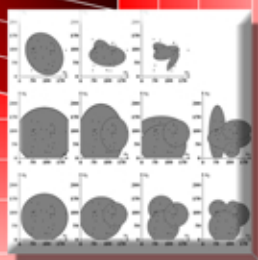
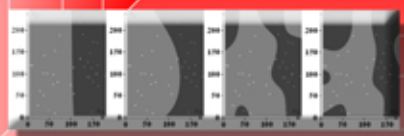
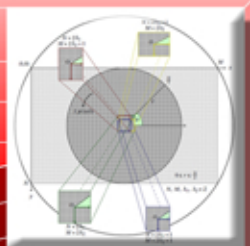


Pattern Recognition

Practices, Perspectives and Challenges

Darrell B. Vincent
Editor



Computer Science, Technology and Applications

NOVA

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

PATTERN RECOGNITION

PRACTICES, PERSPECTIVES

AND CHALLENGES

No part of this digital document may be reproduced, stored in a retrieval system or transmitted in any form or by any means. The publisher has taken reasonable care in the preparation of this digital document, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained herein. This digital document is sold with the clear understanding that the publisher is not engaged in rendering legal, medical or any other professional services.

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

Additional books in this series can be found on Nova's website
under the Series tab.

Additional e-books in this series can be found on Nova's website
under the e-book tab.

COMPUTER SCIENCE, TECHNOLOGY AND APPLICATIONS

PATTERN RECOGNITION
PRACTICES, PERSPECTIVES
AND CHALLENGES

DARRELL B. VINCENT
EDITOR



Copyright © 2013 by Nova Science Publishers, Inc.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system or transmitted in any form or by any means: electronic, electrostatic, magnetic, tape, mechanical photocopying, recording or otherwise without the written permission of the Publisher.

For permission to use material from this book please contact us:

Telephone 631-231-7269; Fax 631-231-8175

Web Site: <http://www.novapublishers.com>

NOTICE TO THE READER

The Publisher has taken reasonable care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained in this book. The Publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the readers' use of, or reliance upon, this material. Any parts of this book based on government reports are so indicated and copyright is claimed for those parts to the extent applicable to compilations of such works.

Independent verification should be sought for any data, advice or recommendations contained in this book. In addition, no responsibility is assumed by the publisher for any injury and/or damage to persons or property arising from any methods, products, instructions, ideas or otherwise contained in this publication.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered herein. It is sold with the clear understanding that the Publisher is not engaged in rendering legal or any other professional services. If legal or any other expert assistance is required, the services of a competent person should be sought. FROM A DECLARATION OF PARTICIPANTS JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Additional color graphics may be available in the e-book version of this book.

Library of Congress Cataloging-in-Publication Data

ISBN: ; 9: /3/8483: /3; : /6 (eBook)

Published by Nova Science Publishers, Inc. † New York

CONTENTS

Preface		vii
Chapter 1	Pattern Recognition Applied to Spectroscopy: Conventional Methods and Future Directions <i>Ana Paula Craig, Adriana S. Franca and Joseph Irudayaraj</i>	1
Chapter 2	Optimization of an Embedded Simplified Fuzzy ARTMAP Implemented on a Microcontroller Using MATLAB GUI Environment <i>Eduardo Garcia-Breijo, Jose Garrigues and Luis Gil-Sanchez</i>	47
Chapter 3	Application of Pattern Recognition in Optimization-Simulation Technique <i>G. M. Antonova</i>	91
Chapter 4	Practical Usage of Algorithmic Probability in Pattern Recognition <i>Alexey S. Potapov</i>	125
Chapter 5	Pattern Recognition Using Quaternion Color Moments <i>E. G. Karakasis, G. A. Papakostas and D. E. Koulouriotis</i>	153

Chapter 6	Pattern Recognition by Bessel Mask and One-Dimensional Signatures <i>Selene Solorza and Josue Alvarez-Borrego</i>	177
Index		185

PREFACE

In this book, the authors present current research in the study of the practices, perspectives and challenges of pattern recognition. Topics include the practical usage of algorithmic probability in pattern recognition; application of pattern recognition in optimization-simulation techniques; pattern recognition applied to spectroscopy; optimization of an embedded simplified fuzzy ARTMAP implemented on a microcontroller using MATLAB GUI environment; pattern recognition using quaternion color moments; and pattern recognition by Bessel mask and one-dimensional signatures.

Chapter 1 – Spectroscopic techniques have gained importance in a wide range of fields because of their appeal as rapid, reliable and nondestructive analysis, in most cases requiring minimum sample pre-treatment. These features make spectroscopy techniques suitable for routine analysis in on-line mode processing facilities. However, due to the high complexity of spectral data, which contain a large number of variables, multivariate statistical analysis is required to recognize patterns from samples. The multivariate nature of these methods makes evaluation of the robustness a much more complex task in comparison to classical ruggedness testing, as applied in univariate methods. In this review, unsupervised and supervised algorithms conventionally used for qualitative and quantitative analysis are explored. Among them, artificial neural networks, hierarchical clustering, linear regression extensions and principal component analysis are highlighted. Following the recent breakthrough in powerful and fast growing spectroscopic technologies, such as hyperspectral imaging, new challenges in pattern recognition are emerging. Thus, the authors present an overview of the new and promising developments in pattern recognition methods for complex

spectral data, including support vector machines and penalized regression methods for robust variable selection.

Chapter 2 – In the present work, a portable system based on a microcontroller has been developed to classify different kinds of honeys. In order to do this classification, a Simplified Fuzzy ARTMAP network (SFA) implemented in a microcontroller has been used. Due to the memory limits when working with microcontrollers, it is necessary to optimize the use of both program and data memory. In order to optimize the necessary parameters to programme the SFA in a microcontroller, a Graphical User Interface (GUI) for Matlab has been developed. The measures have been carried out by potentiometry techniques using a multielectrode of 7 different metals. With the information obtained in the experimental phase, the neural network has been trained in a PC by means of the GUI in MATLAB, with the obtained parameters the microcontroller has been programmed and later new samples have been analyzed using the portable system. An important aspect in this work is that the training algorithm has been implemented to the equipment in such a way that it can add the information of the new samples to the network in order to optimize it.

Chapter 3 – The method of approximate optimization of dynamic stochastic systems widely known as the optimization-simulation method is considered. It is realized by means of the grid's method of uniform probing of the space of parameters called LP_{τ} -search with averaging. The statements of optimization problems, discussion of the algorithms for solving them and application of methods of pattern recognition theory for evaluation the efficiency region of dynamic stochastic systems are involved in this chapter. The efficiency region is defined as the region in the space of parameters where the system quality indices are better than in other regions. Pattern recognition methods allow accelerate the evaluation of efficiency regions by ensuring better quality of processing the results of simulation experiments.

