ generated by $\mathcal{X}^k_{1,3}(\mathbb{S}_{1,3})$ and $\mathcal{X}^k_{2,3}(\mathbb{S}_{2,3})$. According to the reasoning applied in the first simulation method we can find a permutation

$$\psi^\star = \arg\min_{\psi\in\Psi_k} D_k(\gamma^k_{1,3}, \gamma^k_{2,3}, \psi)$$

that provides maximal matching between these partitions.

Analogous with the first method, the random occurrences are defined in $1..k$ as:

$$Y_1(\mathbf{w}) = \gamma_{\mathcal{X}^k_{1,3}}(\mathbf{w}),\ \mathbf{w}\in\mathbb{S}_3,$$
$$Y_2(\mathbf{w}) = \psi^\star(\gamma_{\mathcal{X}^k_{2,3}}(\mathbf{w})),\ \mathbf{w}\in\mathbb{S}_3.$$

Consecutively, the random occurrences in the clusters are defined as:

$$Z^i_1 = \{\mathbf{w}\in\mathbb{S}_1|\ \gamma_{\mathcal{X}^k_{1,3}}(\mathbf{w}) = i\},\ i\in 1..k,$$
$$Z^i_2 = \{\mathbf{w}\in\mathbb{S}_2|\ \psi^\star(\gamma_{\mathcal{X}^k_{2,3}}(\mathbf{w})) = i\},\ i\in 1..k.$$

Note that this approach can be considered as being the spirit of the methods used in [34] and [213], that operate on the sets' intersection. Actually, attaching $\mathbb{S}_3$ to $\mathbb{S}_1$ and $\mathbb{S}_2$ in $\mathbb{S}_{1,3}$ and $\mathbb{S}_{2,3}$ can be deemed as the construction of an appropriate intersection that coincides with $\mathbb{S}_3$ if sets $\mathbb{S}_j$, $j = 1, 2, 3$ are disjoint.

7.6 Application of Kernel-Based Distances

One of the widespread types of two sample statistics is formed by the following distances between distributions:

$$D_{\mathrm{K}}(\mathrm{P},\mathrm{Q}) = \left(E\,\mathrm{K}(X,X') + E\,\mathrm{K}(Y,Y') - 2E\,\mathrm{K}(X,Y)\right)^{\frac{1}{2}}, \tag{7.19}$$

where X, Y, X', Y' are mutually independent random variables such that X, X' are drawn according to P, and Y, Y' are drawn according to Q. $\mathrm{K}(\cdot,\cdot)$ is a kernel function meant to represent a geometrical similarity amid points in a Euclidean space in the spirit of the "kernel trick" for distances [286]. From the cluster standpoint such a probability distance appears to be quite reasonable. Indeed, if we interpret the random variables as occurrences of samples in clusters, then $E\,\mathrm{K}(X,X')$ estimates a general cluster "diameter" built by means of the first sample. The values $E\,\mathrm{K}(Y,Y')$ and $E\,\mathrm{K}(X,Y)$ can be analogously explained. Hence, the summarizing distance value $D_{\mathrm{K}}(\mathrm{P},\mathrm{Q})$ constitutes an error in the found "diameter" estimations and its "small" magnitude hints that the cluster structure can be stable.

In machine learning, kernel methods have recently become progressively common instrument. The main reason for this popularity is the "kernel trick"

proposed in [3]. This method is intended to employ a linear algorithm to solve a non-linear problem so that the patterns are embedded into a higher-dimensional space where the problem can be solved by means of a linear classifier. The source dataset is primarily fed into a feature space

$$\phi : \mathbb{W} \to \mathbb{F}, \ \mathbf{w} \mapsto \phi(\mathbf{w}),$$

followed by an evaluation via a dot product

$$\mathrm{K}(\mathbf{w}, \mathbf{w}') = \langle \phi(\mathbf{w}), \phi(\mathbf{w}') \rangle \,. \tag{7.20}$$

As usual, a kernel $\mathrm{K}(\cdot, \cdot)$ is chosen without any concrete information according to the mapping ϕ in order to avoid working in the latent high-dimensional space $\mathbb{F}$. The Mercer's theorem [285] asserts that each continuous, symmetric, positive semi-definite kernel function can be represented as a dot product in a high-dimensional space. The success of *support vector machines (SVM)* introduced by Cortes and Vapnik [86] has extended the application of kernels to many learning areas (e.g., *Kernel PCA* [288]). Kernel methods are frequently used in clustering applications (e.g., [36, 112, 127]). Encyclopedic accounts of kernel methods are available in [287]. Apparently, the most widespread two-sample test in the area of machine learning, is a kernel two-sample test, introduced in [150, 151, 152, 153, 194, 195] and [359].

7.6.1 Real Positive and Negative Definite Kernels

First we state several known facts about real positive and negative definite kernels, that which can be found, for example, in [37, 161, 285] and [287]. Let $\mathbb{W}$ be a finite set.

Definition 7.1. *Let* $\mathrm{K}(\cdot, \cdot)$ *be a symmetric function on* $\mathbb{W} \times \mathbb{W}$*, i.e.,* $\mathrm{K}(\mathbf{x}, \mathbf{y}) = \mathrm{K}(\mathbf{y}, \mathbf{x})$ *for any* $\mathbf{x}$, $\mathbf{y} \in \mathbb{W}$.

- *If for any* $n \geq 1$ *and for any* $\mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathbb{W}$ *the matrix* $\mathbf{K}_{ij} = \mathrm{K}(\mathbf{w}_i, \mathbf{w}_j)$ *is positive definite, i.e.*

$$\sum_{i,j} \mathbf{K}_{ij} c_i c_j \geq 0,$$

  for any real numbers $c_1, \ldots, c_n$, $\mathrm{K}(\cdot, \cdot)$ *is then referred to as a positive definite kernel. If the equality is reached only if* $c_i = c_j = 0$ *then the kernel is referred to as strongly positive definite.*
- *If for any* $n \geq 1$ *and for any* $\mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathbb{W}$

$$\sum_{i,j} \mathrm{N}(\mathbf{w}_i, \mathbf{w}_j) c_i c_j \leq 0$$

  for any real numbers $c_1, \ldots, c_n$ *such that*

$$\sum_{i=1}^{n} c_i = 0,$$

then $\mathrm{N}(\cdot,\cdot)$ *is referred to as a negative definite kernel. If the equality is reached only if* $c_i = c_j = 0$ *then the kernel is referred to as strongly negative definite.*

Note that positive definite kernels are often referred to as Mercer kernels and the term "conditionally positive definite kernel" is used instead of the term "negative definite kernel". Actually, if $\mathrm{N}(\cdot,\cdot)$ is a conditionally positive definite kernel, then $-\mathrm{N}(\cdot,\cdot)$ is a negative definite kernel.

It is well known that

1. A function $\mathrm{N}(\mathbf{x},\mathbf{y})$ on $\mathbb{W}\times\mathbb{W}$ is a negative definite kernel if and only if the function $\mathrm{N}(\mathbf{x},\mathbf{y}) + f(\mathbf{x}) + f(\mathbf{y})$ is also a negative definite kernel for any function f on $\mathbb{W}$;
2. Suppose that $\mathrm{N}(\mathbf{x},\mathbf{y})$ is a negative definite kernel such that $\mathrm{N}(\mathbf{z},\mathbf{z}) = 0$ for some $\mathbf{z}\in\mathbb{W}$ then the function

$$\mathrm{K}(\mathbf{x},\mathbf{y}) = \mathrm{N}(\mathbf{x},\mathbf{z}) + \mathrm{N}(\mathbf{z},\mathbf{y}) - \mathrm{N}(\mathbf{x},\mathbf{y})$$

   is a positive definite kernel.
3. If $\mathrm{K}(\mathbf{x},\mathbf{y})$ is a positive definite kernel then the function

$$\mathrm{N}(\mathbf{x},\mathbf{y}) = \mathrm{K}(\mathbf{x},\mathbf{x}) + \mathrm{K}(\mathbf{y},\mathbf{y}) - 2\mathrm{K}(\mathbf{x},\mathbf{y})$$

   is a negative definite kernel so that $\mathrm{N}(\mathbf{x},\mathbf{x}) = 0$ for all $\mathbf{x}\in\mathbb{W}$.

A connection between the notions of positive and negative definite kernels is also stated as follows:

Definition 7.2. *Let* K *be a positive definite kernel receiving non-negative values. The kernel is referred to as infinitely divisible if for each positive integer* n*, there exists a positive definite kernel* K_n *such that* $\mathrm{K}{=}\mathrm{K}_n^n$.

The following statements are equivalent:

- $\mathrm{K}(\mathbf{x},\mathbf{y}) = exp(-t\mathrm{N}(\mathbf{x},\mathbf{y}))$ is positive definite kernel for all $t>0$;
- $\mathrm{N}(\mathbf{x},\mathbf{y})$ is negative definite kernel.

For the set $\mathbb{W}$ having a linear structure, translation-invariant positive definite kernels of the type $\mathrm{K}(\mathbf{x},\mathbf{y}) = f(\mathbf{x}-\mathbf{y})$ are quite widespread in applications. Here, f is referred to as a positive definite function. If $\mathbb{W}$ is the Euclidean space $\mathbb{R}^d$, then according to the well-known Bochner's theorem (see, for example, [219]) f is a continuous positive definite function satisfying $f(0) = 1$ if and only if f is a characteristic function (the Fourier transform) of a symmetric distribution on $\mathbb{R}^d$. In kernel terminology, a symmetric distribution Q is infinitely divisible if the kernel $\mathrm{K}(\mathbf{x},\mathbf{y}) = f(\mathbf{x}-\mathbf{y},\mathrm{Q})$ is infinitely

divisible and positive definite. Here, $f(\mathbf{x}, \mathrm{Q})$ is the characteristic function of Q. In this case

$$\lambda(\mathbf{x}) = -\log(f(\mathbf{x}, \mathrm{Q})) = -\int \frac{(\cos(i\,\langle t \cdot \mathbf{x}\rangle) - 1)}{\|t\|^2}\mu(dt), \tag{7.21}$$

where μ is a finite symmetric measure on $\mathbb{R}^d$ (e.g., [219, 284]). The kernel $\mathrm{N}(\mathbf{x}, \mathbf{y}) = \lambda(\mathbf{x} - \mathbf{y})$ is negative definite (e.g., [219]). In particular, functions of the type $\psi(\mathbf{x} - \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^\rho$, $0 < \rho < 2$, produce negative definite kernels generated by the logarithms of the characteristic functions of the so-called symmetric stable distributions. These functions are strongly negative definite for $0 < \rho < 2$. The case $\rho = 2$ corresponds to the symmetric Gaussian laws. In probability theory, equation (7.21) is known as *the Levy-Khintchine representation*. Generalizations of this statement have been considered in [338].

A simple intuitive way to build a feature map satisfying (7.20) is based on the following theorem (see, [19]).

Theorem 7.3. *Let* $\mathrm{K}(\mathbf{x}, \mathbf{y})$ *be a positive definite kernel.* **Then** *there exists a unique Hilbert space* $\mathbb{H}(\mathrm{K})$ *having real functions as elements defined on* $\mathbb{W}$ *such that*

- $k_{\mathbf{y}}(\cdot) = \mathrm{K}(\cdot, \mathbf{y}) \in \mathbb{H}(\mathrm{K})$ *for all* $\mathbf{y} \in \mathbb{W}$*;*
- *for each* $\mathbf{y} \in \mathbb{W}$ *and* $\phi \in \mathbb{H}(\mathrm{K})$ *the relationship*

$$\langle \phi, k_{\mathbf{y}}(\cdot)\rangle = \phi(\mathbf{y})$$

holds.

Based upon this theorem we can create a feature map as

$$\phi : \mathbb{X} \to \mathbb{H}(\mathrm{K}),\ \mathbf{x} \longmapsto k_{\mathbf{x}}(\cdot).$$

In particular, according to the definition of the dot product we have

$$\langle k_{\mathbf{x}}(\cdot), k_{\mathbf{y}}(\cdot)\rangle = \mathrm{K}(\mathbf{x}, \mathbf{y})$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{W}$. Thus, positive definite kernels appear to be nonlinear generalizations of the inherent similarity measure created by the dot product. The space $\mathbb{H}(\mathrm{K})$ is known as *a Reproducing Kernel Hilbert Space (RKHS)*.

By a comparable procedure a generalization of dissimilarity measures can be provided.

Theorem 7.4. *Let* $\mathrm{N}(\mathbf{x}, \mathbf{y})$ *be a negative definite kernel satisfying* $\mathrm{N}(\mathbf{x}, \mathbf{x}) = 0$ *for all* $\mathbf{x} \in \mathbb{W}$. **Then** *there exists a Hilbert space* $\mathbb{H}$ *of real valued functions on* $\mathbb{W}$ *and a mapping* $\phi : \mathbb{W} \to \mathbb{H}$ *such that*

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\|^2 = \mathrm{N}(\mathbf{x}, \mathbf{y}).$$

This theorem has been used as a starting point for the "kernel trick" for distances in machine learning described in [286]. An analog to Theorem 7.3 generating Theorem 7.4 has been offered by Schoenberg [285]:

Theorem 7.5. *A separable complete metric space* $(\mathbb{W};\rho)$ *is isometric to a subset of Hilbert space* **if and only if** ρ^2 *is negative definite kernel.*

7.6.2 Two Sample Test Kernel Distances

Positive definite kernel-based distances. The central idea of the two sample kernel-based tests introduced in [150, 151, 152, 153, 194, 195] and [359] is to use *the maximum mean discrepancy (MMD)*, calculated between two distribution images in an appropriate *RKHS*, as the distance amid the distribution. Let us consider an *RKHS* $\mathbb{H}(\mathrm{K})$ on the separable complete metric space $\mathbb{W}$ with a continuous feature mapping $\phi(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{W}$ and a bounded kernel K. A mapping

$$\mu : \mathcal{P}(\mathbb{W}) \to \mathbb{H}(\mathrm{K})$$

is defined as the expectation of $\phi(\mathbf{w})$ with respect to the mapped probability measure:

$$\mu(\mathrm{P}) = \int \phi(\mathbf{x})\mathrm{P}(d\mathbf{x}).$$

Consequently, *MMD* is a distance between two such mappings:

$$MMD^2_{\mathrm{K}}(\mathrm{P},\mathrm{Q}): \ = \|\mu(\mathrm{P}) - \mu(\mathrm{Q})\|^2 = \iint \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_2)\rangle\, \mathrm{P}(d\mathbf{x}_1)\mathrm{P}(d\mathbf{x}_2) +$$
$$+ \iint \langle\phi(\mathbf{y}_1),\phi(\mathbf{y}_2)\rangle\, \mathrm{Q}(d\mathbf{y}_1)\mathrm{Q}(d\mathbf{y}_2) - 2\iint \langle\phi(\mathbf{x}),\phi(\mathbf{y})\rangle\, \mathrm{P}(d\mathbf{x})\mathrm{P}(d\mathbf{y}).$$

Obviously, this can be represented in the form of (7.19). Moreover, this quantity satisfies all the metric properties excluding the identity quality: the relationship $MMD_{\mathrm{K}}(\mathrm{P},\mathrm{Q}) = 0$ does not imply $\mathrm{P} = \mathrm{Q}$. However, this measure is a metric if the kernel is characteristic. This means that $\mathbb{H}(\mathrm{K})$ is built on a compact metric space, and the kernel is dense with respect to the L_∞ norm in the space of bounded continuous functions defined on this space. As was demonstrated in [303], the Gaussian and the Laplace kernels are universal.

Let us suppose we have two i.i.d. samples $\mathbb{X} = \{\mathbf{x}_i\}_{i\in 1..n}$ and $\mathbb{Y} = \{\mathbf{y}_j\}_{j\in 1..m}$ drawn respectably from P and Q. A biased empirical estimate of the *MMD* is

$$MMD^2_{\mathrm{K}}(\mathbb{X},\mathbb{Y}) = \frac{1}{n^2}\sum_{i,j=1}^{n} \mathrm{K}(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{m^2}\sum_{i,j=1}^{m} \mathrm{K}(\mathbf{y}_i,\mathbf{y}_j) - \frac{2}{nm}\sum_{i,j=1}^{n,m} \mathrm{K}(\mathbf{x}_i,\mathbf{y}_j).$$

Accordingly, when $n = m$ an unbiased estimate of *MMD* is the one-sample U-statistic

$$MMD_{\mathrm{K}}(\mathbb{X},\mathbb{Y}) = \frac{1}{n(n-1)}\sum_{i\neq j}^{n} h(Z_i,Z_j),$$

where $Z_i := (\mathbf{x}_i, \mathbf{y}_i)$ and

$$h(Z_i, Z_j) := \mathrm{K}(\mathbf{x}_i, \mathbf{x}_j) + \mathrm{K}(\mathbf{y}_i, \mathbf{y}_j) - \mathrm{K}(\mathbf{x}_i, \mathbf{y}_j) - \mathrm{K}(\mathbf{y}_j, \mathbf{y}_i).$$

Negative definite kernel-based distances. The approach outlined in the previous section can be considered within a slightly more general framework, based on negative definite kernels. The idea of such a method is based on the characterization theorems stated in [194] and [359]. Several applications of this methodology are presented in [33] and [196], while a general clarification can be found in [195].

Let $(\mathbb{W}; \rho)$ be a separable complete metric space. Suppose that $\mathrm{N}(\cdot, \cdot)$ is a real continuous function on $\mathbb{W} \times \mathbb{W}$, and use $\mathcal{P}_{\mathrm{N}}(\mathbb{W})$ to denote the set of all probability measures satisfying the relationship:

$$\int |\mathrm{N}(\mathbf{w}, \mathbf{w})| \, \mathrm{P}(d\mathbf{w}) < \infty.$$

Theorem 7.6. *Let $\mathcal{M}$ be a real symmetric continuous function on $\mathbb{W} \times \mathbb{W}$. The inequality*

$$\mathcal{M}_{\mathrm{N}}(\mathrm{P}, \mathrm{Q}) = 2 \iint \mathrm{N}(\mathbf{x}, \mathbf{y}) \mathrm{P}(d\mathbf{x}) \mathrm{Q}(d\mathbf{y}) - \iint \mathrm{N}(\mathbf{x}, \mathbf{y}) \mathrm{P}(d\mathbf{x}) \mathrm{P}(d\mathbf{y}) - \\ - \iint \mathrm{N}(\mathbf{x}, \mathbf{y}) \mathrm{Q}(d\mathbf{x}) \mathrm{Q}(d\mathbf{y}) \geq 0 \tag{7.22}$$

holds for all $\mathrm{P}, \mathrm{Q} \in \mathcal{P}_{\mathrm{N}}(\mathbb{W})$ if and only if N is a negative definite kernel. Equality is obtained, in the case where $\mathrm{P} = \mathrm{Q}$, **if and only if** *N is a strongly negative definite kernel.*

Proof. Let $\mathrm{R} \in \mathcal{P}_{\mathrm{N}}(\mathbb{W})$ be an arbitrary measure dominating P and Q. Denote

$$h = \frac{d\mathrm{P}}{d\mathrm{R}} - \frac{d\mathrm{Q}}{d\mathrm{R}}.$$

Hence the inequality (7.22) can be rewritten in the form

$$\iint \mathrm{N}(\mathbf{x}, \mathbf{y}) h(\mathbf{x}) h(\mathbf{y}) \mathrm{R}(d\mathbf{x}) \mathrm{R}(d\mathbf{y}) \leq 0. \tag{7.23}$$

The measure R and the function h with zero mean are arbitrary because the measures P, Q are chosen arbitrarily. Thus, (7.22) and (7.23) are equivalent. However, (7.23) is the same as the definition of the negative definite kernel. Further, let us take a negative definite kernel $\mathrm{N}(\mathbf{x}, \mathbf{y})$ such that $\mathrm{N}(\mathbf{x}; \mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbb{W}$. According to the earlier stated properties of the negative definite kernels, the kernel

$$\mathrm{K}(\mathbf{x}, \mathbf{y}) = \mathrm{N}(\mathbf{x}, \mathbf{z}) + \mathrm{N}(\mathbf{z}, \mathbf{y}) - \mathrm{N}(\mathbf{x}, \mathbf{y})$$

is a positive definite kernel for all $\mathbf{z} \in \mathbb{W}$. So,

$$\mathcal{M}_{\mathrm{N}}(\mu,\nu) = \mathcal{L}(\mu,\mu) + \mathcal{L}(\nu,\nu) - 2\mathcal{L}(\mu,\nu), \tag{7.24}$$

where

$$\mathcal{L}(\mu,\nu) = \iint \mathrm{K}(\mathbf{x},\mathbf{y})\mu(d\mathbf{x})\nu(d\mathbf{y}).$$

For two independent random variables X, Y with distribution functions $\mathrm{P}, \mathrm{Q} \in \mathcal{P}_{\mathrm{N}}(\mathbb{W})$ the distance $\mathcal{M}_{\mathrm{N}}$ is represented as

$$\mathcal{M}_{\mathrm{N}}(\mathrm{P},\mathrm{Q}) = \left(2E\,\mathrm{N}(X,Y) - E\,\mathrm{N}(X,X') - E\,\mathrm{N}(Y,Y')\right)^{\frac{1}{2}},$$

where X', Y' are independent copies of X, Y and all random variables X, Y, X', Y' are mutually independent.

Theorem 7.7. *Let* N *be a strongly symmetric negative definite kernel, satisfying* $\mathrm{N}(\mathbf{w},\mathbf{w}) = 0$ *for all* $\mathbf{w} \in \mathbb{W}$. **Then** $\mathcal{M}_{\mathrm{N}}(\mathrm{P},\mathrm{Q})$ *is a distance on* $\mathcal{P}_{\mathrm{N}}(\mathbb{W})$.