Chapter 4 – Solomonoff universal induction based on Algorithmic Probability (ALP) can be considered as the general theoretical basis for machine learning and, in particular, pattern recognition. However, its practical application encounters very difficult problems. One of them is incomputability caused by usage of the Turing-complete solution space. The Minimum Description Length (MDL) and the Minimum Message Length (MML) principles can be considered as simplified derivations of ALP applied to Turing-incomplete solution spaces. The MDL and MML principles have been successfully used to overcome overlearning and to enhance different pattern

recognition methods including construction of nonlinear discrimination functions, support vector machines, mixture models, and others.

However, restriction of the solution space is not the only simplification in the MDL/MML approaches. All possible models of data are used to calculate ALP, while the only one best model is selected on the base of the MDL/MML principle. In this chapter, the possibility to utilize Turing-incomplete version of ALP in the practical tasks of pattern recognition is considered. This gives theoretically and experimentally grounded approach to use “overcomplex” models (similar to compositions of classifiers or mixtures of experts) without the risk of overlearning, but with better recognition rates than that of minimum description length models.

It is impossible to sum over all models even in Turing-incomplete model spaces. Thus, it is necessary to select some number of models, which should be taken into account. These models can correspond to different extrema of the MDL criterion. For example, if models can have different number of parameters than the best model for each number of parameters can be taken into consideration. Models constructed on some subsets of a training set can be used for calculating ALP in the case of families of models with fixed number of parameters.

Some concrete applications of the ALP-based approach are considered. Its difference from finite mixtures and possible connection with boosting are discussed.

Chapter 5 – Image moments have been established in the area of pattern recognition and classification, since they can represent image content very effectively. One of the first moment family, and probably one of the most used, is the geometric one. However, a large variety of moments have been introduced so far. Orthogonal continuous moments like Zernike, Fourier-Mellin and pseudo Zernike or orthogonal discrete moments like Tchebichef, Krawtchouk and dual Hahn are only some of the most widespread families. Despite this variety in moment families, their vast majority has been applied in pattern recognition or classification problems where only gray images are considered. Only lately scientists try to address the issue of calculating moments for color images. The traditional way to apply moments to such images is either to use a color reduction method, with the known consequences of losing information, or to convert the color model from RGB to HSV in order to use only the H channel. Another perspective is to represent each color pixel as a 3-element vector. The resulted image can be used in order to compute color moments. This method results in to calculate 3-element vectorized moments, where each element is in relation with the corresponding

channel but not with the other elements. Quaternion moments, which have been lately attracting the interest of scientific community, address this problem elegantly. By representing each color pixel as a quaternion and using simple quaternion algebra, the resulted quaternion moments are directly connected to the image color space. In this chapter the basic theory as well as a comparison of the aforementioned two types of color moments is presented. The experimental analysis includes, except of image reconstruction and computation time cases, indicative classification examples in noise free and noisy conditions.

Chapter 6 – Recently, invariant correlation digital systems to position, rotation, scale and illumination are utilized in the pattern recognition field. Such invariants are made of by the Fourier and Fourier-Mellin transforms in conjunction with linear or nonlinear filters (k-law). In this work a new digital system invariant to position, rotation and illumination based on Fourier transform, Bessel masks, one-dimensional signatures and linear correlations are presented. Using one-dimensional signatures instead of diffraction patterns or vectorial signatures of the images reduces the computational time considerably, achieving a step toward the ultimate goal, which is to develop a simple digital system that accomplishes recognition in real time at a low cost. To achieve the invariant to translation the modulus of the Fourier transform of the image is taken. And, using a Bessel binary mask of concentric rings the invariant to rotation is obtained. The discrimination between objects is done by a linear correlation of the one-dimensional signatures assigned to each image and the target, in this manner the computational cost is reduced also. The images classification range are determined by the Fisher transformation statistic theory. The digital system was tested using a reference image database of 21 fossil diatoms images of gray-scale and 307×307 pixel. The system has a confidence level of 95.4% or greater in the classification of the 7,560 problem images using the same illumination. Then, those problem images were altered with eight different illuminations and the system also identifies the 60,480 images with a confidence level of 95.4% or greater.

Chapter 1

PATTERN RECOGNITION APPLIED TO SPECTROSCOPY: CONVENTIONAL METHODS AND FUTURE DIRECTIONS

***Ana Paula Craig^{1,3}, Adriana S. Franca^{2*}
and Joseph Irudayaraj³***

¹PPGCA Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brasil

²DEMEC, Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brasil

³Purdue University, West Lafayette, Indiana, US

ABSTRACT

Spectroscopic techniques have gained importance in a wide range of fields because of their appeal as rapid, reliable and nondestructive analysis, in most cases requiring minimum sample pre-treatment. These features make spectroscopy techniques suitable for routine analysis in on-line mode processing facilities. However, due to the high complexity of spectral data, which contain a large number of variables, multivariate statistical analysis is required to recognize patterns from samples. The multivariate nature of these methods makes evaluation of the robustness a much more complex task in comparison to classical ruggedness testing, as applied in univariate methods. In this review, unsupervised and

* E-mail: adriana@demec.ufmg.br. Tel: +55-31-34093512. Fax: +55-31-34433783.

supervised algorithms conventionally used for qualitative and quantitative analysis are explored. Among them, artificial neural networks, hierarchical clustering, linear regression extensions and principal component analysis are highlighted. Following the recent breakthrough in powerful and fast growing spectroscopic technologies, such as hyperspectral imaging, new challenges in pattern recognition are emerging. Thus, we present an overview of the new and promising developments in pattern recognition methods for complex spectral data, including support vector machines and penalized regression methods for robust variable selection.

1. INTRODUCTION

Spectroscopy is basically the experimental subject concerned with the absorption, emission or scattering of electromagnetic radiation by atoms or molecules. The electromagnetic radiation covers a wide wavelength range, from radio waves to γ -rays and the atoms or molecules may be in the gas, liquid or solid phase or, of great importance in surface chemistry, adsorbed on a solid surface (Hollas 2004). The fundamental measurement obtained in spectroscopy is a spectrum, which is a plot of measured electromagnetic radiation intensity versus some property of light (ie. wavelength, wavenumber) (Smith 2002). Among other techniques, optical spectroscopy has gained importance due to its extraordinary sensitivity, speed, and versatility, being widely applied in a variety of fields, including agriculture, ecology, geology, medicine, meteorology, and so on. Thereby, in this book chapter our main focus is on pattern recognition methods applied to optical spectroscopy, which includes fluorescence, infrared, Raman and UV/Vis.

Pattern recognition algorithms applied to spectral data are mainly based on chemometrics, given the large amount of information provided by each single spectrum. Chemometrics is the science of using statistical and mathematical methods to improve chemical measurements processes and to extract more useful information from chemical and physical data. These methods involve the application of mathematical pretreatments and classification and regression methods. Mathematical pretreatments are used in order to reduce the influence of redundant information, enhance information that is searched and compensate changes in experimental conditions. Due to the susceptibility of some spectroscopy techniques to instrumental instabilities and environment conditions, the use of these treatments is particularly important in the development of calibration models with minimal errors. The pretreatments

methods most commonly used are: filtering (e.g. baseline corrections, derivatives, smoothing), normalizations and scaling and centering. Given that this is not aim of the present review, this topic will not be explored. Nonetheless, reviews on mathematical pretreatments applied to spectral data are available in the literature (Workman Jr 2001; Rinnan, van den Berg and Engelsen 2009; Lasch 2012).

Classification methods can be divided into unsupervised and supervised. The first one aims at the classification of unknown observations into groups, according to similarity correlations. Unsupervised methods can be used when there is no known information of the dataset evaluated, or as a preliminary evaluation of the data information. For example, Principal Components Analysis (PCA) can be employed to evaluate if the studied samples can be discriminated by their spectra profile. If so, the analyst is encouraged to perform more experiments and get enough data to construct a regression model for quantitative analysis.

The supervised methods aim to create classification models based on a training set of data containing observations whose category is known. These models are sequentially used to identify which category new observations from a validation set belong, on the basis of their explanatory variables or features.

Once the classification of samples has been achieved, it can be useful to determine more precisely to what extend samples are different (Roggo et al. 2007).

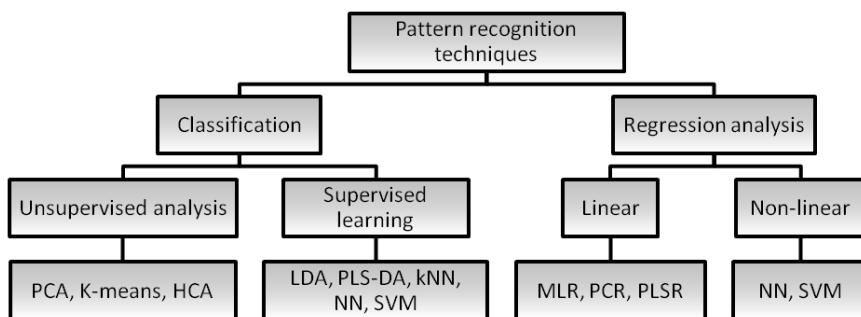


Figure 1. Pattern recognition techniques used for modeling spectral data. PCA = principal components analysis, HCA = hierarchical clustering analysis, LDA = linear discriminant analysis, PLS-DA = partial least squares discriminant analysis, kNN = k -nearest neighbor, NN = neural networks, MLR = multiple linear regression, PCR = principal components regression, PLSR = partial least squares regression, SVM = support vector machines.

The multivariate regression analysis consist of modeling a relationship between a desired physical, chemical or biological attribute, which represent independent variables, of an object, and its spectrum response, or dependent variables. This way, a regression model describes and estimates how the spectra response varies when any attribute of the sample changes. Figure 1 summarizes the conventional multivariate techniques used for spectral data analysis.

This review aims to provide a comprehensive and practical overview of pattern recognition techniques most usually applied to spectral data. Section 2 presents a summarized overview of the basic concepts associated with spectroscopy. On section 3, conventional unsupervised and supervised classification methods are covered. Sequentially, section 4 presents the conventional regression analysis. Following these sections, a brief review on the new and promising developments in pattern recognition methods for complex spectral data, such as hyperspectral data, is presented.

2. BASIC CONCEPTS OF SPECTROSCOPY

Spectroscopy is basically an experimental subject concerned with the absorption, emission or scattering of electromagnetic radiation by atoms or molecules, that may be in gas, liquid, solid phase or, of great importance in surface chemistry, adsorbed on a solid surface. As shown in Figure 2, the electromagnetic radiation covers a wide wavelength range, from low-energy radio waves to high-energy γ -rays. When exposed to radiation, many processes may occur in an atom or molecule. A molecule may undergo rotational, vibrational, electronic or ionization processes, in order of increasing energy. A molecule may also scatter light in a Raman process. An atom may undergo only an electronic transition or ionization since it has no rotational or vibrational degrees of freedom. Nuclear magnetic resonance (NMR) and electron spin resonance (ESR) processes involve transitions between nuclear spin and electron spin states, respectively (Hollas 2004). However, the focus of this review will be restricted to optical spectroscopy, which includes fluorescence, vibrational and UV/Vis spectroscopy.

Fluorescence and time-resolved fluorescence spectroscopy are considered to be primarily research tools in biochemistry and biophysics, and dominant methodologies in biotechnology, flow cytometry, medical diagnostics, DNA sequencing, forensics, and genetic analysis.

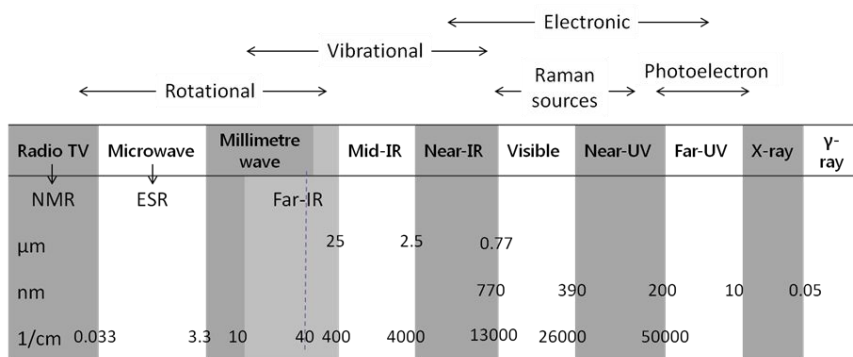


Figure 2. Regions of the electromagnetic spectrum and processes that may occur in an atom or molecule exposed to the radiation. IR = infrared, UV = ultra-violet, NMR = nuclear magnetic resonance, and ESR = electron spin resonance.