Assume now that $(\mathbb{W};\rho)$ is a separable complete metric space such that ρ^2 is a strongly negative definite kernel. As implied by Theorem 7.7, $\mathcal{M}_{\mathrm{N}}(\mathrm{P},\mathrm{Q})$ is a strictly negative definite kernel on $\mathcal{P}_{\mathrm{N}}(\mathbb{W})$. According to Theorem 7.5, the metric space $(\mathcal{P}_{\mathrm{N}}(\mathbb{W}), \mathcal{M}_{\mathrm{N}})$ is isometric to a subset of a Hilbert space. The set $\mathbb{W}$ is naturally identified with the subset of the point measures in $\mathcal{P}_{\mathrm{N}}(\mathbb{W})$, and the considered distance can be described analogously with *MMD* as the distance between the corresponding barycentres in a Hilbert space.

As we have seen, if $\mathrm{K}(\mathbf{x},\mathbf{y})$ is a positive definite kernel, then the kernel $\mathrm{N}(\mathbf{x},\mathbf{y}) = \mathrm{K}(\mathbf{x},\mathbf{x}) + \mathrm{K}(\mathbf{y},\mathbf{y}) - 2\mathrm{K}(\mathbf{x},\mathbf{y})$ is a negative definite one. Hence the previously described *MMD* metric-based construction can be formally embedded into the negative definite kernel-based distances scheme, presented in this section.

7.6.3 Parzen Window-Based Distances

Another commonly used type of kernel is *the Parzen kernel*. Parzen windowing is a density estimation technique based on approximations by means of kernel-function mixtures such that analytical properties of the estimator are inherited from appropriate kernel properties [251]. A suitable Parzen window typically demonstrates the following properties [347]:

- it is unimodal and symmetric;
- it is non-negative;
- it integrates to one.

As usual, operating with a window integrates to one is not essential. Supposedly the window could integrate to a constant, while the scalar factor can be easily eliminated in the estimation process. Parzen windowing is also known as a kernel density estimation method. For a i.i.d. sample $\{\mathbf{w}_t\}_{i\in 1..T}$,

drawn from the true density $f(\mathbf{w})$, the Parzen window estimate for this density is given as

$$\widehat{f}(\mathbf{w}) = \sum_{t=1}^{T} \psi_\sigma(\mathbf{w}, \mathbf{w}_t), \tag{7.25}$$

where $\psi_\sigma(\cdot,\cdot)$ is a Parzen window and a scale parameter σ governs the width of the window. Such estimations can arise in clustering, for example, in information-based clustering approaches [129] and also in attempts to estimate the divergence or "distance" between the overall and cluster densities [275]. In the cluster validation context, however, it is natural to consider the divergence between two cluster density estimations s_1^i, s_2^i obtained by relying upon two samples "drawn" (or simulated) within the cluster:

$$D_i = \int \left(s_1^i(\mathbf{w}) - s_2^i(\mathbf{w})\right)^2 d\mathbf{w},\ i \in 1..k.$$

It is easy to see that

$$D_i = V(s_1^i, s_1^i) + V(s_2^i, s_2^i) - 2V(s_1^i, s_2^i),$$

where $V(s_1^i, s_2^i)$ is a "cross-information potential"

$$V(s_1^i, s_2^i) = \int s_1^i(\mathbf{w}) s_2^i(\mathbf{w}) d\mathbf{w}.$$

The quantity

$$V(f) = \int f^2(\mathbf{w}) d\mathbf{w}$$

is often called the information potential (see, e.g. [268]). It is closely related to the Renyi's quadratic entropy, which is linked by the density function $f(\mathbf{w})$:

$$H(f) = -\log \int f^2(\mathbf{w}) d\mathbf{w}.$$

It is natural to suppose that the divergence value is small for stable clusters. These samples need to be simulated as was described in Section 7.5.

Let us suppose that the densities s_1^i, s_2^i are estimated based on two samples $\mathbb{S}_1, \mathbb{S}_2$ "drawn" from the cluster "i" by means of a Parzen window $\psi_\sigma(\cdot,\cdot)$.

$$\widehat{s}_1^i(\mathbf{w}) = \frac{1}{n_1} \sum_{\mathbf{w}_j \in \mathbb{S}_1} \psi_\sigma(\mathbf{w}, \mathbf{w}_j),\ \widehat{s}_2^i(\mathbf{w}) = \frac{1}{n_2} \sum_{\mathbf{w}_j \in \mathbb{S}_2} \psi_\sigma(\mathbf{w}, \mathbf{w}_j),$$

where $n_j = |\mathbb{S}_l|$, $l = 1, 2$. Hence, we obtain the following estimator for D_i

$$\widehat{D}_i = \frac{1}{(n_1)^2} \sum_{\mathbf{w}_{j_1} \in \mathbb{S}_1} \sum_{\mathbf{w}_{j_2} \in \mathbb{S}_2} \psi_\sigma(\mathbf{w}_{j_1}, \mathbf{w}_{j_2}) + \tag{7.26}$$

$$\frac{1}{(n_2)^2} \sum_{\mathbf{w}_{j_1} \in \mathbb{S}_1} \sum_{\mathbf{w}_{j_2} \in \mathbb{S}_2} \psi_\sigma(\mathbf{w}_{j_1}, \mathbf{w}_{j_2}) - \frac{2}{n_1 n_2} \sum_{\mathbf{w}_{j_1} \in \mathbb{S}_1} \sum_{\mathbf{w}_{j_2} \in \mathbb{S}_2} \psi_\sigma(\mathbf{w}_{j_1}, \mathbf{w}_{j_2}).$$

This expression coincides with the Parzen window-based statistic introduced in [16], while ψ_σ is chosen as a translation invariant Mercer kernel

$$\psi_\sigma(\mathbf{x}, \mathbf{y}) = \mathrm{k}\left(\frac{\mathbf{x} - \mathbf{y}}{\sigma}\right).$$

It appears that Parzen window-based distances of type (7.26) can be interpreted simply from the cluster stability standpoint. However, the consistency of such estimations entails the diminishment of the kernel size σ under increasing sample size. Hence, the corresponding two-sample test converges slowly. Furthermore, the MMD-based tests demonstrate more admirable performance [153].

7.7 The Two-Sample Energy Test

In the current section, we purpose a method for the study of the cluster stability based on a physical point of view. Such standpoint suggests using a physical magnitude of samples mixing within clusters constructed by means of a clustering algorithm. We quantify samples closeness by the relative potential energy between items belonging to different samples for each one of the clusters. This potential energy is closely linked with a "gravity" force between two samples. If the samples within each cluster are well mingled, this quantity is sufficiently small. As known from electrostatics, if the sizes of the samples grow to infinity, then the total potential energy of the pooled samples, tends to zero, in the case of the samples drawn from the same population. *The Two-Sample Energy test* has been constructed based upon this perception [357]. The statistic of the test measures the potential energy of the combined samples. Actually, we use this function as a characteristic of the clustered samples similarity. The partition merit is represented by the worst cluster corresponding to the maximal potential energy value. To ensure readiness of the proposed model and to decrease the uncertainty of the model, we draw many pairs of samples for a given number of clusters and construct an empirical distribution of the potential energy corresponding to the partitions created within the samples. Among all those distributions, one can expect that the true number of clusters can be characterized by the empirical distribution which is most concentrated at the origin. Evidently, the considered method is closely related to the stated earlier kernel based distances approach.

The *Two-Sample Energy test* for the two-sample problem [357] deals with two i.i.d. samples, $\mathbb{X} = \{\mathbf{x}_i\}_{i \in 1..n}$ and $\mathbb{Y} = \{\mathbf{y}_j\}_{j \in 1..m}$. The elements of the first sample are considered to have positive charges equal to $\frac{1}{n}$ and the elements of the second sample are considered to have negative charges equal to $\left(-\frac{1}{m}\right)$. These charges provide a total charge of each sample equal to 1. Let $\Phi_{\mathbb{X},R}$ denote the energy of a charged sample $\mathbb{X}$. This value is calculated as follows:

$$\Phi_{\mathbb{X},R} = \frac{1}{n^2}\sum_{i<j}^{n} R\left(|\mathbf{x}_i - \mathbf{x}_j|\right),\ \Phi_{\mathbb{Y},R} = \frac{1}{m^2}\sum_{i<j}^{m} R\left(|\mathbf{y}_i - \mathbf{y}_j|\right),$$

where the function R is suggested to be a continuous, monotonic decreasing function of the Euclidean distance between the charges. Correspondingly, the intersection energy of two samples $\mathbb{X}$ and $\mathbb{Y}$ is:

$$\Phi_{\mathbb{X},\mathbb{Y},R} = -\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} R\left(|\mathbf{x}_i - \mathbf{y}_j|\right).$$

Respectively, the test statistic $\Psi_{\mathbb{X},\mathbb{Y},R}$ is defined as:

$$\Psi_{\mathbb{X},\mathbb{Y},R} = \Phi_{\mathbb{X},R} + \Phi_{\mathbb{Y},R} + \Phi_{\mathbb{X},\mathbb{Y},R}. \tag{7.27}$$

Common functions for R are:

- $R(r) = -\ln(r)$;
- $R(r) = \frac{1}{r^\alpha},\ \alpha > 0$;
- $R(r) = e^{-r^\alpha},\ \alpha > 0$.

The choice $R(r) = -\ln(r)$ ensures that the test is scale invariant. Although equation (7.27) relies upon physical considerations as opposed to those used in the previous subsections, this equation evidently coincides with the general expression (7.19).

7.7.1 Method Description

Let us consider a finite subset $\mathbb{W}$ of the d-dimensional Euclidean space $\mathbb{R}^d$. As mentioned earlier, we intend to describe cluster stability by means of sampling procedure steadiness. For this purpose, we draw pairs of samples $\mathbb{S}_{1,j}, \mathbb{S}_{2,j}$, $j \in 1..J$, having the same size m to assess the potential energy between their elements within the clusters. To do this we once again estimate the occurrences

$$\mathbb{S}_{l,j}^{i} = \{\mathbf{w} \in \mathbb{S}_{l,j} | \gamma_{l,j}^{k}(\mathbf{w}) = i\},\ l = 1, 2,\ j \in 1..J,\ i \in 1..k,$$

of the samples within the clusters by means of one of the simulation methods stated in Section 7.5.

Following the two-sample test energy methodology, in each group we evaluate the inner potential energy $\Psi_{\mathbb{S}_{1,j}^{i},\mathbb{S}_{2,j}^{i},R},\ i \in 1..k$, according to (7.27). This value characterizes cluster quality. Subsequently, the partition quality is represented by its worst cluster having the maximal inner potential energy:

$$\Psi^{\star}_{\mathbb{S}_{1,j},\mathbb{S}_{2,j},R} = \max_{i}\left(\Psi_{\mathbb{S}_{1,j}^{i},\mathbb{S}_{2,j}^{i},R}\right). \tag{7.28}$$

Another possibility would be to determine the average of $\Psi_{\mathbb{S}_{1,j}^{i},\mathbb{S}_{2,j}^{i},R}$ among all k occurring clusters. However, the first possibility seems to be more stable.

Next, we consider the distributions of $\Psi_{\mathbb{S}^i_{1,j},\mathbb{S}^i_{2,j},R}$ constructed by a multitude of samples for a number of clusters in the range $k \in 2..k_{\max}$, where $k_{\max}$ is the maximal number of tested clusters. To view these distributions in the same scale, we conduct a normalization. In our approach, we divide the range of the distances values into g equal range subgroups and characterize the concentration by the frequency $N_{k,g}$ of the lowest subgroup that is expected to provide the greatest value of $N_{k,g}$ in the case of the true number of clusters.

An algorithm implementing this procedure consists of the following steps:
Algorithm:

Input:
$\mathbb{W}$ — the data to be clustered;
$k_{\max}$ — maximal number of clusters to be tested;
J — number of the drawn sample pairs;
m — the samples size;
Cl_k — a clustering algorithm;
R — a distance function;
g — number of range subgroups.
Output: The "true" number of clusters $k^\star$.

1. For $k = 2$ to $k_{\max}$
2. For $j = 1$ to J do
3. $\mathbb{S}_{1,j} = sample(\mathbb{W}, m), \mathbb{S}_{2,j} = sample(\mathbb{W} \backslash \mathbb{S}_{1,j}, m)$;
4. Perform a simulation step to obtain the occurrences $\mathbb{S}^i_{1,j}, \mathbb{S}^i_{2,j},\ i \in 1..k$, of the samples in the clusters;
5. Measure the potential energy inside the clusters;
6. Calculate the partition quality according to (7.28);
7. end For j;
8. Calculate $N_{k,g}$;
9. End For k;
10. The "true" $k^\star$ is selected as the one which yields the maximal value of $N_{k,g}$.

7.7.2 Numerical Experiments

We exemplify the described approach by means of numerical experiments on synthetic and real datasets provided for 3 functions $R(r)$ mentioned earlier. We perform 10 trials for each experiment, using the second simulation method. The results are shown via the error-bar plots of $N_{k,10}$ within the trials. The sizes of the error bars are equal to two standard deviations, found inside the trials. The standard k-means algorithm is employed.

Synthetic Data. In the first example the datum is simulated as a mixture of 5 two-dimensional Gaussian distributions with independent coordinates owning the same standard deviation $\sigma = 0.35$. The components means are

placed on the unit circle with the angular neighboring distance of $2\pi/5$. The dataset contains 4000 items. We choose $k_{\max} = 7$, $J = 200$ and $m = 100$ in all tests. The results shown in Figure 7.3 demonstrate that for this combination of parameters and kernels, a five-cluster structure is clearly indicated.

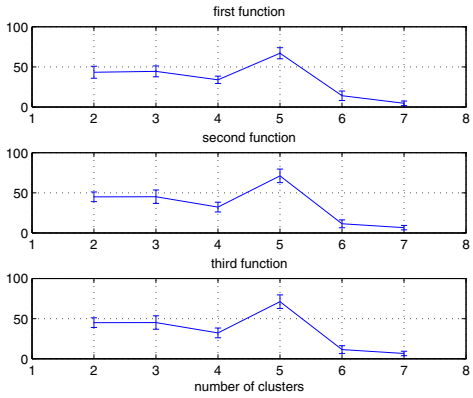

Fig. 7.3. Error-bar plots of $N_{k,g}$ for the five components simulated data

Real-World Data. The second example is the real dataset texts collection $T3$ described in Appendix A.2.1. We take $J = 100$ and $m = 100$. The results depicted in Figure 7.4 show that the number of clusters is properly determined for all functions $R(r)$.

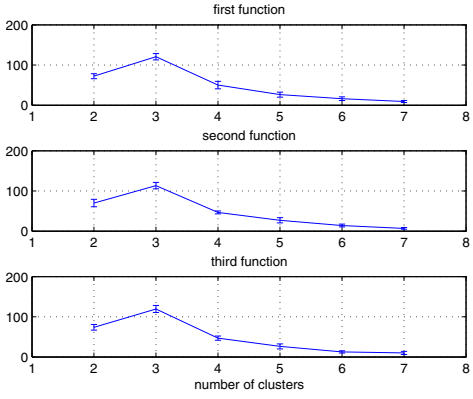

Fig. 7.4. Error-bar plots of $N_{k,g}$ for the three text collection dataset

7.8 Application of Minimal Spanning Tree Methodology

This subsection discusses two different methods for the study of cluster stability stated in [30] and [339]. These methods suggest the geometrical stability of the partition under consideration with respect to an inner stability model. Actually, the Friedman and Rafsky two-sample test statistic [119] is used. In our applications this statistic is expressed by the number of edges connecting points from different samples in a *Minimal Spanning Tree* (*MST*) constructed for each of the clusters. This number is sufficiently large if the samples within each cluster, are well mingled, as illustrated in Figure 7.2. Under this null hypothesis on the homogeneity of the source data, this statistic is approximately normally distributed. Hence, the case of well mingled samples within the clusters leads to a normal distribution of the considered statistic, indicating that the provided stability model assumes a normal distribution of this quantity of edges connecting points from different samples in the clusters. The actual distribution of such edges is expected to be close to a normal one in order to exhibit cluster goodness. We evaluate this closeness in two different ways.

7.8.1 Friedman-Rafsky's *MST* Two-Sample Test

As mentioned earlier, the cluster stability model is based here on the *Friedman-Rafsky's MST two-sample test* within the general methodology mentioned in Section 7.4.

Recall that a spanning tree is a graph that reaches out to all n vertices of the graph and has no loops. Among all the spanning trees of a weighted and connected graph, the one (or possibly more) with the least total weight is called a *minimal spanning tree* (*MST*). A *MST* can be built in $\mathcal{O}(n^2)$ time (including distance calculations) using the Prim, Kruskal, Borůvka or Dijkstra algorithms (e.g., [246]). *MST*s are prevalent because they can be quickly and easily computed and are capable of creating sparse sub-graphs that reflect some essence of the given set.

To apply the test consider $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$ the disjoint union of the samples $\mathbb{S}_1 := \{\mathbf{w}_1, \mathbf{w}_2 \ldots, \mathbf{w}_n\}$ and $\mathbb{S}_2 := \{\mathbf{w}_{n+1}, \mathbf{w}_{n+2} \ldots, \mathbf{w}_{n+m}\}$.
Let $D := \left\{\rho_{i,j} = \rho(\mathbf{w}_i, \mathbf{w}_j) \,|\, i, j \in 1..n+m\right\}$ be the Euclidean distances between all points of $\mathbb{S}$. If all values of $\rho_{i,j}$ are distinct, the set is referred to as *nice*. For a nice set, a unique *MST* exists. On this *MST*, we define a test statistic R_{mn} which is equal to the number of *MST* elements connecting points belonging to different samples. Actually, Friedman and Rafsky [119] offered the statistic $1 + R_{mn}$ which counts the number of disjoint sub-trees resulting from removing all edges that unite vertices of different samples. Henze and Penrose [164] analyzed the asymptotic behavior of R_{mn}.

Theorem 7.8. *(Theorem 1 [164])*
Let

- $m \to \infty$ *and* $n \to \infty$ *such that* $m/(m+n) \to p \in (0,1)$*;*
- *for each* m *and* n *the set* $\mathbb{S}$ *is nice*

The following relationship **then** *holds:*

$$\frac{1}{\sqrt{(m+n)}}\left(R_{mn} - \frac{2mn}{m+n}\right) \to \mathcal{N}\left(0, \sigma_d^2\right). \tag{7.29}$$

Here the convergence is in the distribution, and $N(0, \sigma_d^2)$ denotes the normal distribution with a zero expectation and a variance of

$$\sigma_d^2 = r\left(r + C_d(1-2r)\right),$$

where $r = 2p(1-p)$ and C_d is a constant depending only on the space dimension. Note that if $p = q$, the dispersion σ_d^2 is independent of d.

7.8.2 First Approach

Model description. Let $\mathbb{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T\}$ be a finite subset of the d-dimensional Euclidean space $\mathbb{R}^d$, drawn according to an underlying probability density function $\mathrm{P}(\mathbb{W})$ defined in (6.9). We assume that the source partition corresponding to the true number of clusters satisfies the condition

$$p_1 = p_2 = \ldots = p_k. \tag{7.30}$$

This condition is commonly used in cluster analysis, as mentioned in Section 6.4.

To illustrate the use of the proposed method let us again consider the examples shown in Figure 7.2. The left picture presents a situation where the samples again define actually two different clusters. Thus, there is only one edge in *MST* built within the cluster that connects between the items belonging to different clusters. The right picture demonstrates a consistent cluster where the number of "red" edges is sufficiently large. Thus, a large number of the red edges inside a cluster indicates its consistency. Additionally, according to the Friedman-Rafsky's *MST* test, this quantity is approximately normally distributed for sufficiently large samples if the elements of the samples are well mixed within the cluster. Subsequently, large values of the standard score of the "red" edge quantities can be considered standardized values intended for comparison for different number of clusters.

Generally, the approach can be formalized as follows: Two disjoint samples $\mathbb{S}_1$ and $\mathbb{S}_2$ are drawn from $\mathbb{W}$ without replacement as they have the same size m. Suppose that a clustering algorithm Cl_k is available such that its input parameters include a clustered dataset $\mathbb{S}$ and the number of clusters k. The output parameter is a data partition:

$$\mathcal{X}^k(\mathbb{S}) = Cl_k(\mathbb{S}_1 \cup \mathbb{S}_2). \tag{7.31}$$

The *MST* distances

$$r^i(\mathbb{S}_1, \mathbb{S}_2) = R(\mathbb{S}_1^i, \mathbb{S}_2^i), \qquad i \in 1..k \tag{7.32}$$

are calculated within each cluster $i \in 1..k$ where $\mathbb{S}_1^i, \mathbb{S}_2^i$ are occurrences of the samples $\mathbb{S}_1$ and $\mathbb{S}_2$ in cluster number i:

$$\mathbb{S}_l^i = \mathbb{S}_l \cap \mathbb{X}_i, \quad l = 1, 2, \quad i \in 1..k. \tag{7.33}$$

Referring to assumption (7.30), we can conclude that for sufficiently large values of m the following relation is approximately true:

$$\left|\mathbb{S}_l^i\right| \approx \frac{m}{k}, \quad l = 1, 2, \quad i \in 1..k.$$

Hence, from (7.29) under the homogeneity hypothesis we get the values

$$Y^i = \sqrt{\frac{k}{2m}} \left(r_i(\mathbb{S}_1^i, \mathbb{S}_2^i) - \frac{m}{i} \right), \quad i \in 1..k \tag{7.34}$$

are approximately normally distributed with zero mean and the same standard deviation.

Note that the samples used in the Friedman-Rafsky *MST* test have to be i.i.d. This is unfortunately not the case here because $\mathbb{S}_1^i, \mathbb{S}_2^i$, $i \in 1..k$ were obtained by clustering their unit. On the other hand, it is possible to simulate these portions as independent variables by rerunning the clustering algorithm, as was suggested by one of the simulation methods described in Section 7.5. The simulated sub-samples $\mathbb{S}_1^i, \mathbb{S}_2^i$, $i \in 1..k$ are used to calculate the standardized *MST* distances Y_i according to (7.34). Obviously, these values characterize the clusters qualities.

To achieve partition goodness, an integral index able to summarize these qualities must be used. For example, the mean value can be available in this context. However, we prefer to apply

$$Z^k = \min_{i \in 1..k} \{Y_i\}, \tag{7.35}$$

which characterizes a partition by its "worst" cluster. From this standpoint, an empirical distribution of the index found for different values of k is intended to gain the shortest left tail in the case of a "true" number of clusters. We seek to quantify the tail behavior by means of an asymmetry index.

The best known and most widespread parameter that expresses the lack of distribution symmetry around the sample mean is the skewness statistic. This may have a positive bias in our situation. However, the given experiments show that the sample percentiles demonstrate more stable behavior. Specifically, we use the 25-th percentile (the first quartile), the 75-th percentile (the third quartile) and the 90-th percentile to exhibit the distribution asymmetry.

An algorithm that implements our procedure consists of the following steps:

Algorithm:

Input parameters:
$\mathbb{W}$ — dataset;
$k_{\max}$ — maximal number of cluster to be tested;
J — number of pairs of the drawn sample;
m — samples size;
Cl_k — clustering algorithm;
I_k — asymmetry index.
Output: a "true" number of clusters $k^\star$.

1. For $k = 2$ to $k_{\max}$
2. For $j = 1$ to J
3. $\mathbb{S}_{1,j} = sample(\mathbb{W}, m)$;
4. $\mathbb{S}_{2,j} = sample(\mathbb{W} \backslash \mathbb{S}_{1,j}, m)$;
5. Perform a simulation step;
6. Calculate Y_j^i, $(i \in 1..k)$ according to (7.34);
7. Calculate Z_j^k according to (7.35);
8. end For j;
9. Calculate of an asymmetry index (percentiles) I_k of the collection $\{Z_j^k\}_{j \in 1..J}$;
10. end For k;
11. The "true" $k^\star$ is selected as the one which yields the maximal value of the asymmetry index I_k.

7.8.3 Numerical Experiments

To estimate the capabilities of the described technique, we carried out various numerical experiments on synthetic and real datasets. We chose $k_{\max} = 7$ in all tests except for the last small dataset, and we provided 10 trials for each experiment. For simulation purposes, the second one detailed in Subsection 7.5.2 was chosen. The results are presented via the error-bar plots of the sample percentile means within the trials. The sizes of the error bars are equal to two standard deviations found inside the results trials . The standard version of the *Partitioning Around Medoids* (*PAM*) algorithm (see [187]) has been used for clustering. In comparison to the well-known *k-means* algorithm, *PAM* appears to be more robust (e.g., [99]).

Synthetic Data. The synthesized data are mixtures of $k^\star$ two-dimensional Gaussian distributions with independent coordinates owning the same standard deviation σ. The mean values of the components are placed on the unit circle with the angular neighboring distance of $2\pi/k^\star$. Each dataset contains 4000 items. Here, we take $J = 100$ and $m = 200$.

Four-Component Data Set. Figure 7.5 shows the first dataset with the parameters $k^\star = 4$ and $\sigma = 0.3$. The components overlap slightly in this case.

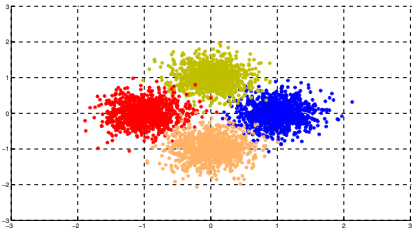

Fig. 7.5. Scatter-plot of the four components simulated data

As seen in Figure 7.6, all three indexes clearly indicate four clusters.

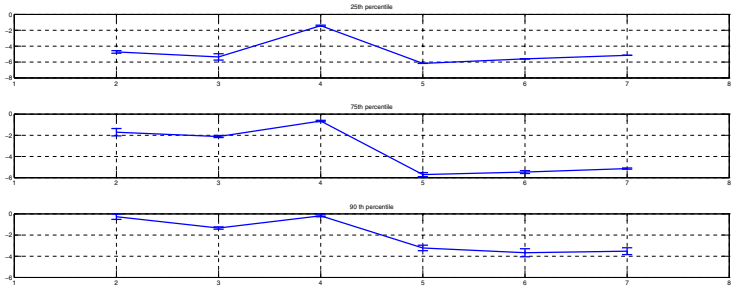

Fig. 7.6. Error-bar plots of the percentiles for the four-component simulated data

Five-Component Data Set. The second dataset has the parameters $k^\star = 5$ and $\sigma = 0.3$. The scatter-plot of such a dataset is shown in Figure 7.7. The components obviously overlap in this case. In Figure 7.8, the true number of clusters has also been successfully found by all indexes in this case, where $J = 100$ and $m = 200$.

Real-World Data. *Three-Text Collection.* The first real dataset is the three-text collection described in Appendix A.2.1. Here we demonstrate the results obtained for the data constructed using the best 600 terms with the parameters $J = 100$ and $m = 200$. As Figure 7.9 shows, the indexes obtain their maximal values at $k = 3$, i.e., the number of clusters is properly determined.

For the "blurred" version of this dataset described in Appendix A.2.2, we can expect that clusters are separated less accurate here. Nevertheless,

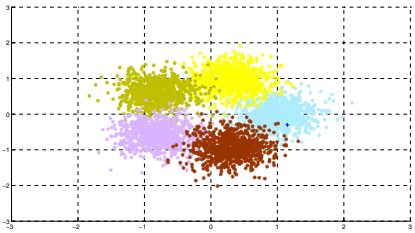

Fig. 7.7. Scatter-plot of the five-component simulated data

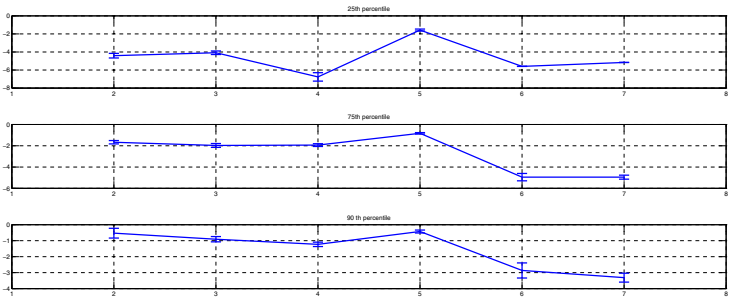

Fig. 7.8. Error-bar plots of the percentiles for the five-component simulated data

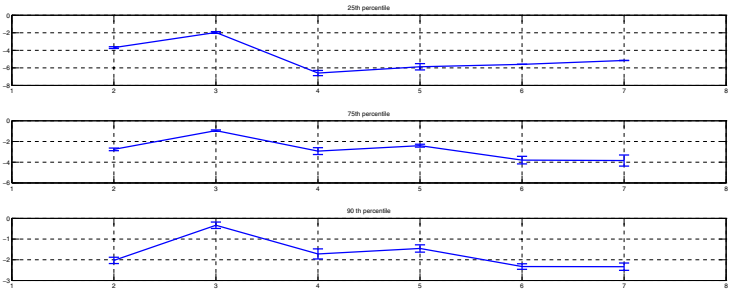

Fig. 7.9. Error-bar plots of the percentiles for the three-text collection dataset presented by 600 terms

the provided experiments demonstrate that our method determines the true number of clusters for this dataset as well (see, Figure 7.10).
Iris Flower Dataset. Another considered dataset is the famous Iris Flower Dataset described in Appendix A.2.3.

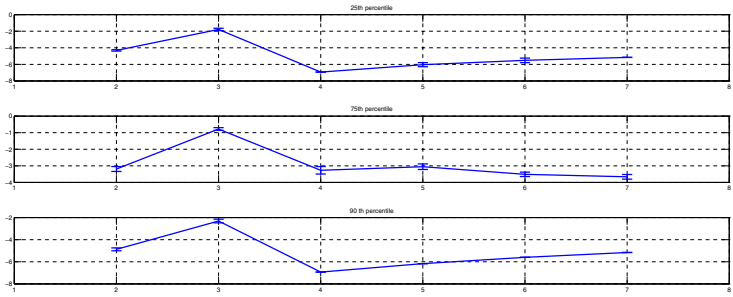

Fig. 7.10. Error-bar plots of the percentiles for the three-text collection dataset presented by 300 terms

The proposed method offers a three-cluster structure for the parameters $J = 200$ and a sample size equal to 70. The results are shown in Figure 7.11. As we can be seen, the true number of clusters is also accurately detected.

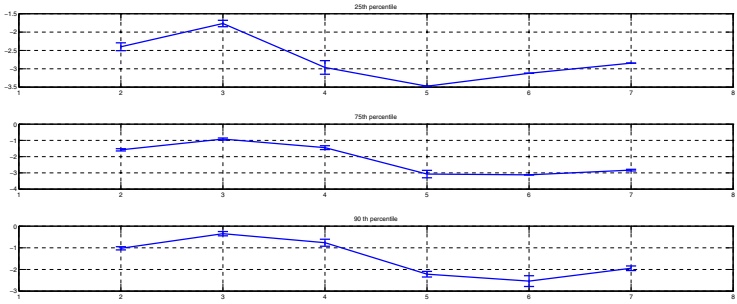

Fig. 7.11. Error-bar plots of the percentiles for the Iris dataset

Flea Beetles. The third considered dataset is the Flea Beetles dataset outlined in Appendix A.2.6.

Compared to the previously considered datasets, this dataset appears to be very small. For this reason we restrict the tested number of clusters to $k_{\max} = 5$, in order to avoid dealing with very small clusters that often appear inherently stable. Note that our method is based on an asymptotic model. Yet, our methodology is also successful in the considered case and is able to find the true number of clusters. We choose $J = 100$ and the sample size equals 30 (see Figure 7.12).

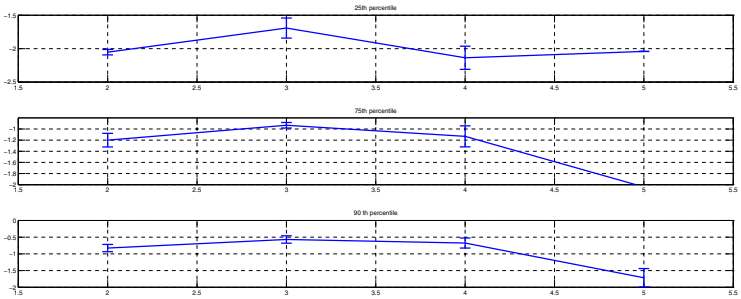

Fig. 7.12. Error-bar plots of the percentiles for the Flea Beetles dataset

7.8.4 Second Approach

In this subsection we discuss another approach related to the one presented earlier. Our aim is to quantify the distances between the cluster elements and the cluster core. We conform to the notion in Hartigan [160] according to which the clusters are recognized as islands in the data ocean. From this point of view, with high density zones defining the clusters it is natural to foresee that in the case of a stable partition clusters characterized by high density zones must be similar to those characterized by the entire set of data. The probability metrics-distances produced by two-sample test statistic are used to measure the dissimilarity between clusters. The distances are calculated in clustered samples drawn from the source population according to two distributions. The first is constructed so that it represents the cluster high density cores and the second is constructed so that it represents the the entire data set. Namely, we compare pairs of partitions, and a pair is considered to be consistent if the obtained clusters match. Similar to the approach described in the previous subsection, this matching is estimated via a *MST* constructed for each one of the clusters, followed by computation of the number of edges connecting the points from the two samples.

Model Description. Within our model we apply a clustering algorithm to a set that is the union of two samples. The first sample is drawn from the dataset $\mathbb{W}$ according to the "cores distribution" and has the probability density function

$$h_{\mathbb{W}}(\mathbf{w}) = exp(a\, f_{\mathbb{W}}(\mathbf{w}))/c(a) \quad (\mathbf{w} \in \mathbb{W}),$$

where $a > 0$ is a constant. The second sample is drawn from the dataset $\mathbb{W}$ according to the "cores surroundings distribution" and has the probability density function

$$g_{\mathbb{W}}(\mathbf{w}) = exp(-a\, f_{\mathbb{W}}(\mathbf{w}))/c(-a) \quad (\mathbf{w} \in \mathbb{W}),$$

where

$$c(a) = \sum_{\mathbf{w} \in \mathbb{W}} exp(a\, f_{\mathbb{W}}(\mathbf{w})).$$

Let us suppose that we have a family of partitions $\{\mathcal{X}^k\}_{k \in 2..k_{\max}}$, where $k_{\max}$ is some predefined number representing the maximal number of clusters under consideration. The proposed approach conjectures that within the partitions built for each possible number of clusters, the most stable one appears when the elements of the two samples are maximally close inside the clusters. In other words, the differences between the samples inside the clusters are expected to be minimal if the number of clusters is chosen correctly.

We apply Theorem 7.8 mentioned previously to characterize cluster stability. This theorem implies that the problem can be reduced to estimation of distances from the empirical distribution of the statistic R_{mn} to a normal distribution for the possible number of clusters. The true number of clusters is assumed to correspond to the minimal distance.

Let us take two disjoint samples $\mathbb{S}_{1,j}, \mathbb{S}_{2,j}, j = 1, 2, ...$ with the the cardinality m_j and n_j respectively and introduce:

$$R^{k,0}_{m_j,n_j}(\mathbb{S}_{1,j}, \mathbb{S}_{2,j}) = \min_{i \in 1..k} \left\{ \sqrt{\frac{k}{2\,(m_j+n_j)}} \left(r^i_{m_j,n_j} - \frac{2m_j n_j}{k(m_j+n_j)} \right) \right\} \quad (7.36)$$

where $r^i_{m_j,n_j}$ is the value of the *MST* statistic calculated within the cluster number i, $i \in 1..k$, in the partition $\mathcal{X}^k_j\,(\mathbb{S}_{1,j} \cup \mathbb{S}_{2,j})$, similar to (7.32)–(7.34).

Once a convergence of partitions $\left\{\mathcal{X}^k_j\right\}$ $j = 1, 2, \ldots$ is considered, we have to assume that the labeling of their elements changes at each step by a suitable sequence of permutations ψ_{m_j} of the label set $1..k$.

Theorem 7.9. *For a given number of clusters k:*

1. *There exists a partition $\mathcal{X}^k_\star$ of the set $\mathbb{W}$ such that, for each sequence of samples $\{\mathbb{S}_j\}$, the sequence of partitions $\{\mathcal{X}^k_j(\mathbb{S}_j)\}$:*

$$\mathcal{X}^k_j(\mathbb{S}_j) = Cl_k(\mathbb{S}_j),$$

converges to $\mathcal{X}^k_\star$, for a sequence ψ_j of permutations of the label set $1..k$. The convergence is such that

$$\lim_{|\mathbb{S}_j| \to \infty} \sum_{i=1}^{k} \left| \mathcal{X}^{\psi_j}_j \oplus \mathcal{X}^i_\star \right| = 0.$$

Here $\oplus$ denotes the symbol XOR (the symmetric difference operation).

2. *The partition $\mathcal{X}^k_\star$ satisfies* (7.30).

Then, *for two sequences of samples $\{\mathbb{S}_{1,j}\}$, $\{\mathbb{S}_{2,j}\}$, having equal size $n_j \to \infty$, such that $\mathbb{S}_j = \mathbb{S}_{1,j} \cup \mathbb{S}_{2,j}$ is nice for each j, the following random variable converges in distribution,*

$$R^{k,0}_{n_j,n_j}(\mathbb{S}_{1,j},\mathbb{S}_{2,j}) \to G_k,$$

where G_k is a random variable representing the minimal value of k independent identically distributed standard normal variables.

Proof. We consider a sequence of equal sized pairs of samples $|\mathbb{S}_{1,j}| = |\mathbb{S}_{2,j}| = n_j$ with nice union sets $\mathbb{S}_j$. Let $\{\mathbb{X}^i_j\}_{i\in 1..k,\ j=1,2,\ldots}$ be a sequence of clusters defined by $\mathbb{S}_j$. Based upon the *MST*, constructed for $\mathbb{X}^i_j$, we introduce the variables

- V^i_j — the number of edges connecting points from different samples belonging to $\mathbb{X}^i_\star$.
- S^i_j — the number of edges connecting points from different samples where at least one of the points does not belong to $\mathbb{X}^i_\star$.

Thus, the value of the *MST* statistic, built within $\mathbb{X}^{\psi_j(i)}_j$, equals

$$M^i_j = V^i_j + S^i_j$$

and

$$\left(V^i_j - R^{k,0}_{n_j,n_j}(\mathbb{S}_{1,j},\mathbb{S}_{2,j})\right) \to 0$$

in distribution. Moreover, due to the conditions mentioned above

$$\lim_{j\to\infty} \left|\mathbb{S}_{1,j} \cap \mathbb{X}^{\psi_j(i)}_j\right| = \lim_{j\to\infty} \left|\mathbb{S}_{2,j} \cap \mathbb{X}^{\psi_j(i)}_j\right| = \frac{n_j}{k}$$

for all $i \in 1..k$ and a sequence ψ_j of permutations of the label set $1..k$. The theorem's assertion is now obtained by substituting the last equalities in the statement of Theorem 7.8.

Note that the convergence condition of Theorem 7.9 is indeed a partial case of a general definition related to the convergence of clustering algorithms (see, for example, [221], section 5.2). For the k-means approach such a property has been discussed in [212] and [256].

For the distribution of the values $R^{k,0}_{n_j,n_j}(\mathbb{S}_{1,j},\mathbb{S}_{2,j})$, the proposed model can be considered as a theoretical etalon characterizing a stable partition situation. According to the model, the *MST* test is iterated many times, for each potential number of clusters, to yield an empirical *MST* test statistic distribution. The distribution, that is closest to normal is assumed to correspond to the true number of clusters.

Let us draw, without replacement, J pairs of samples $\mathbb{S}_{1,j}$ and $\mathbb{S}_{2,j}$, $j \in 1..J$, having the size n_j according to the densities $h_{\mathbb{W}}$ and $g_{\mathbb{W}}$, respectively. We define

$$\bar{r}^i_j = \frac{1}{2}\sqrt{\frac{k}{n_j}}\left(r^i_{n_j,n_j} - \frac{n_j}{k}\right),\ j = 1,\ldots,J;\ i = 1,2,\ldots,k$$

and

$$\tilde{r}_j = \min_{i \in 1..k} \left\{ \bar{r}_j^i \right\}.$$

Thus, under the assumptions in Theorem 7.9, the random variable $\tilde{r}_j$ is distributed as the minimal value of k i.i.d. random variables, distributed according to a normal distribution $\mathcal{N}(\mu, \sigma)$ if k is chosen correctly. In order to estimate the mean μ and the variance σ^2 we calculate the average value of $\bar{r}_j^i$ $(i \in 1..k)$:

$$V_j = Average\{r_j^i \mid i \in 1..k\}.$$

From the J averages V_j $(j \in 1..J)$, their $mean(V)$ and their variance $\{var(V)\}$ are obtained. To this end we substitute:

$$\mu = mean\,(V),\ \sigma^2 = var(V).$$

As a result, we conclude that the estimate of the "true" number of clusters maintains a distribution of the values $\tilde{r}_j$ that is maximally close to the distribution of the minimum of k independent normal variables with the found μ and σ. For this purpose we simulate J values

$$s_j = \min_{i \in k} \{Z^i\}$$

where Z^i, $i \in 1..k$ are the appropriate normal i.i.d. variables. It is well known that the distribution function $\mathrm{P}_{s_j}(\mathbf{w})$ of s_j can be expressed as:

$$\mathrm{P}_{s_j}(\mathbf{w}) = 1 - (1 - \mathrm{P}_{(\mu,\sigma)}(\mathbf{w})),$$

where $\mathrm{P}_{(\mu,\sigma)}(\mathbf{w})$ is the distribution function of $\mathcal{N}(\mu, \sigma)$. These two possibilities often provide similar outcomes; however, we prefer to compare two empirical distributions since approximations of $\mathrm{P}_{(\mu,\sigma)}(\mathbf{w})$ can yield additional calculation instabilities. Therefore, we apply one-dimensional two-sample test statistics to assess the distance between the empirical distributions of $\tilde{r}_j$ and s_j. This can be accomplished by means of the famous Kolmogorov-Smirnov distance, often called the KS distance.

Remarks: 1. We indirectly assume that the sets $\mathbb{S}_{1,j}$ and $\mathbb{S}_{2,j}$ are permanently nice. This suggestion holds for almost all real datasets.

2. The notion of exploring differences between the high and low density regions by the *Friedman-Rafsky's MST* statistic is suggested in [176] and [297]. The goal was to point out an "inconsistent" edge whose length is significantly greater than the average length of the nearby edges. We apply the *MST* two-sample test statistic in the "departure from normality" manner that allows attainment of more detailed results about the data cluster structure.

3. The high and low density regions have been characterized in [176] and [297] via the *MST*, which appears to be computationally costly for large datasets. We use a variant of the nearest neighbors approach for the underlying density estimation.