The methodologies are highly sensitive, and there is no longer the need for the expense and difficulties of handling radioactive tracers for most biochemical measurements. There has been a dramatic growth in the use of fluorescence for cellular and molecular imaging, which can reveal the localization and measurements of intracellular molecules, sometimes at the level of single-molecule detection. The phenomenon of luminescence is based on the emission of light from any substance, and occurs from electronically excited states. Luminescence is formally divided into two categories: fluorescence and phosphorescence, depending on the nature of the excited state. In excited singlet states, the electron in the excited orbital is paired to the second electron in the ground-state orbital. Consequently, the return to the ground state is spin allowed and occurs rapidly by emission of a photon. The emission rates of fluorescence are typically 10^8 s^{-1} , so that a typical fluorescence lifetime is near 10 ns. A fluorescence spectrum is usually a plot of fluorescence intensity *versus* wavelength (nm) or wavenumber (cm^{-1}). Phosphorescence, by contrast, is the emission of light from triplet excited states, in which the electron in the excited orbital has the same spin orientation as the ground-state electron. The transitions to the ground state are forbidden and the emission rates are slow (10^3 to 10^0 s^{-1}), so that phosphorescence lifetimes are typically milliseconds to seconds (Lakowicz 2009).

Vibrational spectroscopy, both infrared (IR) absorption and Raman scattering, has gained importance in many fields because of its appeal in rapid, reliable and nondestructive analysis, in most cases, requiring minimum sample pretreatment. The techniques provide structural and chemical information of

molecules based on their vibrational transitions. In infrared spectroscopy the sample is radiated with infrared light. Different chemical bonds absorb at different wavelengths depending on the atoms connected, the surrounding molecules, and the type of vibration the absorbance gives rise to (Narlikar and Fu 2010). On the other hand, Raman spectroscopy is based on irradiating a sample with a monochromatic visible or near infrared light from a laser. This brings the vibrational energy levels in the molecule to a short-lived, high-energy collision state. Most of the molecules relax back to the low energy original state, whereby a photon of the same wavelength as the exciting light is emitted (Rayleigh scattering). Only a very small percentage of the excited molecules relax back to a vibrationally excited state, hence the emitted photons have a lower frequency (Stokes Raman scattering). The difference between the frequency of the laser and that of the scattered photon is called Raman shift. The Raman shift corresponds to the frequency of fundamental infrared absorbance band of the bond. Because of the small percentage of molecules that use this relaxing pathway, Raman scattering is always of very low intensity and its investigation requires high-quality instrumentation (Thygesen et al. 2003; Hof 2005). Substantial advances in Raman spectroscopy led to new analytical techniques, such as confocal micro-Raman, that allows spatially resolved investigation of the chemical composition of heterogeneous foods, making possible the study of areas down to approximately $1 \times 1 \mu\text{m}$, focusing on different planes below the sample surface, and surface-enhanced Raman spectroscopy (SERS). The latter enhances vibrational absorbance of molecules adsorbed on or in the vicinity of metal particles and/or surfaces. The primary and dominant effects in determining the spectral enhancement characteristics of the analyte molecule are the surface plasmon resonance of the substrate and its variations, which are caused by surface roughness (Narlikar 2010). This mechanism enhancement can range between 4 to 11 orders of magnitude (Vlckova et al. 2007) resulting in new insights in biochemistry and molecular biology (Hudson and Chumanov 2009; Sun and Irudayaraj 2009) and application in material sciences, food safety, drugs, explosives, and environmental pollutants (Das and Agrawal 2011).

Although infrared and Raman spectroscopy probe molecular vibrations, they do not provide exactly the same information. Whereas IR spectroscopy detects vibrations due to electrical dipole moment changes, Raman spectroscopy is based on the detection of vibrations due to changes in polarization.

This implies that bonds that connect two identical or nearly identical parts of a molecule, for example, the C=C bond, tend to be more active than a

weakly polarizable bond, such as the OH bond. For this reason, water is practically invisible in Raman spectroscopy, but it dominates the IR spectrum (Thygesen et al. 2003).

Regarding UV-Vis spectroscopy, the transitions that result in the absorption of electromagnetic radiation are transitions between electronic energy levels.

Thus, as a molecule absorbs energy, an electron is promoted from an occupied orbital to an unoccupied orbital of greater potential energy. The UV-Vis spectrum is generally recorded as a plot of absorbance versus wavelength, but it is customary to then replot the data with either molar absorptivity (ϵ) or $\log \epsilon$ plotted instead of absorbance. Most organic molecules and functional groups are transparent in the regions of ultraviolet (UV) and visible (Vis), consequently, absorption spectroscopy is of limited utility in this range of the electromagnetic spectrum. However, in combination with another techniques, such as infrared and nuclear magnetic resonance (NMR) spectroscopy, UV-Vis can provide valuable information (Pavia 2009). Figure 3 presents an energy level diagram showing the states involved in the techniques briefly described in this section.

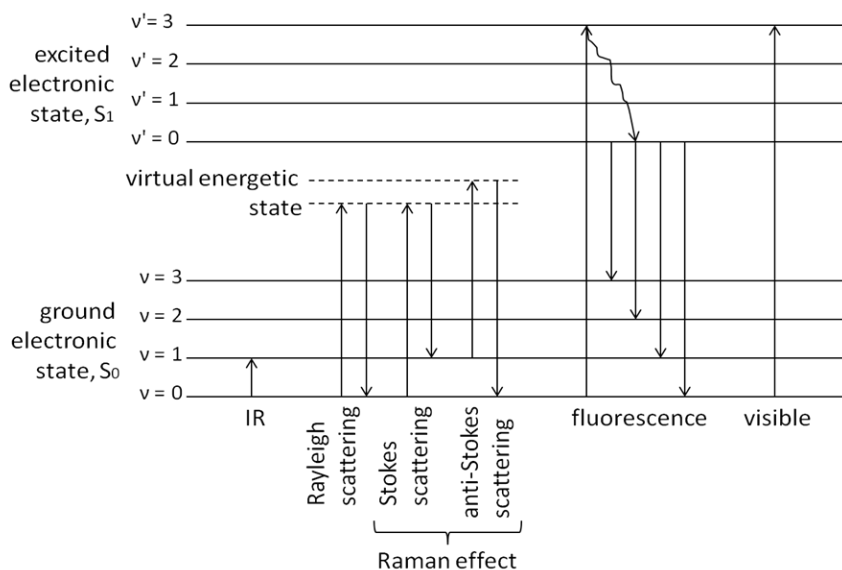


Figure 3. Energy level diagram showing the states involved in IR and visible absorption, Raman scattering and fluorescence emission.