7.8.5 Experimental Results

Several numerical experiments on synthetic and real datasets were implemented to evaluate the suggested approach. Performance of the proposed method is demonstrated by comparing the obtained results to the assumed "true" structure for two simulated and three real datasets. Estimates of the core density h and the core surroundings density g are constructed by replacing $f(\mathbf{w})$ by its estimate, obtained via a variant of the K-Nearest Neighbors approach as follows:

For each point $\mathbf{w} \in \mathbb{W}$ we determine its K nearest neighbors:

$$\mathbb{Y}_K = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_K\}$$

where K is chosen as the minimal value between 100 and $|\mathbb{W}/2|$. A sphere radius is calculated by

$$R = \frac{1}{10K} \sum_{i=1}^{K} \|\mathbf{w} - \mathbf{y}_i\|$$

in order to estimate the value of $f(\mathbf{w})$ as follows

$$\tilde{f}(\mathbf{w}) = \frac{|\mathbf{y} \in \mathbb{Y}_K : \|\mathbf{w} - \mathbf{y}\| < R|}{|\mathbb{W}|}.$$

We provide 10 trials for each experiment. The results are presented via the error-bar plots of the KS-distance mean within the trials. The sizes of the error bars are two standard deviations, calculated within the trials. The regular k-means algorithm is selected as the clustering algorithm. We choose $a = \ln(4)$ and the maximal tested number of clusters is $k_{\max} = 7$.

Artificial Datasets. These datasets have been simulated as 4000 points drawn from a mixture of two-dimensional, spherical Gaussian distributions with the same standard deviation. The first dataset consists of three components, centered at the coordinates $(\cos(\frac{2\pi i}{3}), \sin(\frac{2\pi i}{3}))$, $i = 0, 1, 2$, and having standard deviations of $\sigma = 0.4$. Two components include 1333 items and the last one includes 1334 items.

A Scatter plot of this data is shown in Figure 7.13.

The error-bar plot for $J = 100$ and a sample size of $m = 225$ is shown in Figure 7.14.

The second dataset was constructed as a mixture of five equally sized spherical Gaussian components, centered at the points $(\cos(\frac{2\pi i}{5}), \sin(\frac{2\pi i}{5}))$, $i = 0, 1, 2, 3, 4$, and having standard deviations of $\sigma = 0.2$.

The error-bar plot for $J = 300$ and a sample size of $m = 700$ is shown in Figure 7.16.

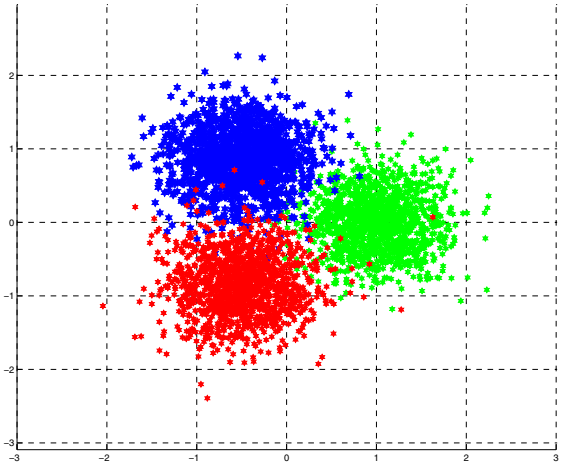

Fig. 7.13. Scatter plot of the three-component simulated data

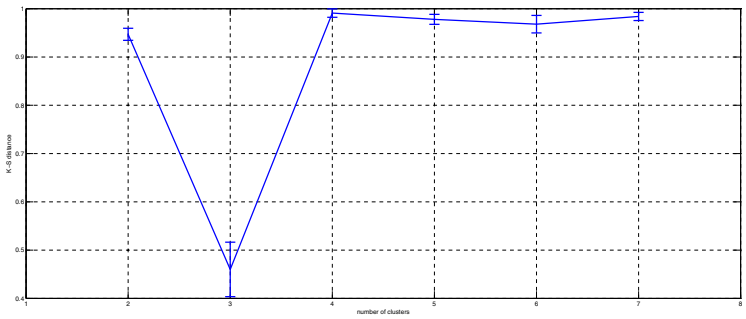

Fig. 7.14. Error-bar plot of KS distance for three-component simulated data

Real-World Data. *Three-Text Collection Datasets.* The first real dataset is the three-text collection described in Appendix A.2.1. Here we choose $J = 100$ and the sample size equals $m = 300$.

As Figure 7.17 shows, the indexes receive their maximal values at $k^\star = 3$, i.e., the number of clusters is properly determined.

For the "blurred" version of this dataset described in Appendix A.2.2, unstable outcomes are expected. However, as Figure 7.18 shown, the true number of clusters has again been clearly detected.

The Iris Flower Dataset. Another considered dataset is the famous Iris flower dataset described in Appendix A.2.3.

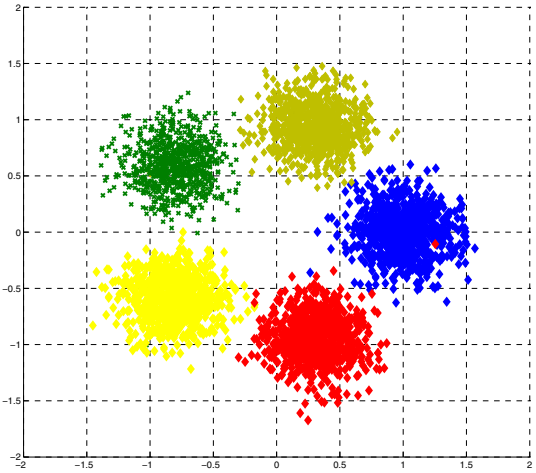

Fig. 7.15. Scatter plot of five-component simulated data

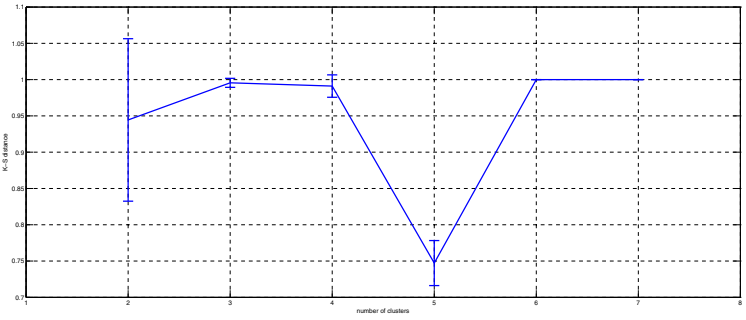

Fig. 7.16. Error-bar plot of the KS distance five-component simulated data

Our approach is able to identify a three-cluster structure using the parameters $J = 200$ and a sample size of $m = 70$. The results are shown in Figure 7.19.

7.9 Binomial Model and the $K - NN$ Approach

In this section, we consider the cluster stability problem by using the probabilistic characteristics of *the K-Nearest Neighbors ($K - NN$) approach*. In spite of straightforward nature of this the methodology, the proposed approach is capable of providing correct outcomes for even relatively small datasets.

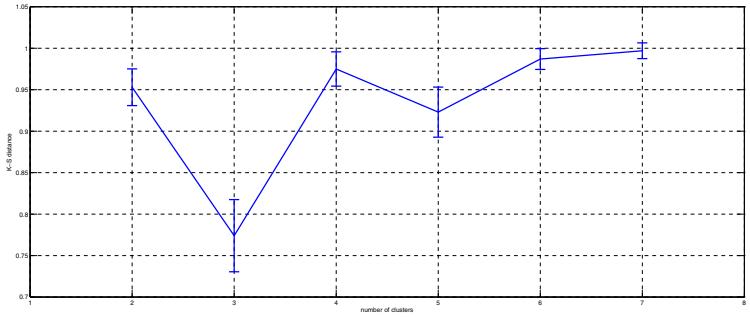

Fig. 7.17. Error-bar plot of the KS distance for the three-text collection dataset for 600 terms

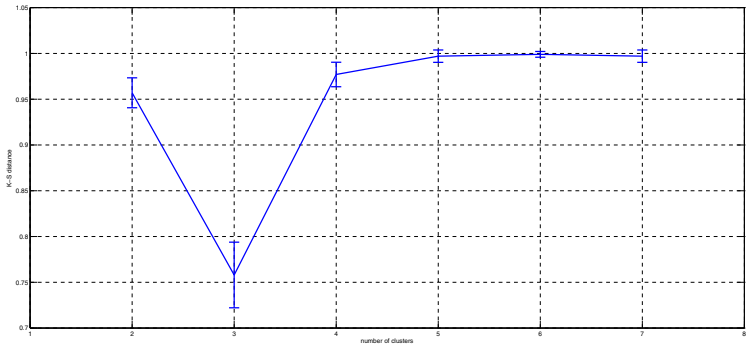

Fig. 7.18. Error-bar plot of the KS distance for the three-text collection dataset for 300 terms

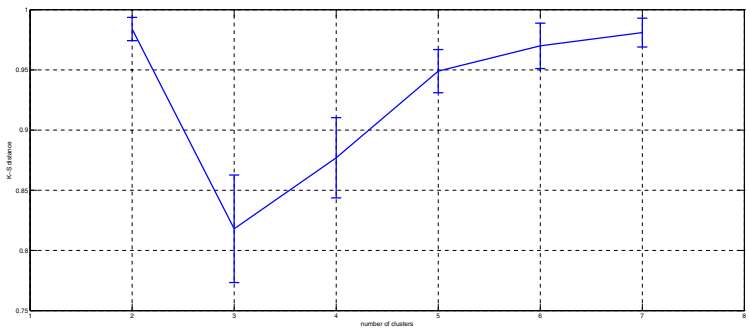

Fig. 7.19. Error-bar plot of the KS distance for the Iris dataset

In the spirit of the general concept stated previously (see Section 7.4), namely that if one takes two independent disjoint samples without replacement from the analyzed population, the elements of the samples should be mingled in each of the clusters as well as they are mingled in the entire population. Moreover, if the samples are equally sized, it is expected that every population subset should consist of similar numbers of elements of both samples. Carrying this idea even further leads us to expect that the $K-NN$ quantity of each point should also be evenly divided between the two samples.

Let us take an item $\mathbf{w}$ from one of the samples and consider, for a given K, the number $m_K(\mathbf{w})$ of its $K-NN$ belonging to an element's sample. This quantity has a hypergeometric distribution:

$$\mathrm{P}(m_K(\mathbf{w}) = n) = \frac{\binom{T-1}{n}\binom{T}{K-T}}{\binom{2T-1}{K}},\ n \in 0..K,$$

where $T > K+1$ is the sample size. In accordance with the well-known 10% Rule, we can conclude that if $T-1 > 10K$, the binomial probability $Bin(K, 0.5)$ with the parameters K and 0.5 does not differ much from the actual (hypergeometric) probability. Explicitly, under the assumption of a stable partition, the number of elements inside a cluster belonging to the element's own sample is suggested to have *the Binomial Distribution* $Bin(K, 0.5)$. This is the null hypothesis characterizing well-constructed clusters. Moreover, as will be demonstrated in the numerical experiments section, *the Binomial Model* often provides a good representation of the actual distribution, even in situations where the 10% Rule does not formally hold.

Dissimilarities between the observed and the expected $K-NN$ proportions belonging to the points own samples can be characterized via the p-values, which are calculated for an alternative hypothesis under the null hypothesis distribution $Bin(K, 0.5)$. One-tailed or two-tailed p-values can be employed. Generally, the use of the second one appears to be more appropriate, although the first provides more flexible outcomes.

To describe cluster quality, the obtained p-values have to be aggregated. In the current approach, cluster merit is assigned by a summarizing index relying on a version of the multiple testing statistics. The index is chosen to represent the fraction of the cluster elements providing p-values that exceed a given threshold, i.e., the cluster part where the null hypothesis is rejected at the significant level that is equal to the threshold value. Partition quality is defined by the worst partition cluster, where the fraction of the points with null hypothesis rejection is maximal. The procedure offers to produce empirical distributions of a p-values such that the true number of clusters is attained by the empirical distribution having a minimal right tail.

7.9.1 Methodology

As was emphasized earlier in Section 7.7 we consider a finite subset $\mathbb{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T\}$ of the d-dimensional Euclidean space $\mathbb{R}^d$, drawn according to an underlying probability density function $\mathrm{P}(\mathbb{W})$ defined in (6.9) and assume that the source partition corresponding to the true number of clusters satisfies the condition (7.30).

The mixing of any pair of disjoint samples $\mathbb{S}_1$ and $\mathbb{S}_2$ randomly drawn without replacement, is modeled by means of the Binomial distribution of the $K - NN$ classification approach.

For this purpose, we introduce $\mathbb{S} = \mathbb{S}_1 \cup \mathbb{S}_2$ and obtain

$$\mathcal{X}^k(\mathbb{S}) = Cl_k(\mathbb{S}). \tag{7.37}$$

Let us take an item $\mathbf{w} \in \mathbb{S}$. If

$$|\mathbb{S}_1| = |\mathbb{S}_2| \tag{7.38}$$

then, as mentioned earlier, for a given K, the amount of $K-NN$ of $\mathbf{w}$ belonging to its own sample is distributed according to the Binomial distribution $Bin(K, 0.5)$ with K trials and the success probability 0.5. Therefore, the null hypothesis is

$$H_0 : \rho_K(\mathbf{w}) = 0.5,$$

where $\rho_K(\mathbf{w})$ is the fraction of $K - NN$ belonging to its own sample of an element $\mathbf{w} \in \mathbb{X}_j$, that exhibits a goodness of cluster $\mathbb{X}_j$ at the point $\mathbf{w}$. An alternative p-value calculated under a general alternative seems to reflect dissimilarity between what is observed and what is expected under H_0 neighbors' portions.

Two alternative p-values can be calculated as

- $H_1 : \rho_K(\mathbf{w}) > 0.5, P_1(\mathbf{w}) = 1 - binocdf(KNN(\mathbf{w}) - 1, K, 0.5)$,
- $H_1 : \rho_K(\mathbf{w}) \neq 0.5, P_2(\mathbf{w}) = 2\min(P_1(\mathbf{w}), binocdf(KNN(\mathbf{w}), K, 0.5))$,

where $KNN(\mathbf{w})$ is the $K - NN$ amount of $\mathbf{w}$ belonging to its own sample, *binocdf* denotes *the Binomial Cumulative Distribution Function* at z:

$$binocdf(z, K, 0.5) = \left(\frac{1}{2}\right)^K \sum_{i=0}^{z} \binom{K}{i}.$$

The use of P_2 generally appears to be more appropriate, though P_1 provides more flexible outcomes. Indeed, a variant of one sample extract binomial test is considered here.

To summarize the results obtained at each cluster point, we can apply a version of the multiple testing methodologies statistics:

$$Ind_{1,l}(\mathbb{X}_i) = \frac{|th < P_l(\mathbf{w})|\mathbf{w} \in \mathbb{X}_i|}{|\mathbb{X}_i|},\ l = 1, 2;\quad i \in 1..k, \tag{7.39}$$

where "th" is an additional parameter representing a threshold value, and $Ind_{1,l}(\mathbb{X}_i)$ represents the fractions of $\mathbb{S}$ where the null hypothesis is rejected at the significant level th. Hence, the index values close to zero can indicate a well constructed cluster.

Another index is the p-values mean:

$$Ind_{2,l}(\mathbb{X}_i) = mean(P_l(\mathbf{w})|\mathbf{w} \in \mathbb{X}_i),\ l = 1, 2;\ i \in 1..k. \tag{7.40}$$

The entire partition can be characterized by its worst element, i.e., by the value

$$IND_k = \max_{j,l\in\{1,2\},\ i\in 1..k}(Ind_{j,l}(\mathbb{X}_i)). \tag{7.41}$$

This value is provided by the worst cluster of the partition, where the null hypothesis has been rejected at significance level "th" for the largest fraction points among all partition clusters in the case $Ind_{1,l}(\mathbb{X}_i)$. Once $Ind_{2,l}(\mathbb{X}_i)$ has been chosen, IND_k is defined by the worst cluster having the maximal mean p -value. Consequently, the null hypothesis appears to be less realistic here. Note, that the cluster sizes are ignored in this formulation. Therefore, in line with the concept stated in [67], it is indirectly suggested that (7.38) holds.

In order to employ this technique to the partition $\mathcal{X}^k(\mathbb{S})$, we have to employ one of the simulation methods described in Section 7.5.

Outliers of the data together with the algorithm's drawbacks can make a major contribution to model noise. Hence, the suitable number of clusters should be determined based upon a sufficiently large amount of data. In the current approach this determination is achieved by constructing an appropriate empirical index distribution. The natural expectation is that the distribution having the shortest right tail indicates the true number of clusters. Distribution asymmetry can be quantified by means of a distribution attribute generally referred to here as "an asymmetry indicator" AI.

Algorithm:

Input parameters of an algorithm implementing our procedure are:

$k_{\max}$ — maximal number of clusters to be tested;
J — number of pairs of the drawn samples;
m — samples size;
K — number of the Nearest Neighbors;
th — the threshold value for calculation of Ind_1;
Cl — the clustering algorithm.

Output: the "true" number of clusters $k^\star$.

1. For $k = 2$ to $k_{\max}$
2. For $j = 1$ to J
3. $\mathbb{S}_{1,j} = sample(\mathbb{S}, m),\ \mathbb{S}_{2,j} = sample(\mathbb{S}\backslash\mathbb{S}_{1,j}, m)$;
(a pair of disjoint samples $\mathbb{S}_1$ and $\mathbb{S}_2$, of size m from the set $\mathbb{S}$, is drawn without replacement)

4. Apply a simulation method and obtain the occurrences $\mathbb{S}^i_{1,j}, \mathbb{S}^i_{2,j}$, $i \in 1..k$ of the samples in the clusters;
5. Calculate index values $Ind_{1,l}(\mathbb{X}_i(\mathbb{S}_j))$ for all clusters $i \in 1..k$ according to (7.39) or (7.40), ($l = 1$ or 2);
   (the obtained values characterize the qualities of the clusters)
6. Based on (7.41), calculate the partition index value representing the partition quality

$$I_k(j) = IND_k(\mathbb{S}_j);$$

7. end For j;
   (after this loop an array I_k of the partition index, having size J, is constructed for the current number of clusters k)
8. Calculate an asymmetry indicator $AI(k)$ of the array $\{I_k(j)\}_{j \in 1..J}$;
9. end For k;
   (after this loop an array $\{AI(k)\}_{k \in 2..k_{\max}}$ of an asymmetry indicator is constructed)
10. The "true" number of clusters $k^\star$ is the number that maintains the shortest right tail of I_k. The tail is represented by an extreme value of the array AI.

There are many possibilities for choosing an asymmetry indicator AI that will yield the empirical index distribution having the shortest right tail. The most common parameter that expresses the lack of distribution asymmetry around the sample mean is the skewness statistic. The experiments showed that the sample percentiles demonstrate more stable behavior.

7.9.2 Numerical Experiments

For the purpose of estimating the ability of the described technique, various numerical experiments were carried out on synthetic and real datasets. For the simulation step we apply the second simulation method described in Subsection 7.5.2.

An additional issue emerging in resampling approach applications is the selection of parameters. This issue is usually omitted or else covered very briefly. A theoretical solution of this problem is almost not available (e.g., [99, 208] and [213]). A common recommended tactic is to split the data in order to obtain the maximal achievable consistency. This practice appears to be quite natural for small datasets, but for medium or large datasets it can lead to oversized samples. Another common practice is to set the sample size to be equal to a tenth of the data set so that its minimal size is about $5K$. As for the choice of $k_{\max}$, two opposite tendencies should be taken into account. (Larger values of k diminish the influence of noise on the classification (e.g., [106]) although make margins among classes to be less clear). Experiments reveal that a balanced value $k_{\max} = 10$ often leads to good results. The value of J can be set by an iterative procedure. An initial

relatively low (about 30) number is selected, and then consecutively increased. The process is stopped once the obtained outcomes begin varying slowly. Due to the assumed procedure consistency, such a condition may indicate that the method's limitation has been reached.

We exhibit results obtained for Ind_1 with $th = 0.01$ and $k_{\max} = 7$, presented via the error-bar plots for the 75-th, 90-th and 95-th percentiles calculated within the 10 trials and marked by pentagrams, circles and diamonds, respectively. The bar sizes are equal to two standard deviations, found inside the trials. As the number of clusters is known in all the given cases, we can conclude that all true numbers of clusters are stably detected by the appropriate minimal values of the percentiles with minor variation within the trials. Moreover, there are no significant differences in the quality of the three percentile results. Examination of the contents of the clusters reveals that in the case of the true number of clusters, the majority of the items are placed in their actual subgroups.

For clustering, the *Partitioning Around Medoids, PAM* [187] algorithm has been used because it appears to be more robust than the k-means approach (see, for example, [99]).

Simulated Data. The first dataset of 100 elements has been simulated as a mixture of three two-dimensional Gaussian distributions with independent

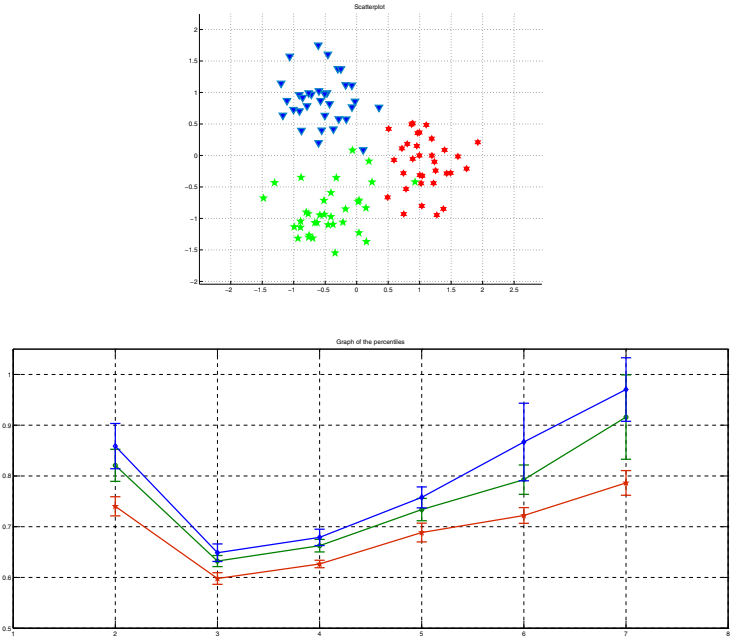

Fig. 7.20. Scatter plot of the three-component Gaussian dataset and the percentiles error-plot graph

coordinates owning the same standard deviation $\sigma = 0.4$. Mean values of the components are placed on the unit circle on the angular neighboring distance $2\pi/3$.

The scatter-plot of this data in Figure 7.20 shows that the inherent clusters overlap. Nevertheless, our method, using the parameters $J = 50$, $m = 50$ and $K = 10$, succeeds in finding the true number of clusters (see Figure 7.20).

Another dataset of 200 items was created according to the same principle based upon four Gaussian components with the centers located on the unit circle on an angular neighboring distance $2\pi/4$ and a standard deviation of 0.3. The results obtained are shown in Figure 7.21. Although the dataset components are also not separated, the true number of clusters has been detected under the parameters $J = 100$, $m = 50$ and $K = 10$.

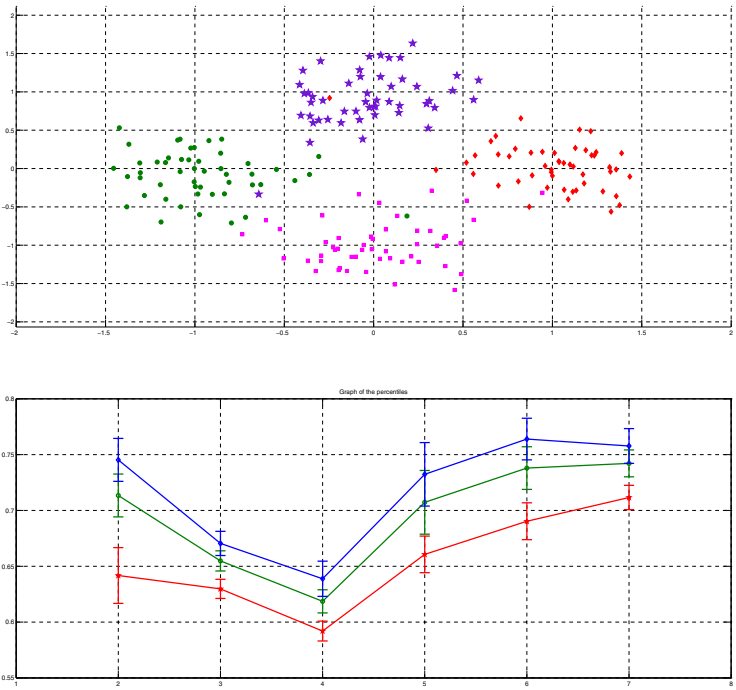

Fig. 7.21. Scatter plot of the four-component Gaussian dataset and the percentiles error-plot graph

Real World Data. *Flea Dataset.* The first real world dataset chosen for numerical experiments testing this methodology was the Flea beetles dataset described in Appendix A.2.6. As can be seen in Figure 7.22, the true number of clusters is detected.

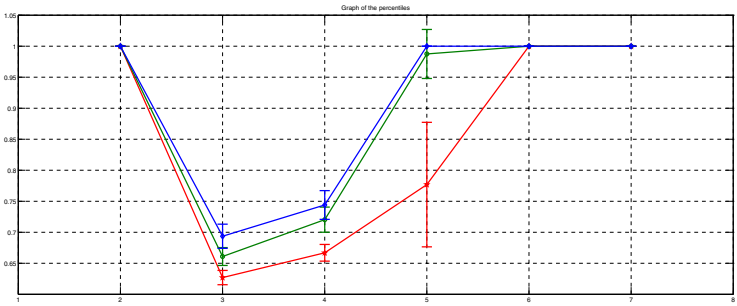

Fig. 7.22. Error-plot graph of the percentiles for the Flea Beetles dataset. $J = 50$, $m = 30$ and $k_{\max} = 10$.

The Iris Flower Dataset. The second real world dataset was the Iris flower dataset described in Appendix A.2.3. The true cluster structure was detected using the parameters $J = 50$, $m = 50$ and $K = 10$ (see Figure 7.23).

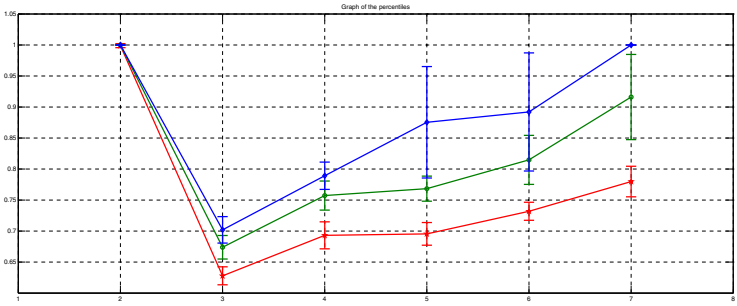

Fig. 7.23. Error-plot graph of the percentiles for the Iris flower dataset. $J = 50$, $m = 30$ and $k_{\max} = 10$.

The Three-Text Collection. The third real world dataset considered is the three-text collection described in Appendix A.2.1. In this case, clustering was carried out for the first two leading components, and the true number of clusters was easily detected. Here, $J = 50$, $m = 100$ and $K = 10$. The outcomes are presented in Figure 7.24.

Even in the case of a "blurred" variant of this dataset based on only 300 terms, our method demonstrates stable performance and provides the correct outcomes (see Figure 7.25).

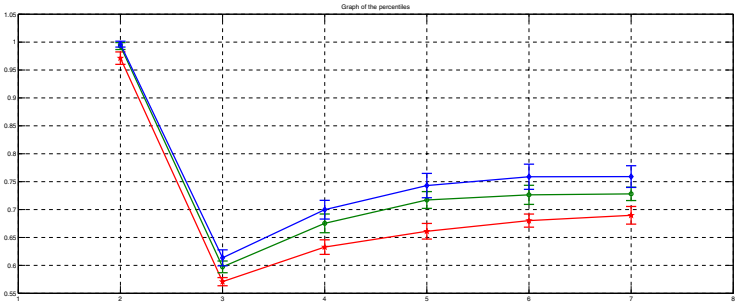

Fig. 7.24. Error-plot graph of the percentiles for the three-text collection data constructed on 600 terms. $J = 50$, $m = 100$ and $k_{\max} = 10$.

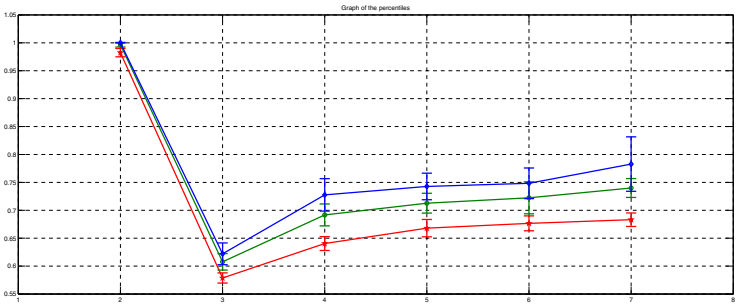

Fig. 7.25. Error-plot graph of the percentiles for the three-text collection data constructed on 300 terms. $J = 50$, $m = 100$ and $k_{\max} = 10$.

7.9.3 Comparison with Other Methods

In order to evaluate the capacity of our method, we compare our results to outcomes obtained for the same datasets by means of several other cluster validation methods. Specifically, we compare our results to those produced by the Calinski and Harabasz index [60] (CH-index), the Hartigan index (H-index) [159], the Krzanowski and Lai index (KL-index) [201] and the Sugar and James index (SJ-index) [309], described in Section 7.2. The analyzed datasets are denoted as follows: $G - 3 - 04$, $G - 4 - 03$, Flea, Iris, D_3_600 and D_3_300, according to their order of appearance in this subsection. Our method, designated here as $K - NN$ method, succeeds quite well in the comparison because it is the only one that yields the true number of clusters for all datasets.

Table 7.2. Number of clusters obtained by various methods

	$G-3-04$	$G-4-03$	$Flea$	$Iris$	D_3_600	D_3_300
CH-index	3	4	4,7	3	-	6
H-index	3	4	2	2	3	3
KL-index	3	5	2	2	3	-
SJ-index	6	5	3	3	3	-
$K-NN$ method	3	4	3	3	3	3
True	3	4	3	3	3	3

7.10 Hoteling Metrics

Within the framework of probabilistic metric methodology described earlier and based upon the Gaussian Mixture Model (GMM) presented in Section 6.4, we consider here an approach to the cluster validation problem. We assume the case of components having common covariance matrices, i.e., components having the same shape. According to our general perception stated in Section 7.4, we claim that sequences of clustered samples can be interpreted as normally distributed i.i.d. samples, if the number of clusters is chosen correctly. Here, sample closeness within the clusters is measured by means of the values calculated for the appropriate *Hotelling's* T*-square statistic*. Data outliers and clustering algorithm shortcomings can cause small p-values. However, we assume that their empirical distribution is least concentrated at the origin if the number of clusters is chosen correctly.

7.10.1 Approach Description

For two i.i.d. disjoint samples $\mathbb{S}_1$ and $\mathbb{S}_2$, independently drawn from $\mathbb{W} \subset \mathbb{R}^d$ and a partition $\mathcal{X}^k(\mathbb{W})$, we consider the sample occurrences in clusters:

$$\mathbb{S}_l^i = \mathbb{S}_l \cap \mathbb{X}_i,\ l = 1, 2,\ i \in 1..k.$$

The distance between the sets $\mathbb{S}_1^i$ and $\mathbb{S}_2^i$ can be measured by the Hotelling's two-sample T-square statistic:

$$\widehat{\mathrm{t}}^2\left(\mathbb{S}_1^i, \mathbb{S}_2^i\right) = \frac{\left|\mathbb{S}_1^i\right| \cdot \left|\mathbb{S}_2^i\right|}{\left|\mathbb{S}_1^i\right| + \left|\mathbb{S}_2^i\right|} \cdot \left(\overline{\mathbb{S}}_1^i - \overline{\mathbb{S}}_2^i\right)' \mathbf{R}^{-1} \left(\overline{\mathbb{S}}_1^i - \overline{\mathbb{S}}_2^i\right)$$

where $\overline{\mathbb{S}}_1^i, \overline{\mathbb{S}}_2^i$ are the set means and $\mathbf{R}$ is the mutual variance-covariance matrix. Under our assumption, the sets $\mathbb{S}_1^i$ and $\mathbb{S}_2^i$ for each $i \in 1..k$ could have been, consistently, drawn from the Gaussian distribution $G(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\Gamma})$ with the mean $\boldsymbol{\mu}$ and the mutual covariance matrix $\boldsymbol{\Gamma}$. In this case the variable $\widehat{\mathrm{t}}^2\left(\mathbb{S}_1^i, \mathbb{S}_2^i\right)$ has the Hotelling's T-square distribution $\mathrm{T}^2\left(d, \left|\mathbb{S}_1^i\right| + \left|\mathbb{S}_2^i\right| - 2\right)$ with parameters d and $\left|\mathbb{S}_1^i\right| + \left|\mathbb{S}_2^i\right| - 2 > 0$ (see, for example, [26]). In this context and under the null hypothesis, a cluster merit can be represented as the p-value calculated for the observed $\widehat{\mathrm{t}}^2\left(\mathbb{S}_1^i, \mathbb{S}_2^i\right)$ value:

$$\delta_i = 1 - \mathrm{T}^2 CDF\left(d, \left|\mathbb{S}_1^i\right| + \left|\mathbb{S}_2^i\right| - 2\right) \cdot \widehat{\mathrm{t}}^2\left(\mathbb{S}_1^i, \mathbb{S}_2^i\right) \tag{7.42}$$

where $\mathrm{T}^2 CDF$ denotes *the Cumulative Distribution Function* of the corresponding *Hotelling's* T-*square distribution*. Consequently, the partition quality is given by:

$$\delta\left(\mathcal{X}^k\right) = \min_{i \in 1..k} \delta_i. \tag{7.43}$$

That is, the partition is characterized by the cluster that mostly differs from a Gaussian cluster.

A situation in which only in the case of a true number of clusters, the value $\delta\left(\mathcal{X}^k\right)$ defined in (7.43) exceeds a given threshold, say 0.05, is desirable, i.e., a situation in which the null hypothesis is not rejected in all clusters but merely in the case of the stable partition. Data inconsistency with the underlying model together with the shortcoming of the algorithm seriously increases the procedure's ambiguity. For this reason, the number of clusters has to be inferred, as in the previous discussed approaches relying large amounts of data. This inference can be accomplished, for instance, by considering an empirical distribution of $\delta\left(\mathcal{X}^k\right)$ for several different numbers of clusters. The natural anticipation is that the distribution with the shortest left tail belongs to the true number of clusters.

A meta-algorithm implementing the proposed method can be expressed as follows:

Algorithm:

1. Repeat for each tested number of clusters from k_{min} to k_{max}:
   a) Repeat the following steps for pairs of samples randomly drawn without replacement;
      i. Simulate the sample occurrences in the clusters;
      ii. According to (7.42), calculate the p-values of the distances between the occurrences of different samples within the clusters;
   b) According to (7.43), calculate the partition merit;
2. The estimate of the true number of clusters is the $k^\star$, for which the distribution is the least concentrated at the origin.

7.10.2 Experimental Results

This subsection presents the experimental results obtained from implementation of the described methodology. The follow notations are used:

- k_{min} — the minimal tested number of clusters (the default value is 2);
- k_{max} — the maximal tested number of clusters (the default value is 8);
- J — the number of drawn sample pairs;
- m — the size of the drawn samples;
- Cl_k — the used clustering algorithm (the default is the standard k-means algorithm).

The distribution concentrations at the origin are characterized by two attributes:

- P_{50} — the sample median;
- N_g — the frequency of the lowest subgroup obtained by dividing the values range into g equal range subgroups (the default value of g is 30).

The second simulation method presented in Subsection 7.5.2 was applied. The results obtained for several synthetic and real datasets are shown by comparison with the "known" source number of clusters. In order to exhibit the stability of the approach, the procedure is repeated 10 times for each datum, and the results are represented via error-bar plots of P_{50} and N_g, calculated as functions of the tested number of clusters. The bar sizes are equal to two standard deviations found within the trials.

Synthetic Data. The first three simulated datasets, each with a size of 4000, are drawn from mixtures of two-dimensional Gaussian distributions with 4, 5 and 6 components respectively and having the same standard deviation $\sigma = 0.35$. The components means are located on the unit circle at equal angular neighboring distances from each other. The dataset components overlap, as seen in Figures 7.26–7.28.

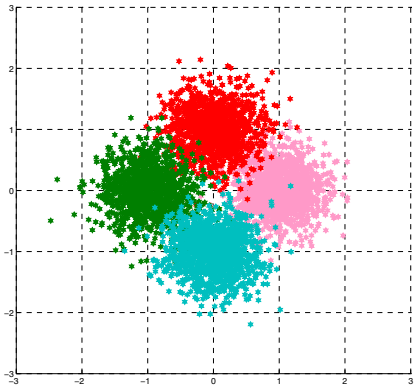

Fig. 7.26. Scatter plots of the first simulated dataset

The results obtained for the two first datasets for the parameters $J = 100$ and $m = 300$ are shown in Figure 7.29 and Figure 7.30. The "true" number of clusters has been correctly found for these two datasets. However, in the case of six-component dataset having a very blurred cluster structure, the true number of clusters is not detected (see Figure 7.31).

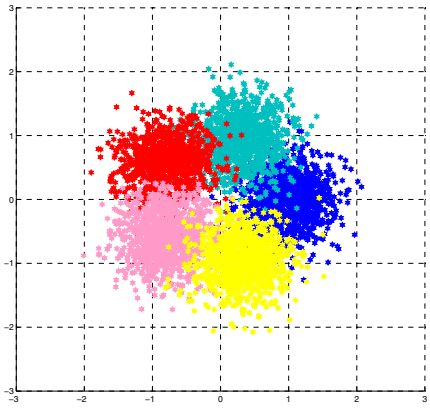

Fig. 7.27. Scatter plots of the second simulated dataset

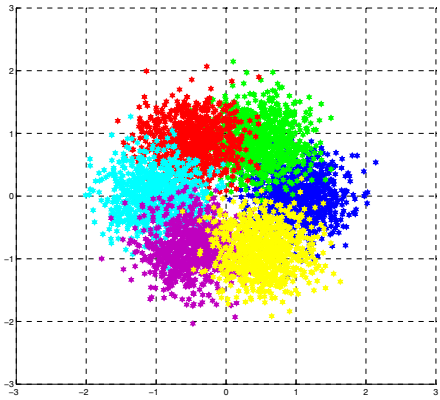

Fig. 7.28. Scatter plots of the third simulated datasets

Even for a better separated six-component set, where $\sigma = 0.3$ (see Figure 7.32), the "true" number of clusters is not clearly determined (see Figure 7.33).

Nevertheless, increasing the sample size to $m = 400$ for $\sigma = 0.3$ leads to determining the correct number of clusters (see Figure 7.34). This fact testifies to the consistency of the proposed methodology.

Three-Texts Collection. The real dataset chosen for examination here the three-text collection $T3$ described in Appendix A.2.1

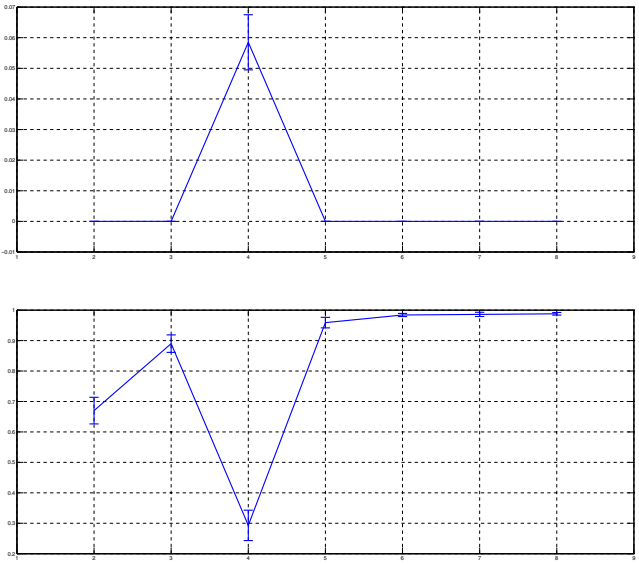

Fig. 7.29. Error-bar plots of P_{50} and N_{30} for the simulated four-component Gaussian dataset

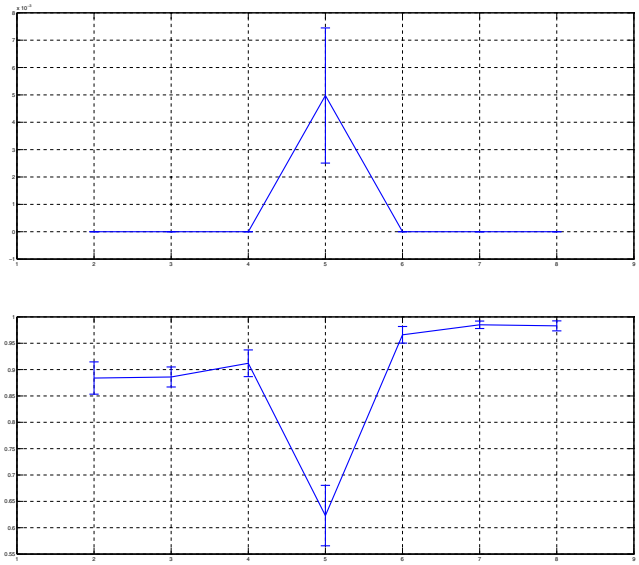

Fig. 7.30. Error-bar plots of P_{50} and N_{30} for the simulated five-component Gaussian dataset

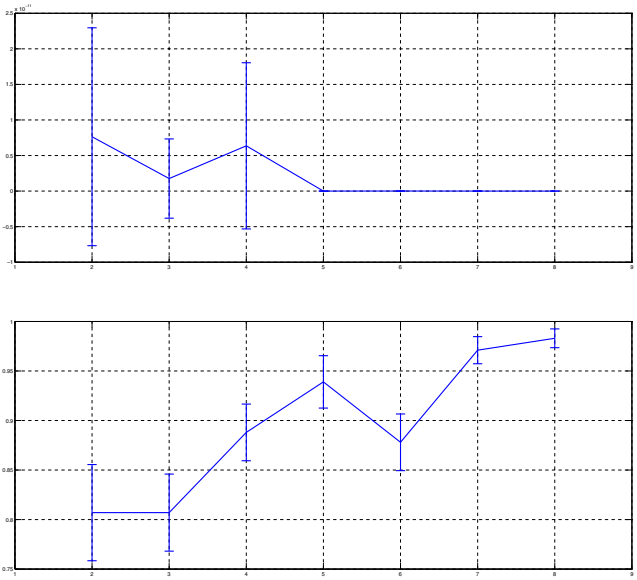

Fig. 7.31. Error-bar plots of P_{50} and N_{30} for the simulated six-component Gaussian dataset

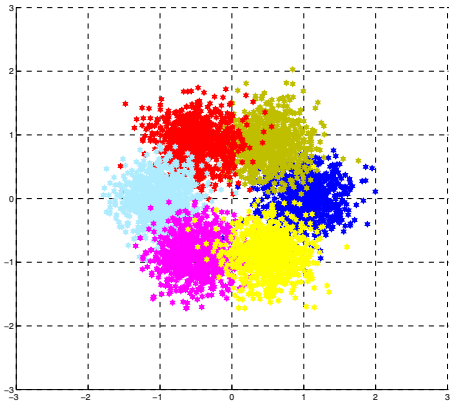

Fig. 7.32. Scatter plots of the six-component dataset with $\sigma = 0.3$

The results in Figure 7.35 show that the number of clusters is properly determined.