3. CLASSIFICATION TECHNIQUES

Classification aims to group samples together according to their spectra. The techniques can be divided into unsupervised and supervised ones. In unsupervised techniques, samples are classified based solely on their spectra, without any prior knowledge about the data. For the second class, a prior knowledge of the samples is required, i.e. the category membership of samples. Classification models are developed based on a training set of samples with known categories, and then applied to a new set of samples, or a validation set of samples, for evaluation of model performance (Massart 1988). The training and the validation sets must be independent. In this section, the most common classification techniques applied to spectral data will be briefly presented.

3.1. Unsupervised Analysis

3.1.1. Principal Component Analysis (PCA)

PCA is one of the simplest schemes to reduce the dimensionality of data, with successful application in many fields. The analysis is concerned with explaining the variance-covariance structure of a set of variables through a few linear combinations of these variables, providing data reduction and interpretation (Johnson and Wichern 2007). In other words, PCA involves rotating and transforming the original p axes, each representing an original variable, into new axes. This transformation is performed in a way so that the new axes lie along the directions of maximum variance of the data with the constraint that the axes are orthogonal, i.e. the new variables are uncorrelated (Adams 1995). Although p components are required to reproduce the total system variability, often much of this variability and information can be accounted for by a small number of k principal components (PCs). This way, the original data set, consisting of n measurements on p variables, is reduced to a data set consisting of n measurements on k PCs (Johnson and Wichern 2007).

Traditionally, PCA starts as the solution to an eigenvalue equation of the input covariance matrix, C_{xx} (Arenas-García and Petersen 2009):

$$C_{xx}u_i = \lambda_i u_i, \quad (\text{Eq. 1})$$

where the eigenvalues $\{\lambda_i\}_{i=1}^{n_p}$, with $\lambda \geq \lambda_{i+1}$ are the n_p largest eigenvalues of C_{xx} , and $\{u_i\}_{i=1}^{n_p}$ their corresponding eigenvectors, which define the projection vectors.

The PCA algorithm is implemented by extracting the projection vectors one by one, using a sequential method consisting of the following steps (Arenas-García and Petersen 2009):

- Step 1. The leading vector of the covariance matrix and its corresponding eigenvalue are extracted. Different methods can be used to extract the largest eigenvector of a symmetric matrix.
- Step 2. The covariance matrix is deflated to remove the eigenvector obtained in the first step:

$$C_{xx} \leftarrow C_{xx} - \lambda_i u_i u_i^T \quad (\text{Eq. 2})$$

This deflation process amounts to projecting the data matrix onto the orthogonal complement of the direction given by u_i , and can also be expressed as (Arenas-García and Brandt Petersen 2009):

$$\tilde{X} \leftarrow \tilde{X}[I - u_i u_i^T] \quad (\text{Eq. 3})$$

The power of PCA is in revealing relationships based on similarity and difference between objects or samples that were not previously suspected. Thereby PCA allows interpretations, in chemical or physicochemical terms, that would not ordinarily result. In addition, this analysis frequently serves as an intermediate step in other data analysis techniques, e.g. PCs may serve as inputs to linear discrimination analysis (LDA), multiple regression or cluster analysis (Adams 1995; Johnson and Wichern 2007). For example, PCA was recently employed for discrimination between pure coffee, corn and coffee husks, since the later are commonly employed for coffee adulteration (Reis et al., 2013a). The resulting scatter plots (first vs. second PC component) based on FTIR data for raw and processed spectra are shown in Figure 4. Notice that pure and adulterated coffees are clearly separated into two groups, and that group discrimination is affected by spectra pretreatment. Other recent examples of PCA analysis applied to spectroscopic data can be found in the literature, including applications in FTIR (Arzberger and Lachenmeier 2008; Gurdeniz and Ozen 2009; Rubio-Diaz et al. 2010; Craig et al., 2011a, 2012ab; Reis et al., 2013b), NIRS (Said et al. 2011; Santos et al. 2012; Mireei and

Sadeghi 2013) and Raman (Abbas et al. 2009; Chen and Han 2011; El-Abassy et al., 2011; Özbalci et al. 2013).

Despite PCA's scheme simplicity, applications are not restricted to simple cases. Pirro et al. (2012) applied for the first time an interactive PCA-brushing approach to hyperspectral desorption electrospray ionization mass spectrometry (DESI-MS) imaging data.

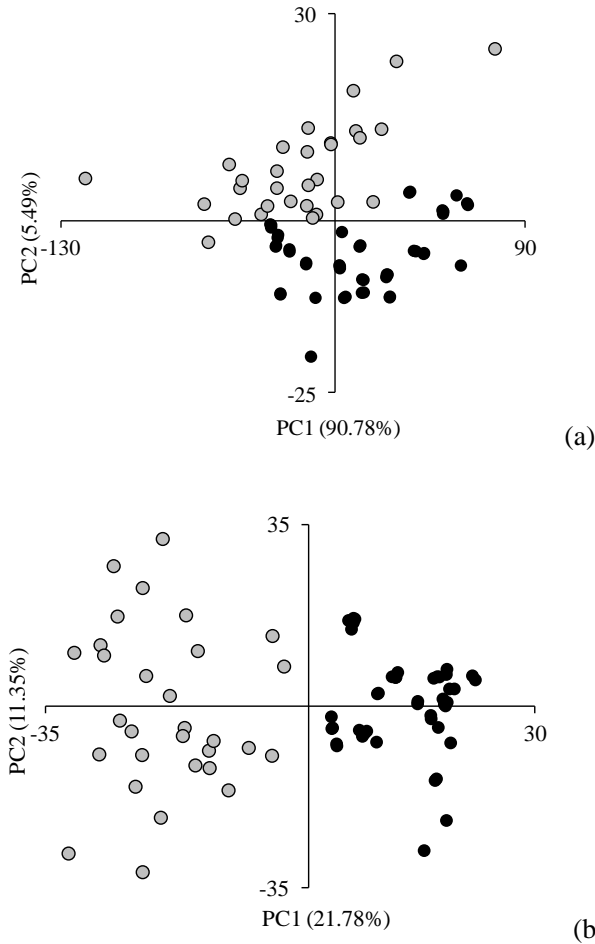


Figure 4. Scatter plots (PC1 vs. PC2) based on diffuse reflectance spectra of pure (●) and adulterated (●) roasted coffee samples (a) based on raw spectra; (b) based on spectra first derivatives (Reis et al., 2003a).