Analogous results are observed in a case where only 300 "best" terms are used. This dataset can be considered as a "noisy" version of the previous one.

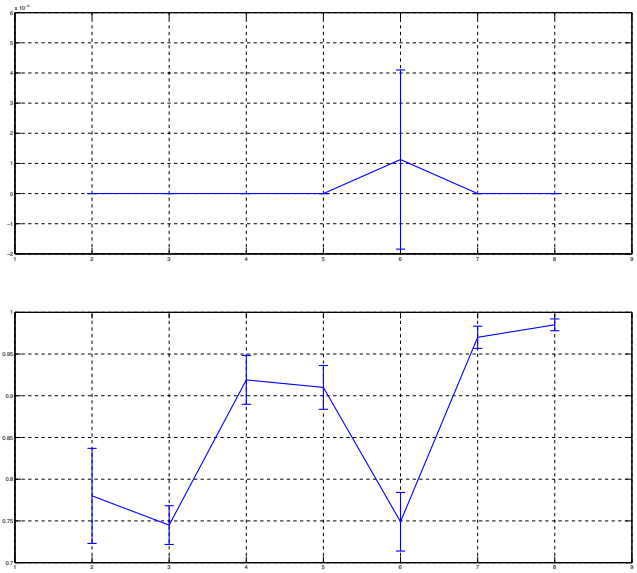

Fig. 7.33. Error-bar plots of P_{50} and N_{30} for the simulated six-component Gaussian dataset with $\sigma = 0.3$

7.11 Cluster Validation as an Optimization Problem

Based on the inner index approach proposed by Sugar and James [309] in the framework of rate distortion theory (see Section 7.2), the task of determining the true number of clusters can formally be represented as follows. Let $\mathbf{w}$ be a d-dimensional random variable having a mixture distribution of $k^\star$ components, each with covariance $\mathbf{\Gamma} > 0$. Then the minimum achievable distortion associated with fitting k centers to the data is

$$G_k = \frac{1}{d} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_k} E\left[\left(\mathbf{w} - \mathbf{x}_{\gamma(\mathbf{w})}\right)^{\mathrm{T}} \mathbf{\Gamma}^{-1} \left(\mathbf{w} - \mathbf{x}_{\gamma(\mathbf{w})}\right), \right],$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_k$ is a set of k cluster centers obtained by running a standard clustering procedure; $\mathbf{x}_{\gamma(\mathbf{w})}$ is the nearest cluster center to $\mathbf{w}$. Note that in a case where $\mathbf{\Gamma}$ is the identity matrix distortion is simply the mean squared error. Given the distortions G_k, a "jumping differential" curve is constructed according to the following rule:

$$J_k = G_k^{-\lambda} - G_{k-1}^{-\lambda},$$

where λ is the transformation power. According to asymptotic results obtained from distortion rate theory [309], its preferred value is

$$\lambda = d/2.$$

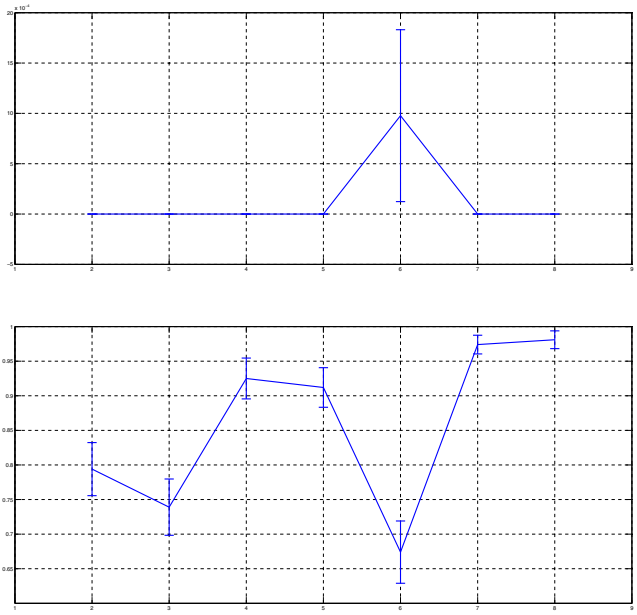

Fig. 7.34. Error-bar plots of P_{50} and N_{30} for the simulated six-component Gaussian dataset with $\sigma = 0.3$ for $m = 400$

Moreover, for rather high values of d, the differential distortion J_k asymptotically tends toward zero subject to the condition that the number of clusters k is less than the number of components $k^\star$. Thus, for large d, transformed distortion J_k is almost zero when $k < k^\star$. The value then jumps jumps abruptly and increases linearly when $k \geq k^\star$. The Sugar and James' "jump" algorithm exploits such behavior in order to determine the most likely value of k as the true number of clusters. The estimated number of clusters corresponds to the index of maximal value of transformed distortion function J_k:

$$k^\star = \arg\max_k J_k.$$

Thus, the problem of finding the true number of clusters can be interpreted as a particular case of more general problems, namely, the fault detection problem or the problem of locating the discontinuity points of an implicitly defined function. Generally, the problem can be formulated as follows. Let us take a real-valued function f on the interval $[0, 1]$ having no more than one jump discontinuity $x^\star \in (0, 1)$. The problem consists of finding a confidence interval for $x^\star$ subject to:

As7.1: The function $f(\cdot)$ is Lipschitz continuous with a Lipschitz constant L on the intervals $(0, x^\star)$ and $(x^\star, 1)$.

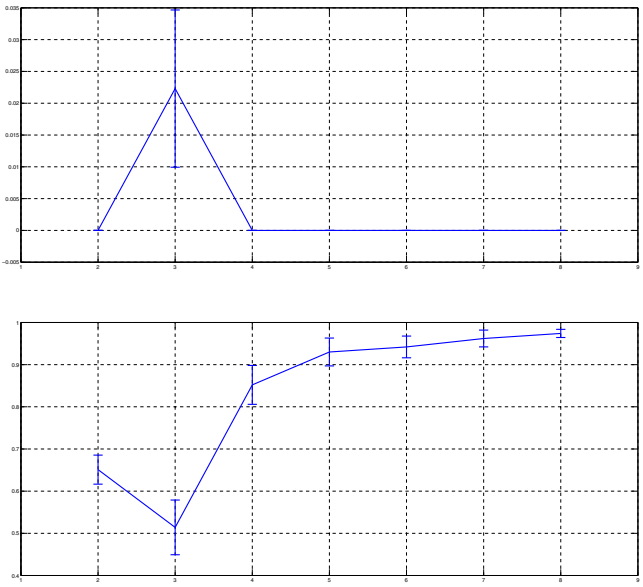

Fig. 7.35. Error-bar plots of P_{50} and N_{30} for the three-text collection dataset constructed on 600 terms with $J = 100$ and $m = 100$

As7.2: If jump discontinuity exists at the point $x^\star$, then the jump size at this point is above a certain constant value $B > 0$.

Let us introduce a continuous piecewise linear function f as follows:

$$f_G\left(\frac{k}{k_{\max}}\right) = G_k^{-\lambda},\ f_G(x) = G_k^{-\lambda} + \left(x - \frac{k}{k_{\max}}\right) J_{k+1},$$

for $\frac{k}{k_{\max}} \le x \le \frac{k+1}{k_{\max}}$, $k \in 0..k^\star - 2 \cup k^\star..k_{\max} - 1$, and

$$f_G(x) = G_{k^\star-1}^{-\lambda},$$

for $\frac{k^\star-1}{k_{\max}} \le x \le \frac{k^*}{k_{\max}}$. Denote $r(x) = \left(\frac{k}{k_{\max}}, \frac{k+1}{k_{\max}}\right)$ if $\frac{k}{k_{\max}} \le x \le \frac{k+1}{k_{\max}}$.

7.11.1 Fast Randomized Algorithm for Finding True Number of Clusters

An algorithm that implements the approach in question can be described as follows:

1. Choose the reliability parameter $\beta \in (0, 1)$.
2. Choose the parameters N, S so that:

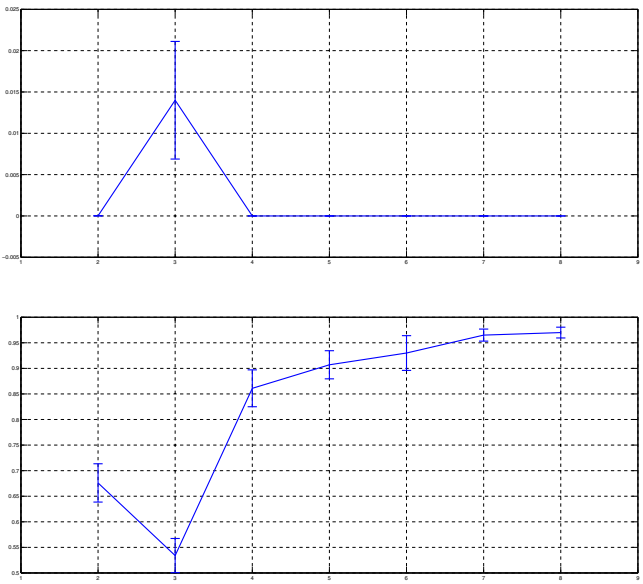

Fig. 7.36. Error-bar plots of P_{50} and N_{30} for the three-text collection dataset constructed on 300 terms with $J = 100$ and $m = 100$

$$S = \left[\frac{4\sqrt{2}Lk_{\max}}{\sqrt{1-\beta}BN} - \frac{1}{N} \right] + 1. \tag{7.44}$$

3. Randomly choose S groups of N points from interval $(0, 1)$:

$$Z_s = \{z_{s,n}\}_{n \in 1..N},\ s \in 1..S.$$

and denote

$$Z = \bigcup_s Z_s.$$

The proof of Theorem 1 in [240] shows that the largest distance between two sequential points belonging to Z does not exceed $\frac{B}{4L}$ with probability of $1 - \beta$.

4. Choose $D, M > 0$. For each one of the groups Z_s, $s \in 1..S$ construct a uniform approximation for $f_G(x)$:

$$g_s(x) = \sum_{m=0}^{M} a_{s,m} p_m(x),\ s \in 1..S, \tag{7.45}$$

minimizing the error

$$\delta_s = \max_{x \in Z_s} |g_s(x) - f_G(x)|$$

subject to

$$|a_{s,m}| \le D,\ m \in 0..M,\ s \in 1..S.$$

Here a convex optimization can be applied (MATLAB TOOLBOX: YALMIP, SeDuMi or cvx).

If one of the approximation problems is not resolved, return to Step 1 and start over.

5. The set

$$\hat{X} = \{\tilde{x} = xk_{\max} : x \in (0,1),\ \chi(x) > l(x)\} \tag{7.46}$$

where

$$\chi(x) = \max_{s \in 1..S} g_s(x) - \min_{s \in 1..S} g_s(x),\ x \in (0,1) \tag{7.47}$$

$$l(x) = B - \frac{B}{4Lk_{\max}} h(x) - 2\delta,$$

$$h(x) = \max_{z \in r(x)} \max_{s \in 1..S} |g_s'(z)|, \tag{7.48}$$

is a confidence set for $x^\star$ and following theorem holds:

Theorem 7.10. *(Theorem 1 [240]) If conditions As7.1,7.2 formulated above are satisfied* **then** *with probability of at least $p = \beta$ the set $\hat{X}$ defined in* (7.46) *is not empty and contains the point $x^\star k_{\max}$ which equals to the true number of clusters.*

7.11.2 Numerical Experiments

To check whether the proposed algorithm can be applied to a large number of clusters, a synthetic dataset was generated. It contained 3100 clusters, each composed of 8–19 instances. The instances in each cluster were generated according to a Gaussian distribution on the unit square $[0,1] \times [0,1]$ with a random center for each cluster.

We consider the interval $[1, 8000]$ that contains the real number of clusters. For each point the transformed distortion function G_k is calculated using the Sugar and James algorithm. Note that for each $k \in 1..8000$ we apply the clustering algorithm (k-means) and after that compute G_k. The results are shown in Figure 7.37.

The scenario approach described above in Subsection 2.2.2 allows us to significantly reduce the number of clustering algorithm computations. Assuming that $B > 1.0$ and $L < 0.002$, we choose $\beta = 0.9$; $M = 20$; $N = 29$; $D = 0.6$. Hence, subject to (7.44) $S = 10$. Thus, having computed only 290 values of G_k, we reduced the interval of uncertainty 14 times, instead of computing 7400 values as implied by the traditional deterministic approach, in order to obtain the confidence interval $\hat{X}$ with probability of at least $\beta = 0.9$. Three approximation curves $g_t(\cdot)$ are shown in Figure 7.38, along with the resulting function $\chi(\cdot)$.

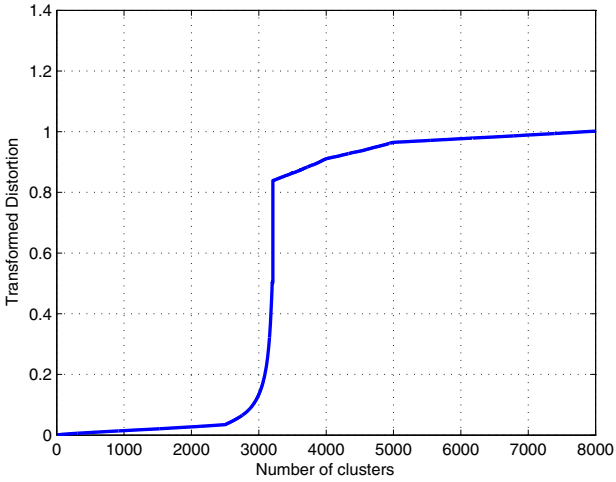

Fig. 7.37. Index function for initial data set

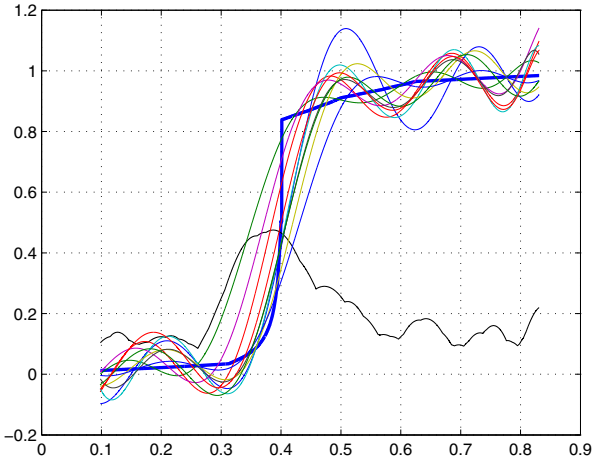

Fig. 7.38. Approximation curves

On the assumption that $B > 1.0$ and $L < 0.002$, we obtain the decision-making level, shown in Figure 7.39 along with the resulting function $\chi(\cdot)$. A peak near the point $x^\star = 3200$ can be observed. If we consider segment $[2750, 3410]$ to be the confidence interval $\hat{X}$, then 660 computations of index function G_k are required to obtain an eventual solution. Thus the total number of index function computations is 950, which is considerably less than the initial number of 8000.

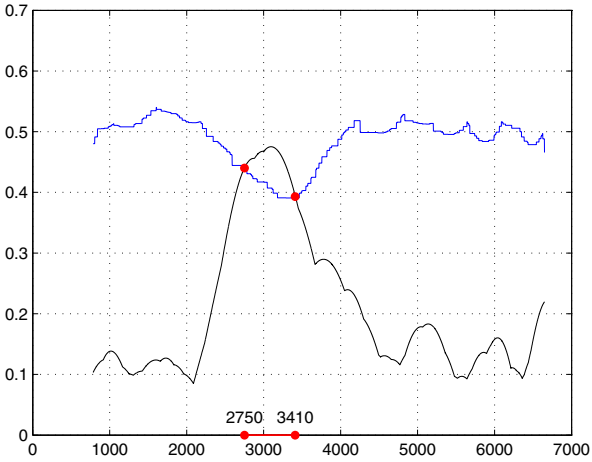

Fig. 7.39. Resulting function and confidence level

A

Appendix— Experimental Data Sets

The approaches described in Chapters 6 and 7 are exemplified by means of various numerical experiments on synthetic and real datasets, as described in this Appendix.

A.1 Synthetic Data

A.1.1 G5: Five-Cluster Simulated Gaussian Dataset

The first synthetic data set consists of a mixture of five two-dimensional Gaussian distributions that have independent coordinates with the same standard deviation $\sigma = 0.25$. The components means are placed on the unit circle with an angular neighboring distance of $2\pi/5$. The dataset (denoted as $G5$) contains 4000 items. The scatter plot of this data is shown in the next figure

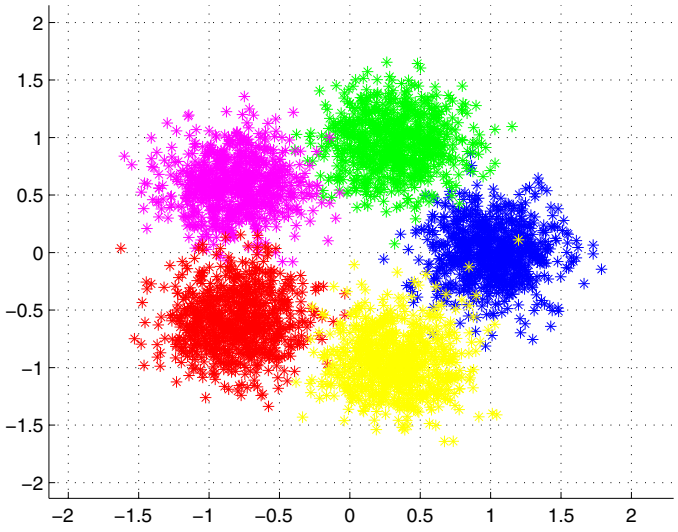

Fig. A.1. Scatter plot of the Gaussian dataset

A.2 Real-World Data

A.2.1 Three-Text Collections

The first real dataset was chosen from the text collection
http : *//ftp.cs.cornell.edu/pub/smart/*.
This set (denoted as $T3$) includes the following three text collections:

- DC0–Medlars Collection (1033 medical abstracts);
- DC1–CISI Collection (1460 information science abstracts);
- DC2–Cranfield Collection (1400 aerodynamics abstracts).

This dataset has been considered in many studies ([95, 197, 198, 199] and [340]). Typically, 600 "best" terms were selected according to the well-known "bag of words" approach (see, [94] for term selection details). Accordingly the dataset was mapped into Euclidean spaces with dimensions of 600 with dimension reduction provided by *Principal Component Analysis (PCA)*. The considered dataset is recognized to be well-separated by means of the two leading principal components. We used this data representation in our experiments.

A.2.2 Three-Text Collection–Second Version

A "blurred" version of the previous dataset can be considered as based on the 300 best terms. Comparable scatter plots of the dataset built on 600 and 300 terms are shown in Figure A.2, marked by circles and plus signs, respectively.

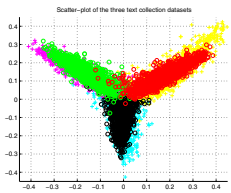

Fig. A.2. Scatter-plot of the three text collection datasets built on 300 and 600 terms

A.2.3 Iris Flower Dataset

Another real dataset chosen is the well-known Iris flower dataset or Fisher's Iris dataset available, for example, at
http : *//archive.ics.uci.edu/ml/datasets/Iris*.
The collection includes 50 samples from each of three species of Iris flowers:

- I. setosa;
- I. virginica;
- I. versicolor.

These species comprise three clusters situated so that one cluster is linearly separable from the others, but the other two are not separable from each other. This dataset has been analyzed in many studies. A two-cluster structure was detected in [278].

A.2.4 The Wine Recognition Dataset

The third real dataset contains 178 results of a chemical analysis of three different types (cultivates) of wine according to their 13 ingredients. This collection is available at

http://archive.ics.uci.edu/ml/machine-learning-databases/wine.

This data collection is relatively small, yet exhibits a high dimension.

A.2.5 The Glass Dataset

This dataset is taken from the UC Irvine Machine Learning Repository collection (http://archive.ics.uci.edu/ml/index.html). The study of the classification of glass types was motivated by criminology investigation. The glass found at the scene of a crime can be used as evidence.

Number of Instances: 214.

Number of Attributes: 9.

Type of glass: (class attribute)

- $building\_windows\_float\_processed$;
- $building\_windows\_non\_float\_processed$;
- $vehicle\_windows\_float\_processed$;
- $vehicle\_windows\_non\_float\_processed(notpresented)$;
- $containers$;
- $tableware$;
- $headlamps$.