It exploited the chemical information provided by mass spectrometry, allowing the characterization of more than 40 samples of normal and cancerous kidney, prostate, germ and bladder tissues. The interactive brushing procedure was performed in order to understand the relationships between the PC space and the image space, connecting chemical and spatial information. In particular, the score plot allowed a visual inspection of the pixel distribution in PC space. In the score plot, it was possible to visualize groupings that indicated similarities among pixels, on the basis of the information derived from the mass spectra, and which could be associated with the particular characteristics of the samples analyzed. Using this procedure, pixels with similar chemical profiles could be manually selected from the score plot in order to identify correspondences between the groups of points in the PC score plot and particular regions of the DESI-MS image. Sequentially, an examination of both the loading plot and the score plot allowed chemical characterization of the highlighted region of the image to be achieved, revealing which m/z peaks are the most important in defining the pixels under consideration. The results obtained indicate that the proposed methodology may be valuable in cancer diagnosis and complementary to the traditional histopathological examination (Pirro et al. 2012).

3.1.2. Clustering Algorithms

The concept of similarity between objects provides the richness and variety of the wide range of techniques available for cluster analysis, as hierarchical clustering analysis (HCA) and k -means. The general scheme is based on the conversion of the data into some corresponding set of similarity, or dissimilarity, measures between each sample and the clustering of similar objects while the separation between different clusters is maximized. This similarity is measured as the distance between objects in a multidimensional space defined by the axes, each of which corresponds to a variable. Numerous distance metrics have been proposed and applied in the scientific literature, but the most commonly referenced are the Euclidean distance (ED) and Mahalanobis distance (MD) (Adams 1995). Both distances can be calculated in the original variable space and in the principal component (PC) space (De Maesschalck, Jouan-Rimbaud and Massart 2000). The ED is defined by:

$$d_{AB} = [\sum_j (x_{1j} - x_{2j})^2]^{1/2} \quad (\text{Eq. 4})$$

where x_{1j} is the value of the j 'th variable measured in the i 'th object.

The MD, a weighted distance measure, is defined by:

$$d_{AB} = [(a - b)^T Cov^{-1}(a - b)]^{1/2} \quad (\text{Eq. 5})$$

where Cov is the full variance-covariance matrix for the original data.

The MD occurs not only in cluster analysis but it is also used for detecting outliers during calibration or prediction, detecting extrapolation of the model and in discriminant analysis (Adams 1995; Mark and Workman Jr 2007ab). In the original variable space, the MD takes into account the correlation in the data, since it is calculated using the inverse of the variance–covariance matrix of the data set of interest. However, the computation of the variance–covariance matrix in data containing large number of variables can cause problems. When the data is measured over a large number of variables, it can contain much redundant or correlated information. This so-called multicollinearity leads to a singular variance–covariance matrix that cannot be inverted. A second limitation for the calculation of the variance–covariance matrix is that the number of objects in the data set has to be larger than the number of variables. For these reasons, it is clear that in many cases, feature reduction is needed. This can be done by selecting a small number of meaningful variables or using latent variables (PCs), obtained after PCA, instead of the original variables. Since PCs are by definition orthogonal (uncorrelated), the MD does not need to correct for the covariance between the variables. Nevertheless, the way each of the residual PCs is weighted in the computation of the distance must be taken into account. A detailed discussion on this topic is presented by De Maesschalck and co-workers (2000).

When using HCA, the original data are separated into a few general classes, each of which is further subdivided into still smaller groups until finally the individual objects themselves remain. Such methods may be agglomerative or divisive. In agglomerative clustering, small groups, starting with individual samples, are fused to produce larger groups. In contrast, divisive clustering starts with a single cluster, containing all samples, which is successively divided into smaller partitions. Agglomerative methods, the most commonly used, involve the calculation of a four step algorithm (Adams 1995):

- Step 1. Calculate the between-object distance matrix. The different between-group distance measures can be defined in terms of the general formula presented as follows:

$$d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}| \quad (\text{Eq. 6})$$

where d_{ij} is the distance between objects i and j and $d_{k(ij)}$ is the distance between group k and a new group (ij) formed by the fusion of groups i and j .

- Step 2. Find the smallest elements in the distance matrix and join the corresponding objects into a single cluster.
- Step 3. Calculate a new distance matrix, taking into account that clusters produced in the second step will have formed new objects and taken the place of original data points.
- Step 4. Return to Step 2 or stop if the final two clusters have been fused into the final, single cluster.

The specific between-group metric to be used varies accordingly to different values of the coefficients α_i , α_j , β , and γ . Table 1 lists the most common metrics and the corresponding values for α_i , α_j , β , and γ (Adams 1995).

Table 1. Common distance metrics used in cluster analysis

Metrics	Coefficients			
	α_i	α_j	β	γ
Nearest neighbor (single linkage)	0.5	0.5	0	-0.5
Further neighbor (complete linkage)	0.5	0.5	0	0.5
Centroid	$\frac{n_i}{n_i + n_j}$	$\frac{n_j}{n_i + n_j}$	$-\alpha_i \alpha_j$	0
Ward's method	$\frac{n_i + n_k}{n_i + n_j + n_k}$	$\frac{n_j + n_k}{n_i + n_j + n_k}$	$-\frac{n_k}{n_i + n_j + n_k}$	0

The number of objects in any cluster i is n_i .

An example of HCA application to separation of coffees of different qualities is shown in Figure 5. It can be noticed that four major clusters are present, in reference to black, dark sour, non-defective and immature/light sour coffees.

Overall clustering seems to be related to sample surface color, with the darker samples (black and dark sour) being completely separated from the remaining lighter samples (non-defective, light sour and immature).

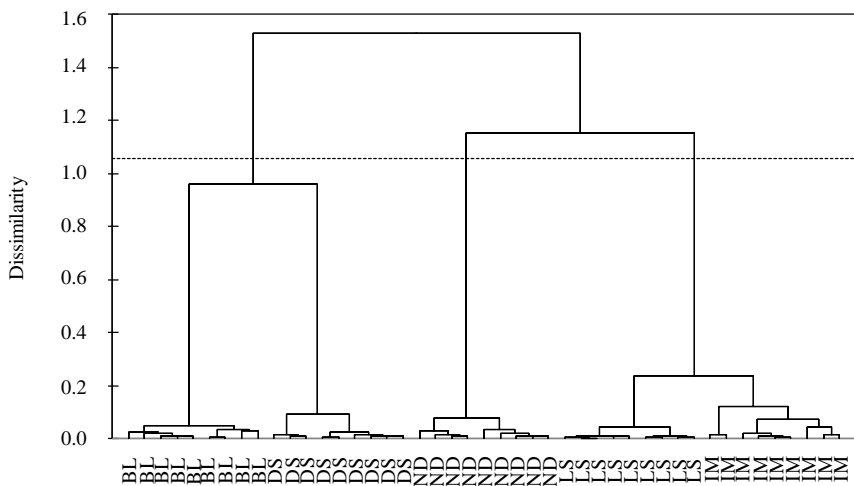


Figure 5. Hierarqic cluster analysis (HCA) of normalized FTIR spectra of green coffes (ND – non-defective; SO- sour; LS – light sour; DS- dark sour; IM – immature; BL – black) (Craig et al., 2011b).