A.2.6 The Flea-Beetles Dataset

This dataset contains the genus of the 74 flea beetle Chaetocnema, which contains three species:

- concinna (Con),
- heikertingeri (Hei),
- heptapotamica (Hep).

It is available at
http://lib.stat.cmu.edu/DASL/Datafiles/FleaBeetles.html.

References

1. Achlioptas, D.: Proc. ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, pp. 274–281 (2001)
2. Agarwal, P.K., Procopiuc, C.M.: Algorithmica 33(2), 201–226 (2002)
3. Aizerman, M.A., Braverman, E.M., Rozonoer, L.I.: Automat. Remote Contr. 25, 821–837 (1964)
4. Akaike, H.: Biometrika 60, 255–265 (1973)
5. Akaike, H.: IEEE Trans. on Automat. Contr. 19(6), 716–723 (1974)
6. Albert, A.: Regression, and the Moor-Penrose Pseudoinverse. Academic Press, New York (1972)
7. Alexandrov, A.G.: Autom. Remote Contr. 71(6), 977–992 (2010)
8. Allen, M.J.: Proc. AIAA 2006-1510, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (2006)
9. Alspector, J., Meir, R., Jayakumar, A., Lippe, D.: Hanson, S.J., Cowan, J.D., Lee, C. (eds.) Advances in Neural Information Processing Systems, San Mateo, CA, pp. 834–844. Morgan Kaufmann Publishers Inc. (1993)
10. Amelin, K.S.: Cybernetics and Physics 1(2), 79–88 (2012)
11. Amelin, K., Amelina, N., Granichin, O., Granichina, O.: Proc. of the 51st IEEE Conference on Decision and Control, Maui, Hawaii, USA, December 10-13, pp. 2134–2139 (2012)
12. Amelin, K., Amelina, N., Granichin, O., Granichina, O., Andrievsky, B.: Proc. of the 11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing (ALCOS 2013), Caen, France, July 3-5, pp. 205–208 (2013)
13. Amelin, K.S., Granichin, O.N.: Gyroscopy and Navigation 2(3), 277–284 (2011)
14. Amelin, K.S., Granichin, O.N.: Proc. of the 2012 American Control Conf., Montreal, Canada, pp. 851–856 (2012)
15. Amelin, K.S., Granichin, O.N.: IEEE Trans. Autom. Control (submitted 2014)
16. Anderson, N.H., Hall, P., Titterington, M.: J. of Multivariate Analysis 50(1):41–54 (1994)
17. Andrievsky, B.R., Matveev, A.S., Fradkov, A.L.: Autom. Remote Control 71(4), 71(12), 572–633 (2010)
18. Antal, C., Granichin, O., Levi, S.: Proc. of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, pp. 3656–3661 (2010)

19. Aronszajn, N.: Trans. of the American Mathematical Society 68 (1950)
20. Avros, R., Granichin, O., Shalymov, D., Volkovich, Z., Weber, G.W.: Holmes, D.E., Jain, L.C. (eds.) Data Mining: Found and Intell Paradigms, vol. 1, pp. 131–155. Springer, Heidelberg (2012)
21. Bai, E.W., Nagpal, K.M., Tempo, R.: Automatica 32, 985–999 (1996)
22. Baker, D., McCallum, A.: In: Van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (eds.) Proc. of SIGIR 1998, 21st ACM International Conference on ReResearch and Development in Information Retrieval, Melbourne, Australia, pp. 96–103 (1998)
23. Banfield, J.D., Raftery, A.E.: Biometrics 49, 803–821 (1993)
24. Barabanov, A.E., Granichin, O.N.: Autom. Remote Contr. 45(5), 578–584 (1984)
25. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: Constructive Approximation 28(3), 253–263 (2007)
26. Baringhaus, L., Franz, C.: J. of Multivariate Analysis 88(1),190–206 (2004)
27. Barmish, B.R., Lagoa, C.M.: Math. Control, Signals, Syst. 10, 203–222 (1997)
28. Baron, D., Sarvotham, S., Baraniuk, R.G.: IEEE Tran. on Signal Processing 58(1), 269–280 (2010)
29. Barty, K., Roy, J.-S., Strugarek, C.: Mathematical Programming 119(1), 51–78 (2009)
30. Barzily, Z., Volkovich, Z., Akteke-Ozturk, B., Weber, G.-W.: Informatica 20, 187–202 (2009)
31. Beck, A., Teboulle, M.: SIAM Journal on Imaging Sciences 2(1), 183–202 (2009)
32. Bellman, R.: Bulletin of the American Math. Soc. 60(6), 503–516 (1954)
33. Belopolskaya, Y., Klebanov, L., Volkovich, V.: J. of Mathematical Sciences 127(1), 1682–1686 (2005)
34. Ben-Hur, A., Elisseeff, A., Guyon, I.: Pacific Symposium on Biocomputing, pp. 6–17 (2002)
35. Ben-Hur, A., Guyon, I.: Methods in Molecular Biology, 159–182 (2003)
36. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: J. of Machine Learning Research 2, 125–137 (2001)
37. Berg, C., Christensen, J.P.R., Ressel, P.: Harmonic Analysis on Semigroups. Springer (1984)
38. Berinde, R., Indyk, P.: Proc. 47th Annual Allerton Conf. on Communication, Control, and Computing, pp. 36–43, Allerton (2009)
39. Berinde, R., Indyk, P., Ruzic, M.: Proc. 46th Annual Allerton Conf on Communication, Control, and Computing, pp. 198–205. Allerton (2008)
40. Biernacki, C.: Choix de Modeles en Classification. PhD Thesis, Universite de Technologie de Compiegne (1997)
41. Biernacki, C., Celeux, G., Govaert, G.: IEEE Trans. on Pattern Analysis and Machine Intelligence 22(7), 719–725 (2000)
42. Biernacki, C., Celeux, G., Govaert, G.: Pattern Recognition Letters 20, 267–272 (1999)
43. Bioucas-Dias, J., Figueiredo, M.: IEEE Trans. on Image Processing 16(12), 2992–3004 (2007)
44. Blanchini, F., Miani, S.: Set-Theoretic Methods in Control. Birkhauser, Boston (2008)
45. Blanchini, F., Sznaier, M.: IEEE Trans. Automat. Contr. 57(1), 219–224 (2012)

46. Blondel, V.D., Tsitsiklis, J.N.: Automatica 36, 1249–1274 (2000)
47. Blum, J.R.: Ann. Math. Statist. 9, 737–744 (1954)
48. Blumensath, T., Davies, M.E.: IEEE Trans. on Signal Processing 56(6), 2370–2382 (2008)
49. Borkar, V.S.: Stochastic approximation. A dynamical systems viewpoint. Cambridge University Press (2008)
50. Boufounos, P.T., Baraniuk, R.G.: Proc. 42nd Conf. Inf. Sc. Systems (CISS), Princeton, NJ, pp. 19–21 (2008)
51. Bozdogan, H.: Psychometrika 52(3), 345–370 (1987)
52. Bozdogan, H.: Proc. of the First US/Japan Conf. on the Frontiers of Statistical Modeling: An Informational Approach, Boston, vol. 2, pp. 69–113. Kluwer Academic Publishers (1994)
53. Braatz, R.P., Young, P.M., Doyle, J.C., Morari, M.: IEEE Trans. Autom. Control 39, 1000–1002 (1994)
54. Bredies, K., Lorenz, D.A.: SIAM J. on Scientific Computing 30(2), 657–683 (2008)
55. Breckenridge, J.: Multivariate Behavioral Research 24, 147–161 (1989)
56. Bucy, R.S., Kalman, R.E.: J. Basic Eng. ASME 83(1), 95–108 (1961)
57. Calafiore, G.C., Campi, M.C.: IEEE Trans. Automat. Control 51(5), 742–753 (2006)
58. Calafiore, G., Dabbene, F., Tempo, R.: Automatica 47, 1279–1293 (2011)
59. Calafiore, G., Polyak, B.T.: IEEE Trans. Autom. Control 46, 1755–1759 (2001)
60. Calinski, R., Harabasz, J.: Commun Statistics (3), 1–27 (1974)
61. Campi, M.C.: European Journal of Control 5, 419–430 (2010)
62. Campi, M.C., Sangho, K., Weyer, E.: Automatica 45, 2175–2186 (2009)
63. Campi, M.C., Weyer, E.: IEEE Trans. Automat. Control 55(12), 2708–2720 (2010)
64. Candes, E., Romberg, J.: Inverse Problems 23(3), 969–985 (2007)
65. Candes, E., Romberg, J., Tao, T.: IEEE Trans. Inform. Theor. 52(2), 489–509 (2006)
66. Candes, E., Wakin, M.: IEEE Signal Processing Magazine 25(2), 21–30 (2008)
67. Celeux, G., Govaert, G.: Computational Statistics and Data Analysis 14(3), 15, 315–332 (1992)
68. Celeux, G., Soromenho, G.: J. of Classification 13(2),195–212 (1996)
69. Cevher, V., Duarte, M., Hegde, C., Baraniuk, R.: Proc. of the Workshop on Neural Inform Processing Systems, NIPS (2008)
70. Chakravarthy, S.V., Ghosh, J.: IEEE Trans on Neural Networks 7(5), 1250–1261 (1996)
71. Chan, P., Schlag, M., Zien, J.: IEEE Trans. CAD-Integrated Circuits and Systems 13, 1088–1096 (1994)
72. Chartrand, R.: IEEE Signal Process. Lett. 14, 707–710 (2007)
73. Chartrand, R.: Proc. IEEE Int. Sympos. on Biomedical Imaging (ISBI), pp. 1349–1360 (2009)
74. Chartrand, R., Yin, W.: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, Nevada, USA (2008)
75. Chen, H.-F.: Ann. Statist. 16, 1330–1334 (1988)
76. Chen, H.-F., Duncan, T.E., Pasik-Duncan, B.A.: IEEE Trans. Automat. Contr. 44(3), 442–453 (1999)
77. Chen, H.-F., Guo, L.: Int. J. of Control 44(5), 1459–1477 (1986)

78. Chen, S.S., Donoho, D.L., Saunders, M.A.: SIAM J. Scientific Computing 43(1), 129–159 (2001)
79. Cheng, R., Milligan, G.W.: J. of Classification 13, 315–335 (1996)
80. Chretien, S.: IEEE Signal Processing Letters 17(2), 181–184 (2010)
81. Cohen, A., Dahmen, W., DeVore, R.: Instance Optimal Decoding by Thresholding in Compressed Sensing. Preprint (2008), http://www.igpm.rwth-aachen.de/Download/reports/pdf/IGPM289.pdf
82. Colorni, A., Dorigo, M., Maniezzo, V.: Actes de la Premiere Conf. Europeenne sur la vie Artificielle, Paris, France, pp. 134–142. Elsevier Publishing (1991)
83. Combettes, P.L., Wajs, V.R.: SIAM Journal on Multiscale Modeling and Simulation 4(4), 1168–1200 (2005)
84. Conover, W.J., Johnson, M.E., Johnson, M.M.: Technometrics 23, 351–361 (1981)
85. Cormode, G., Muthukrishnan, S.: Combinatorial Algorithms for Compressed Sensing. Springer, Heidelberg (2006)
86. Cortes, C., Vapnik, V.: Machine Learning, 273–297 (1995)
87. Cuevas, A., Febrero, M., Fraiman, R.: The Canadian J. of Statistics 28(2), 367–382 (2000)
88. Cuevas, A., Febrero, M., Fraiman, R.: Computational Statistics and Data Analysis 28, 441–459 (2001)
89. Davenport, M.A., Wakin, M.B.: IEEE Trans. on Inform. Theory 56(9), 4395–4401 (2010)
90. Dempster, A.P., Laird, N.M., Rubin, D.B.: J. of the Royal Statistical Society B 39, 1–38 (1977)
91. Derevitskii, D.P., Fradkov, A.L.: Autom. Remot. Contr. (1), 67–75 (1974)
92. Derevitskii, D.P., Fradkov, A.L.: Izvestiya AN USSR TK 5, 93–99 (1975)
93. Deutsch, D.: Proc. of the Royal Society of London Ser. A 400, 97–117 (1985)
94. Dhillon, I., Kogan, J., Nicholas, C.: A Comprehensive Survey of Text Mining, pp. 73–100. Springer, Heidelberg (2003)
95. Dhillon, I.S., Modha, D.S.: Machine Learning 42(1), 143–175 (2001)
96. Diaconis, P.: Bull. AMS 46(9), 179–205 (2009)
97. Donoho, D.: IEEE Trans. Inform. Theory 52(4), 1289–1306 (2006)
98. Duarte, M.F., Davenport, M.A., Takhar, D., Laska, J.N., Ting, S., Kelly, K.F., Baraniuk, R.G.: IEEE Signal Processing Magazine 25(2), 83–91 (2008)
99. Dudoit, S., Fridlyand, J.: Genome Biology 3(7), 112–129 (2002)
100. Dunn, J.C.: J. Cybern. 4, 95–104 (1974)
101. Duran, B.S.: Communications in Statistics – Theory and Methods 5, 1287–1312 (1976)
102. Egiazarian, K., Foi, A., Katkovnik, V.: In: Proc. IEEE Int Conf on Image Processing ICIP, San Antonio, Texas, USA, pp. 549–552 (2007)
103. Ermakov, S.M.: Die Monte Carlo Methode und Verwandte Fragen. VEB Deutscher Verlag der Wissenschaften, Berlin (1975)
104. Estivill-Castro, V., Yang, J.: Pacific Rim International Conference on Artificial Intelligence, pp. 208–218 (2000)
105. Everritt, B.S., Hand, D.J.: Finite Mixture Distributions. Chapman and Hall, London (1981)
106. Everritt, B.S., Landau, S., Leese, M., Stahl, D.: Miscellaneous Clustering Methods in Cluster Analysis. John Wiley and Sons, Chichester (2011)
107. Fabian, V.: Ann. Math. Statist. 38, 191–200 (1967)

108. Fedin, D.S., Granichin, O.N., Dedkov, Y.S., Molodtsov, S.L.: Rev. Sci. Instr. 79, 036103 (2008)
109. Feldbaum, A.A.: Autom. Remote Control 21, 874–880, 1033–1039, 22, 1–12, 109–121 (1960-61)
110. Feng, Y., Hamerly, G.: Proc. of the 12th Annual Conf. on Neural Information Processing Systems, NIPS (2006)
111. Figueiredo, M.A.T., Jain, A.K.: IEEE Trans. on Pattern Analysis and Machine Intelligence 24(3), 1–16 (2002)
112. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: Pattern Recognition 41(1), 176–190 (2008)
113. Fisher, R.A.: The design of experiments. Oliver and Boyd, Edinburgh (1935)
114. Fonseca Jaime, R.S.: Proc. of the 2008 8th Int. Conf. on Hybrid Intelligent Systems, pp. 543–548. IEEE Computer Society (2008)
115. Fonseca Jaime, R.S., Cardoso, M.G.M.S.: Intell. Data Anal. 11, 155–173 (2007)
116. Fomin, V.N.: Matematical Theory of the Learning Systems of Identifying. Leningrad University Publisher, Leningrad (1976) (in Russian)
117. Fomin, V.N., Fradkov, A.L., Yakubovich, V.A.: Adaptive Control of Dynamical Plants. Nauka, Moscow (1981) (in Russian)
118. Fomin, V.N.: Recurrent Estimation and Adaptive Filtering. Nauka, Moscow (1984) (in Russian)
119. Friedman, J.H., Rafsky, L.C.: Annals of Statistics 7, 697–717 (1979)
120. Forgy, E.W.: Biometrics 21(3), 768 (1965)
121. Fowlkes, E.W., Mallows, C.L.: J. Am. Stat. Assoc. 78, 553–584 (1983)
122. Fraley, C., Raftery, A.E.: Computer Journal 41(8), 578-588 (1998)
123. Fridlyand, J., Dudoit, S.: Application of Resampling Methods to Estimate the Number of Clusters and to Improve the Accuracy of a Clustering Methods. Stat. Berkely Tech. Report 600 (2001)
124. Fujisaki, Y., Dabbene, F., Tempo, R.: Automatica 39, 1323–1337 (2001)
125. Gibbs, J.W.: Elementary Principles in Statistical Mechanics. Woodbridge, Connecticut (1902)
126. Gerencser, L.: IEEE Trans. on Automat. Control 44, 894–905 (1999)
127. Girolami, M.: IEEE Trans. on Neural Networks 13(3), 780–784 (2002)
128. Glover, F., McMillan, C.: Computers and Operations Research 13(5), 563–573 (1986)
129. Gokcay, E., Principe, J.C.: IEEE Trans. on Pattern Analysis and Machine Intelligence 24(2), 158–171 (2002)
130. Goldenshluger, A.V., Polyak, B.T.: Mathematical Methods of Statistics 2(1), 18–29 (1993)
131. Gordon, A.D.: Computational Statistics and Data Analysis 18 (1994)
132. Granichin, O.N.: Proc. of All-Union Conf. Theory of Adaptive Systems and Its Applications L VINITI 26 (1983) (in Russian)
133. Granichin, O.N.: Vestnik Leningrad University: Math. 21(3), 56–58 (1988)
134. Granichin, O.N.: Vestnik Leningrad University: Math. 22(1), 27–31 (1989)
135. Granichin, O.N.: Autom. Remote Control 53(2), 232–237 (1992)
136. Granichin, O.N.: Autom. Remote Control 62(3), 422–429 (2001)
137. Granichin, O.N.: Autom. Remote Contr. 63(1), 25–35 (2002)
138. Granichin, O.N.: Autom. Remote Contr. 63(2), 209–219 (2002)
139. Granichin, O.N.: Autom. Remote Contr. 63(9), 1482–1488 (2002)
140. Granichin, O.N.: Autom. Remote Contr. 64(2), 252–262 (2003)
141. Granichin, O.N.: IEEE Trans. Automat. Contr. 49(10), 1830–1835 (2004)

142. Granichin, O.N.: Autom. Remote Contr. 72(12), 2578–2582 (2011)
143. Granichin, O.N.: Autom. Remote Contr. 73(1), 20–30 (2012)
144. Granichin, O.N., Amelina, N.O.: IEEE Trans. Automat. Contr. (2014) (cond. accepted)
145. Granichin, O.N., Fomin, V.N.: Autom. Remote Contr. 47(2), 238–248 (1986)
146. Granichin, O.N., Izmakova, O.A.: Autom. Remote Contr. 66(8), 1239–1248 (2005)
147. Granichin, O.N., Molodtsov, S.L.: Stochasticheskaya Optimizatsiya v Informatike 2, 219–253 (2006) (in Russian)
148. Granichin, O.N., Polyak, B.T.: Randomized Algorithms of an Estimation and Optimization under Almost Arbitrary Noises. Nauka, Moscow (2003) (in Russian)
149. Granichin, O., Shalymov, D., Avros, R., Volkovich, Z.: Autom. Remote Contr. 72(4), 754-765 (2011)
150. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: Advances in Neural Information Processing Systems, vol. 19, pp. 513–520. MIT Press, Cambridge (2007)
151. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: Procs of the 22nd Conf. on Artificial Intelligence (AAAI 2007), pp. 1637–1641 (2007)
152. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A Kernel Method for the Two-Sample Problem, CoRR, abs/0805.2368 DBLP (2008), http://arxiv.org/abs/0805.2368, http://dblp.uni-trier.de
153. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: J of Machine Learning Research 4, 1-10 (2008)
154. Guo, L., Ljung, L., Wang, G.J.: IEEE Trans. on Automat Contr. 42, 761–770 (1997)
155. Hall, P., Tajvidi, N.: Biometrika 89(2), 359–374 (2002)
156. Halpern, M.E., Evans, R.J.: Systems & Control Letters 42(4), 253–260 (2001)
157. Hamerly, G., Elkan, C.: Proc. of the Seventeenth Annual Conf. on Neural Information Processing Systems (NIPS), pp. 281–288 (2003)
158. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2001)
159. Hartigan, J.: J. Classification 2 (1985)
160. Hartigan, J.: Clustering Algorithms. John Wiley, New York (1975)
161. Haussler, D.: Convolution Kernels on Discrete Structures. UCSC-CRL-9910. Department of Computer Science University of California (1999)
162. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1984)
163. Henze, N.: Annals of Statistics 16, 772–783 (1988)
164. Henze, N., Penrose, M.: Ann. of Statistics 27, 290–298 (1999)
165. Hill, R.D.: Systems and Control Letters 57(6), 489–496 (2008)
166. Hoeffding, W.: J. Am. Stat. Assoc. 58, 13–30 (1963)
167. Hofmann, T.: Proc. ACM SIGIR. ACM Press, New York (August 1999)
168. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
169. Hubert, L., Schultz, J.: J. Math. Statist. Psychol. 76, 190–241 (1974)
170. Hurvich, C.M., Tsai, C.-L.: Biometrika 76(2), 297–307 (1989)
171. Ishii, H., Basar, T., Tempo, R.: Automatica 40, 839–846 (2004)
172. Jaccard, P.: Bulletin de la Society Vaudoise des Sciences Naturelles 37, 547–579 (1901)

173. Jafarpour, S., Xu, W., Hassibi, B., Calderbank, R.: IEEE Trans. Inform. Theory 55(9), 4299–4308 (2009)
174. Jain, A.K., Moreau, J.V.: Pattern Recognition 20, 547–568 (1987)
175. Jain, A.K., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
176. Jain, A.K., Xu, X., Ho, T.K., Xiao, F.: Proc. ICPR 4, 281–284 (2002)
177. Jimenez-Lizarraga, M., Poznyak, A., Alcorta, M.A.: Proc. American Control Conf. (2008)
178. Johnson, W., Lindenstrauss, J.: Proc. Conf. in Modern Analysis and Probability, pp. 189–206 (1984)
179. Juditsky, A.: IEEE Trans. Automat Contr. 38, 794–798 (1993)
180. Juditsky, A., Nemirovski, A.: On Verifiable Sufficient Conditions for Sparse Signal Recovery via ℓ_1 minimization (2010), Preprint `http://arxiv.org/abs/0809.2650`
181. Kanev, S., De Schutter, B., Verhaegen, M.: Proc. of Conf. on Decision and Control, Las Vegas, USA, pp. 2248–2253 (2002)
182. Karp, R.M.: Discrete Applied Mathematics 34, 165–201 (1991)
183. Kashin, B.S.: Izvestiya AN USSR Ser. Math. 42(2), 334–351 (1977)
184. Kass, R.E.: J. Amer. Statist. Assoc. 90, 928–934 (1995)
185. Katkovnik, V.Y.: Linear Estimation and Stochastic Optimization Problems. Nauka, Moscow (1976) (in Russian)
186. Katkovnik, V., Egiazarian, K.: Proc. Int. Workshop on Local and Non-Local Approximation in Image Processing, pp. 46–53 (2009)
187. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons Inc., New York (1990)
188. Kass, R.E., Raftery, A.: J. Am. Stat. Assoc. 90, 773–795 (1995)
189. Kenny, A., Stylometric, A.: Study of the New Testament. Oxford University Press, USA (1986)
190. Khargonekar, P.P., Tikku, A.: Proc. of Conf. on Decision and Control, pp. 3470–3475 (1996)
191. Khlebnikov, M.V., Polyak, B.T., Kuntsevich, V.M.: Autom. Remote Contr. 72(11), 2227–2275 (2011)
192. Kiefer, J., Wolfowitz, J.: Annals of Math Statist 23, 462–466 (1952)
193. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Science 220, 671–680 (1983)
194. Klebanov, L.: One Class of Distribution Free Multivariate Tests. SPb Math Society Preprint (2003)
195. Klebanov, L.B.: N-distances and Their Applications. Charsel University in Prague. The Karolinum Press (2005)
196. Klebanov, L., Kozubowskii, T., Rachev, S., Volkovich, V.: Statistics and Probability Letters 53, 241–247 (2001)
197. Kogan, J., Nicholas, C., Volkovich, V.: Berry, M.W., Pottenger, W.M. (eds.) Proc. of the Workshop on Text Mining held in Conjunction with the Third SIAM International Conference on Data Mining, pp. 5–16 (2003)
198. Kogan, J., Nicholas, C., Volkovich, V.: Computing in Science and Engineering, 52–59 (2003)
199. Kogan, J., Teboulle, M., Nicholas, C.: Proc. of the Workshop on Clustering High Dimensional Data and its Applications (2003)
200. Krylov, N.M., Bogoliubov, N.N.: Introduction to Nonlinear Mechanics. Princeton University Press, Princeton (1947)
201. Krzanowski, W., Lai, Y.: Biometrics 44, 23–34 (1985)

202. Kuhn, H.: Naval Research Logistics Quarterly 2, 83–97 (1955)
203. Kulchitskii, O.Y.: Probl. Inform. Trans. 21(4), 49–63 (1985)
204. Kushner, H.J.: IEEE Trans. Automat. Control 6, 921–930 (1977)
205. Kushner, H.J., Clark, D.S.: Stochastic Approximation Methods for Constraine and Unconstraind System. Springer, Berlin (1978)
206. Kushner, H.J., Yin, G.G.: Stochastic approximation algorithms and recursive algorithms and applications. Springer, New York (2003)
207. Kupfmuller, K.: Elektrische Nachrichtentechnik 5(11), 459–467 (1928)
208. Lange, T., Roth, V., Braun, L.M., Buhmann, J.M.: Neural Computation 16(6), 1299–1323 (2004)
209. Lange, T., Braun, M., Roth, V., Buhmann, J.M.: Advances in Neural Information Processing Systems, vol. 15 (2003)
210. Lebesgue, H.: Integrale, Longueur, Aire. Universite de Paris (1902)
211. Ledger, G.: An Exploration of Dierences in the Pauline Epistles using Multivariate Statistical Analysis. Oxford University Press (1995)
212. Lember, J.: Journal of Approximation Theory 120, 20–35 (2003)
213. Levine, E., Domany, E.: Neural Computation 13, 2573–2593 (2001)
214. Ljung, L.: IEEE Trans. Automat Contr. (4):551–575 (1977)
215. Ljung, L., Söderström, T.: Theory and Practice of Recursive Identification. MIT Press, Cambridge (1983)
216. Ljung, L., Guo, L.: IEEE Trans. Automat. Contr. 42, 1230–1239 (1997)
217. Ljung, L.: System Identification – Theory for the User, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
218. Lord, R.: Biometrica 282 (1958)
219. Lukacs, E.: Characteristic Functions. Griffin (1970)
220. Lustig, M., Donoho, D.L., Santos, J.M., Pauly, J.M.: IEEE Signal Processing Magazine 25(2), 72–82 (2008)
221. Luxburg, U.V.: Statistics and Computing 17(4), 395–416 (2007)
222. Ma, J., Le Dimet, F.X.: IEEE Trans. on Geoscience and Remote Sensing 47(3), 792–802 (2009)
223. Maeda, Y., Kanata, Y.: Trans. of IEE of Japan 113-C, 402–408 (1993)
224. Maeda, Y., De Figueiredo, R.J.: IEEE Trans. on Neural Networks 8, 1119–1130 (1997)
225. Mallat, S., Zhang, Z.: IEEE Trans. on Signal Processing 41(12), 3397–3415 (1993)
226. Mardia, J., Kent, K., Bibby, J.: Multivariate Analysis. Academic Press, San Diego (1979)
227. McQuarrie, A., Shumway, R., Tsai, C.-L.: Probability Letters 34, 285–292 (1997)
228. McEneaney, W.M.: Sys. Contr. Lett. 33, 315–325 (1998)
229. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, Chichester (2000)
230. McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extentions. Wiley, Chichester (1996)
231. Mealand, D.L.: JSNT 59, 61–92 (1995)
232. Meerkov, C.M.: Autom. Remote Contr. (3), 6–75 (5):63–67 (1972)
233. Merris, R.: Linear Algebra and its Applications, 197–198, 143–176 (1994)
234. Metropolis, N., Ulam, S.: J. Am. Stat. Assoc. 44, 335–341 (1949)
235. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: J. Chemical Physics 21(6), 1087–1092 (1953)
236. Milligan, G., Cooper, M.: Psychometrika 50 (1985)

237. Mishali, M., Eldar, Y.C.: IEEE Trans. on Signal Processing 56(10), 4692–4702 (2008)
238. Mohimani, H., Babaie-Zadeh, M., Jutten, C.: IEEE Trans. Signal Processing 57(1), 289–301 (2009)
239. Moore, H.: Electronics 38(8) (1965)
240. Morozkov, M., Granichin, O., Volkovich, Z., Zhang, X.: Proc. of 2012 24th Chinese Control and Decision Conf. (CCDC), pp. 2013–2018 (2012)
241. Mufti, G.B., Bertrand, P., Moubarki, L.: Proc. of ASMDA 2005, pp. 404–414 (2005)
242. Nascimento, M., Carvalho, A.D.: European Journal of Operational Research 2116(2), 221–231 (2011)
243. Needell, D., Vershynin, R.: Foundations of Computational Mathematics 9, 317–334 (2009)
244. Needell, D., Tropp, J.A.: Appl. Comp. Harmonic Anal. 26, 301–321 (2008)
245. Nemirovski, A.: J. Matrix Anal. Appl. 6, 99–105 (1993)
246. Nesetril, J., Milkova, E., Nesetrilova, H.: Discrete Mathematics 1, 3–36 (2001)
247. Nevel'son, M.B., Khasminskii, R.Z.: Stochastic Approximation and Recursive Estimation. Amer. Math. Soc., Providence (1973)
248. Ng, A., Jordan, M., Weiss, Y.: Advances in Neural Information Processing Systems 14 (2001)
249. Nyquist, H.: Trans. AIEE 47, 617–644 (1928)
250. Oishi, Y., Kimura, H.: Proc. Conf. on Decision and Control, Orlando, USA, pp. 2025–2030 (2001)
251. Parzen, E.: Ann. Math. Stat. 32, 1065–1076 (1962)
252. Pascual, D., Pla, F., Sanche, J.S.: Pattern Recognition Letters 31, 454–461 (2010)
253. Pelleg, D., Moore, A.: Proc. of the 17th Int. Conf. on Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)
254. Pereira, F.C.N., Tishby, N., Lee, L.: Meeting of the Association for Computational Linguistics, pp. 183–190 (1993)
255. Perkins, P.: Reading the New Testament: An Introduction, vol. 47. Paulist Press (1988)
256. Pollard, D.: Annals of Statistics 9(1), 135–140 (1981)
257. Polyak, B.T.: Automat. Remote Contr. 38, 537–542 (1977)
258. Polyak, B.T.: Introduction to Optimization. Optimization Software, NewYork (1987)
259. Polyak, B.T.: Automat. and Remote Contr. 51, 937–946 (1990)
260. Polyak, B., Gryazina, E.: European Journal of Operational Research (submitted 2014)
261. Polyak, B.T., Sherbakov, P.S.: Robust Stability and Control. Nauka, Moscow (2002) (in Russian)
262. Polyak, B.T., Tempo, R.: Syst. Control Lett. 43, 343–353 (2001)
263. Polyak, B.T., Tsybakov, A.B.: Problems Inform. Transmission 26(2), 126–133 (1990)
264. Polyak, B.T., Tsybakov, A.B.: Khasminskii, R.Z. (ed.) Topics in Nonparametric Estimation (1992); Advances in Soviet Mathematics, Amer. Math. Soc., Providence 12, 107–113
265. Polyak, B.T., Tsypkin, Y.Z.: Automat. Remote Contr. 40, 378–389 (1979)
266. Poznyak, A., Azhmyakov, V., Mera, M.: Int. J. of Contr. 84(8), 1408–1416 (2011)

267. Poznyak, A.S., Sanches, E.N., Wen, Y.: Dynamic Neural Networks for Nonlinear Control: Identification, State Estimation and Trajectory Tracking. World Scientific (2001)
268. Principe, J., Xu, D., Fisher, J.: Information Theoretic Learning, Unsupervised Adaptive Filtering I, pp. 265–319. John Wiley & Sons, New York (2002)
269. Rachev, S.T.: Probability Metrics and the Stability of Stochastic Models. Wiley Series in Probability and Mathematical Statistics, Chichester (1991)
270. Rand, W.: J. Am. Stat. Assoc. 66, 846–850 (1971)
271. Rastrigin, L.A.: Autom. Remote Contr. 24(10), 1337–1342 (1963)
272. Rissanen, J.: Automatica 14, 465–471 (1978)
273. Robbins, H., Monro, S.: Annals of Math. Statist. 22(3), 400–407 (1951)
274. Robbins, H., Siegmund, D.: In: Rustagi, J.S. (ed.) Optimizing Methods in Statistics, pp. 233–257. Academic Press, New York (1971)
275. Robert, J., Torbjorn, E.: Neurocomputing 72(1-3), 23–31 (2008)
276. Rose, K., Gurewitz, E., Fox, G.: Physical Review Letters 65(8), 945–948 (1990)
277. Rosenbaum, P.: J. of the Royal Statistical Society B 67(4), 515–530 (2005)
278. Roth, V., Lange, T., Braun, M., Buhmann, J.: A Resampling Approach to Cluster Validation, COMPSTAT (2002), http://www.cs.uni-bonn.De/~braunm
279. Rousseeuw, P.J.: J. of Computational and Applied Mathematics 20(1), 53–65 (1987)
280. Rzevski, G.: Proc. of 13th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2012), Kyoto, Japan, August 8-10, pp. 434–437 (2012)
281. Sadegh, P., Spall, J.C.: IEEE Trans. on Automat Contr. 43(10), 1480–1484 (1998)
282. Saridis, G.M., Stein, G.: IEEE Trans. on Automat. Contr. AC-13(4), 592–584 (1968)
283. Saridis, G.M., Lobbia, R.N.: IEEE Trans. on Automat. Contr. AC-17(1), 52–60 (1972)
284. Sato, K.: Levy Processes and Infinitely Divisible Distributions. Cambridge University Press (1999)
285. Schoenberg, I.J.: Trans. of the American Mathematical Society 44(3), 522–536 (1938)
286. Schölkopf, B.: Proc. of the Workshop on Neural Inform Processing Systems (NIPS), pp. 301–307 (2000)
287. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press (2002)
288. Schölkopf, B., Smola, A.J., Muller, K.-R.: Neural Computation 10(5), 1299–1319 (1998)
289. Schwartz, G.: Annals of Statist. 6(2), 461–464 (1978)
290. Shannon, C.E.: Proc. Institute of Radio Engineers, vol. 37(1), pp. 10–21 (1949)
291. Shi, J., Malik, J.: IEEE Trans. Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
292. Shiryaev, A.N.: Probability, 2nd English edn. Springer (1995)
293. Shor, P.: SIAM J. Comput. 26, 1484–1509 (1997)
294. Slonim, N., Tisbby, N.: Proc. of the Workshop on Neural Inform. Processing Systems, NIPS-12 (1999)
295. Slonim, N., Tisbby, N.: Research and Development in Information Retrieval, pp. 208–215 (2000)
296. Smith, R.L.: Operations Res. 32, 1296–1308 (1984)

297. Smith, S.P., Jain, A.K.: IEEE Trans. Pattern Anal. Machine Intelligence 6(1), 73–80 (1984)
298. Sokolov, V.F.: Systems & Control Letters 6(2), 93-98 (1985)
299. Sokolov, V.F.: Automat. Remote Contr. 71(9), 1741–1756 (2010)
300. Spall, J.C.: IEEE Trans. Automat. Contr. 37(3), 332–341 (1992)
301. Spall, J.C.: Automatica 33, 109–112 (1997)
302. Spall, J.C.: Introduction to Stochastic Search and Optimization. Wiley & Sons, New Jersey (2003)
303. Steinwart, I.: J. Mach. Learn. Res. 2, 67–93 (2001)
304. Stengel, R.F., Ray, L.R.: IEEE Trans. Autom. Control 36, 82–87 (1991)
305. Stewart, G.W., Sun, J.G.: Matrix Perturbation Theory. Computer Science and Scientific Computing. Academic Press Inc., Boston (1990)
306. Still, S., Bialek, W.: Neural Computation 16(12), 2483–2506 (2004)
307. Stoer, M., Wagner, F.: J. ACM 44(4), 585–591 (1997)
308. Stuetzle, W.: J. Classification 20(5), 25–47 (2003)
309. Sugar, C., James, G.: J. of the Amer. Statistical Association 98, 750–763 (2003)
310. Tang, Q.-Y., Chen, H.-F., Han, Z.-J.: IEEE Trans. Automat. Contr. 42(10), 1442–1447 (1997)
311. Taubman, D.S., Marcellin, M.V.: JPEG 2000: Image Compression Fundamentals, Standards and Practice. Kluwer Academic, Norwell (2001)
312. Tempo, R., Calafiore, G., Dabbene, F.: Randomized Algorithms for Analysis and Control of Uncertain Systems, with Applications. Springer, London (2012)
313. Tibshirani, R.: J. Royal Statist. Soc. B 58(1), 267–288 (1996)
314. Tibshirani, R., Walther, G., Hastie, T.: J. Royal Statist. Soc. B 63(2), 411–423 (2001)
315. Tibshirani, R., Walther, G.: J. of Computational & Graphical Statistics 14(3), 511–528 (2005)
316. Tien, D.K.: Contemporary Physics 44, 51–71 (2003)
317. Tishby, N., Pereira, F., Bialek, W.: Proc. of the 37th Annual Allerton Conf. on Communication, Control and Computing, pp. 368–377 (1999)
318. Tishby, N., Slonim, N.: Proc. of the Workshop on Neural Inform. Processing Systems (NIPS), pp. 640–646 (2000)
319. Toledano-Kitai, D., Avros, R., Volkovich, Z., Weber, G.-W., Yahalom, O.: J. of Intelligent and Fuzzy Systems, Special Issue: Recent Advances in Intelligent and Fuzzy Systems (2012)
320. Topchy, A.P., Minaei-Bidgoli, B., Punch, W.F.: ITCC(2), pp. 188–192 (2004)
321. Tropp, J., Gilbert, A.C.: IEEE Trans. on Information Theory 53(12), 4655–4666 (2007)
322. Turchin, V.F.: Theory of Probability and its Applications 16(4) (1971)
323. Tsypkin, Y.Z.: Adaptation and Learning in Automatic Systems. Nauka, Moscow (1968) (in Russian)
324. Tsypkin, Y.Z.: Information Theory of an Identification. Nauka, Moscow (1995) (in Russian)
325. Vakhitov, A.T., Granichin, O.N., Sysoev, S.S.: Autom. Remote Contr. 67(4), 589–597 (2006)
326. Vakhitov, A.T., Granichin, O.N., Gurevich, L.S.: Autom. Remote Control 70(11), 1827–1835 (2009)
327. Vakhitov, A., Granichin, O., Vlasov, V.: Proc. of the 2010 American Control Conf., Baltimore, MD, USA, pp. 4004–4009 (2010)

328. Vapnik, V.N., Chervonenkis, A.Y.: Soviet Math. Doklady 9, 915–918 (1968)
329. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
330. Vetterli, M., Marziliano, P., Blu, T.: IEEE Trans. Signal Processing 50(6), 1417–1428 (2002)
331. Vidyasagar, M.: A Theory of Learning and Generalization with Applications to Neural Networks and Control Systems. Springer, London (1997)
332. Vidyasagar, M.: IEEE Control Systems 12, 69–85 (1998)
333. Volkovich, Z., Avros, R., Golani, M.: J. of Pattern Recognition Research 2, 230–243 (2011)
334. Volkovich, Z., Barzily, Z., Weber, G.W., Toledano-Kitai, D., Avros, R.: Machine Learning 85(1-2), 209–248 (2011)
335. Volkovich, Z., Barzily, Z., Avros, R., Toledano-Kitai, D.: Proc. of the XIII Int. Conf. Applied Stochastic Models and Data Analysis (ASMDA 2009), Vilnius (2009)
336. Volkovich, Z., Barzily, Z.: 1st European Conf. on Data Mining (ECDM 2007), Lisbon, Portugal, pp. 5–7 (2007)
337. Volkovich, Z., Barzily, Z., Morozensky, L.: Pattern Recognition 41, 2174–2188 (2008)
338. Volkovich, Z., Barzily, Z., Sureanu, P.: J. of Pure and Applied Mathematics 25(1), 103–121 (2005)
339. Volkovich, Z., Barzily, Z., Weber, G.-W., Toledano-Kitai, D., Avros, R.: Central European J. of Operation Research 20(1), 119–139 (2012)
340. Volkovich, V., Kogan, J., Nicholas, C.: Dhillon, I., Kogan, J. (eds.) Proc. of the Workshop on Clustering High Dimensional Data and its Applications (held in conjunction with SDM 2004), pp. 17–22 (2004)
341. Volkovich, Z., Weber, G.-W., Avros, R.: Proc. of the Third Global Conf on Power Control and Optimization (PCO 2010), Gold Coast, Australia (2010)
342. Volkovich, Y.V., Granichin, O.N.: Stochasticheskaya Optimizatsiya v Informatike 1, 17–28 (2005) (in Russian)
343. von Neumann, J.: Math. Annalen 100, 295–320 (1928)
344. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press (1944)
345. Wang, J., Chen, J.: Clustering to Maximize the Ratio of Split to Diameter. CoRR abs/1206.4605 (2012), http://arxiv.org/abs/1206.4605
346. Wakin, M., Laska, J., Duarte, M., Baron, D., Sarvotham, S., Takhar, D., Kelly, K., Baraniuk, R.: Proc. Picture Coding Sympos. PCS, Beijing (2006)
347. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman and Hall, London (1995)
348. Wang, I.-J., Chong, E.: IEEE Trans. on Automat. Contr. 43, 1745–1749 (1998)
349. Wasan, M.T.: Stochastic Approximation. Cambridge Univ. Press (1969)
350. White, H.: Artifical Neural Networks. Blackwell, Oxford (1992)
351. Wiener, N.: Cybernetics, or Communication and Control in the Animal and the Machine. MIT Press, Cambridge (1948)
352. Wilks, S.S.: Ann. Math. Statist. 9, 60–62 (1938)
353. Wishart, D.: Numerical Taxonomy 76, 282–311 (1969)
354. Eric, P.X., Andrew, Y.N., Michael, I.J., Russell, S.: Advances in Neural Information Processing Systems 15, pp. 505–512. MIT Press (2002)
355. Young, P.C.: Recursive Estimation and Time-Series Analysis. An Introduction. Springer, Heidelberg (1984)
356. Yu, S.X., Shi, J.: Proc. Int. Conf. on Computer Vision (2003)

357. Zech, G., Aslan, B.: The J. of Statistical Computation and Simulation 75(2), 109–119 (2005)
358. Zhigljavsky, A., Zilinskas, A.: Stochastic Global Optimization. Springer, Berlin (2008)
359. Zinger, A.A., Kakosyan, A.V., Klebanov, L.B.: Stability Problems for Stochastic Models, VNIISI, pp. 47–55 (1989)
360. Zolotarev, V.M.: Modern Theory of Summation of Random Variable. Brill Academic Publishers (1997)

Index