Other examples of the application of HCA to spectral data include the classification of edible and lampante virgin olive oil using synchronous fluorescence and total luminescence spectroscopy (Poulli, Mousdis and Georgiou 2005), discrimination between different strains of lactic acid bacteria with UV-resonance Raman spectroscopy (Gaus et al. 2006) and design of an automated approach to distinguish between genuine and counterfeit tablets of Viagra® with Raman spectroscopy (De Veij et al. 2008). In the study by Shanmukh et al. (2008) HCA was used to classify respiratory syncytial virus (RSV) strains by surface-enhanced Raman spectroscopy (SERS). The HCA dendrogram was generated using *K*-means classification based on the distance between groups of scores using the first seven principal components of the data set. HCA demonstrated excellent classification results for the A/Long and B1 strains, and also to distinguish an A2 strain-related G gene mutant virus (ΔG) from the A2 strain.

Unlike HCA, the *K*-means is a non-hierarchical clustering algorithm, where each observation is assigned to the cluster having the nearest centroid (mean). The main objective of this method is to partition the m objects, each one characterized by n variables, into K clusters, so that the square of the within-cluster sum of distances is minimized. Being an optimization-based technique, the number of possible solutions cannot be predicted and the best

possible partitioning of the objects cannot be achieved. It means that the method finds a local optimum, defined as being a classification in which no movement of an observation from one cluster to another will reduce the within-cluster sum of squares (Adams 1995; Johnson and Wichern 2007). The process is composed, in a simplified way, of the three following steps (Johnson and Wichern 2007):

- Step 1. Partition the items into K initial clusters.
- Step 2. Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest. Euclidean distance is usually used with either standardized or unstandardized observations. Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.
- Repeat Step 2 until no more reassignments take place.

3.2. Supervised Learning

3.2.1. Linear Discriminant Analysis (LDA)

LDA is a parametric and linear classifier. The method focuses on finding optimal boundaries between classes. As PCA, LDA is a feature reduction method, however, while PCA selects a direction that retains maximal structure in a lower direction among the data, LDA selects the direction that achieves a maximum separation among the different classes (Sharaf, Illman and Kowalski 1986). For establishing a reliable LDA classifier model, the number of objects required needs to be higher than the number of variables. Hence, for spectral data analysis, variables reduction is usually necessary (Wang and Mizaikoff 2008). A common practice is using PCs obtained from PCA as inputs for LDA.

The algorithm is based on the assumption that the classes have multivariate normal distributions. Their means $\hat{\mu}_j$ and the pooled covariance matrix $\hat{\Sigma}$ can easily be estimated from the training set (Schmid et al. 2009):

$$\hat{\mu}_j = \sum_{g_i=j} x_i / N_j \quad (\text{Eq. 7})$$

$$\hat{\Sigma} = \sum_{j=1}^J \sum_{g_i=j} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T / (N - J) \quad (\text{Eq. 8})$$

The notation $\sum_{g_i=j}$ means summing over all observations i belonging to class j . A new observation x is assigned to class j by maximizing the discriminant function $d_j(x)$ (Eq. 9). The first term of $d_j(x)$ refers to the squared Mahalanobis distance $D(x, \mu_j)$ between observation x and the class centroid μ_j of class j . The second term $\ln \Pi_j$ corresponds to the prior probability of class j . The prior Π_j are estimated by $\Pi_j = N_j/N$. Maximizing $d_j(x)$ is equivalent to maximizing the *a posteriori* probability (Eq. 10), which is the probability that a new object x belongs to class j . Given the class-conditional density $f_j(x)$, the *a posteriori* probability $\Pr(G = j|X = x)$ can be calculated according to the Bayes theorem (Eq. 11).

$$d_j(x) = -D(x, \mu_j)/2 + \ln \Pi_j \quad (\text{Eq. 9})$$

$$f_j(x) = \Pr(X = x) | G = j) = |2\pi\Sigma|^{-1/2} e^{-D(x, \mu_j)/2} \quad (\text{Eq. 10})$$

$$\Pr(G = j|X = x) = \frac{f_j(x)\Pi_j}{\sum_{j=1}^J f_j(x)\Pi_j} \quad (\text{Eq. 11})$$

Although LDA has often been shown to produce the best classification results, it still has numerical limitations. In particular, for large data sets with too many correlated predictors, LDA uses too many parameters that are estimated with a high variance, which leads to a lack of interpretability (Le, Boitard and Besse 2011). The regularization and introduction of sparsity in LDA to obtain a parsimonious model was then proposed and discussed by Shao et al. (2011) and Clemmensen et al. (2011).

As an illustrative example, consider the application of LDA for discrimination between roasted coffee, corn and coffee husks shown in Figure 6 (Reis et al., 2013c), since the later are commonly used for adulteration of roasted and ground coffee. Notice that coffee and adulterants are clearly separated. The developed models presented 100% accuracy in terms of validation and prediction.

Other examples include the discrimination among 10 different edible oils and fats by FTIR, FT-NIR and FT-Raman spectroscopy (Yang, Irudayaraj and Paradkar 2005), classification of malignant gliomas by infrared spectroscopy (Krafft et al. 2006) and detection of adulteration in gasoline by FTIR (Pereira et al. 2006). Dochow et al. (2011) applied optical traps in combination with Raman spectroscopy to acquire spectra of single cells of erythrocytes,

leukocytes, acute myeloid leukaemia cells (OCI-AML3), and breast tumour cells BT-20 and MCF-7, in microfluidic glass channels.

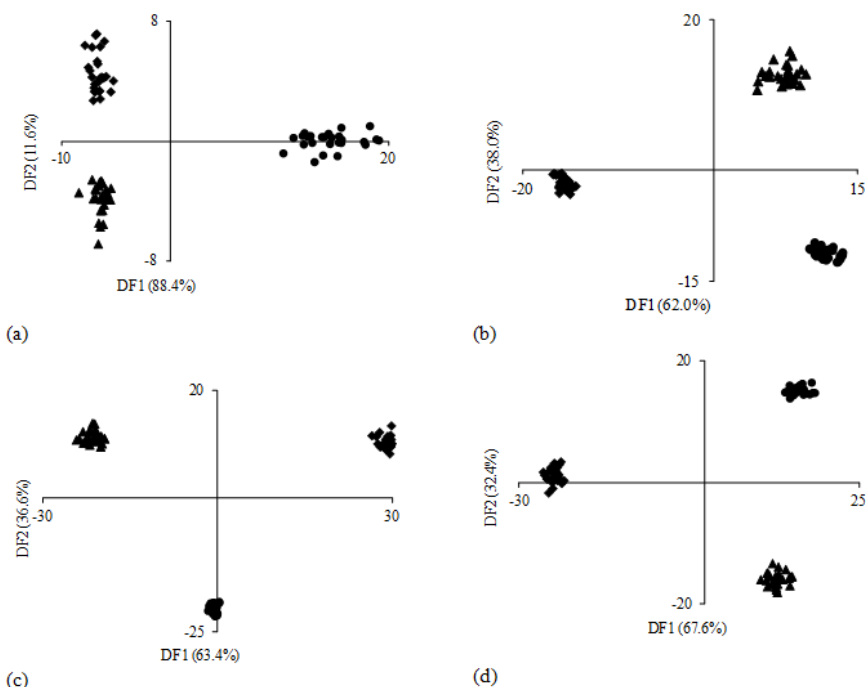


Figure 6. Scores of the discriminant functions provided by the LDA models of FTIR spectra (3100 - 700 cm^{-1}) after the following pretreatment steps: (a) no treatment; (b) normalization; (c) two and (d) three point baseline correction (● coffee; ▲ coffee husks; ◆ corn) (Reis et al., 2013c).

The classification models based on LDA achieved accuracies that were comparable with previous Raman experiments of dried cells and fixed cells in a Petri dish (Dochow et al. 2011).

Li et al. (2011) developed a novel method for the non-destructive age determination of blood stains. The method was based on visible reflectance spectroscopy of the hemoglobin component, using a microspectrophotometer, and LDA was used to predict blood stain age. Excellent results of up to 99.2% correct classification rate were obtained when training and test spectra were taken from the same blood stain. Although accuracy was poorer when using separate blood stains, this technique shows promise for forensic applications. Nevertheless, further studies are required to investigate the effect of other

variables such as temperature, humidity, illumination, stain thickness and substrate.

3.2.2. Partial Least Squares Discriminant Analysis

Partial least squares discriminant analysis (PLS-DA), also a parametric and linear model, is one of the most applied classification methods for spectral data. Applications vary from classification models for metabonomic NMR data based (Beckwith-Hall et al. 2002; Gavaghan, Wilson, and Nicholson 2002; Holmes and Antti 2002; Pears et al. 2005; MacIntyre et al. 2010) to food authentication based on infrared and UV-Vis spectroscopy (Roussel et al. 2003; Gurdeniz and Ozen 2009; Liu et al. 2008; Cozzolino et al. 2011).

The algorithm works as follows. First, a partial least squares regression (PLSR) is performed from an indicator response matrix Y ($N \times J$), which contains the class indices for the objects in X , on the predictor matrix X ($N \times P$). The ij th element of Y is 1 if the i th observation falls in class j , 0 otherwise. A new observation x is classified by computing the fitted output vector \hat{y} for x on the basis of the PLS-DA model of the training set and classifying according to the largest component in \hat{y} , i.e. the component closest to one (Schmid et al. 2009). The latent variables are identified by the PLSR in the featured spaces that have maximal covariance within the predictor variables. The number of latent variables to be used is determined by cross-validation (Roggo et al. 2007).

There is a tight link between LDA and PLS-DA: it can be shown that for two-class problems LDA and PLS-DA are equivalent, but for multi-class problems, some authors have demonstrated that LDA overcomes PLS-DA in accuracy (Indahl et al. 1999; Hastie, Tibshirani, and Friedman 2009; Le, Boitard, and Besse 2011). Nonetheless, Chevallier et al. (2006) presented some strategies to increase the performance of PLS-DA, and Le, Boitard and Besse (2011) introduced sparsity in PLS-DA to obtain a parsimonious model. The obtained model, called sPLS-DA, often gave similar classification performance to competitive sparse LDA approaches in multiclass problems. In addition, the computational efficiency and the valuable graphical outputs also provided easier interpretation of the results, making sPLS-DA a great alternative to other types of variable selection techniques in a supervised classification framework. In the recent study by Pizarro et al. (2012), visible spectroscopy fingerprints were applied for the discrimination of Spanish extra virgin olive oils in accordance with geographical origin. The results reported showed that PLS-DA strategy led to better geographical discrimination results

than the LDA strategy especially when classification was performed on visible fingerprints and fused matrices, achieving, in this case, 100% of correct classification rate.

3.2.3. *k*NN (*k*-Nearest Neighbor)

In spite of its simplicity, the nonparametric *k*NN algorithm often yields excellent classification results for a broad variety of data sets. One example is the effective application of *k*NN in the classification of single cancer cell phases based on features extracted from cell nuclei using time-lapse fluorescence imaging (Chen, Zhou, and Wong 2006). As in the case of clustering algorithms (see section 3.1.2), proximity is measured by Euclidean distance. Each unknown sample on the validation set is classified according to the class to which belongs the majority of its *k* nearest neighbors in the training set (Schmid et al. 2009). The drawback of this method is that for large sized datasets, a large number of computations has to be performed (Markou and Singh 2003ab).

3.2.4. Neural Networks

The trend in computer science is moving toward simulation of the brain and thought processes through neural networks (NN), fuzzy logic (a metaphor to human approximation thinking), speech interactive networks, termed ASR (automatic speech recognition), and VR (virtual reality) (Workman et al. 1996). It is a fact that humans can perform many tasks even better than machines, such as recognition of images, voices and smells, and tasks involving science, technology and business. Then, it has been of great interest to understand how we do so. NN was traditionally used to refer to a network or circuit of biological neurons, in other words, to model and simulate the architecture and physiology of the brain. But in the context of this book chapter, this term is applied to describe the so called *artificial* NN, which have been developed by a community originally biologically motivated (although many NN methods are not) and has been used for pattern recognition (Ripley 2008).

An important advantage of NN is that a very small number of parameters need to be optimized for training networks and no *a priori* assumptions on the properties of data are made. These features made a number of different NN architectures be extensively used for novelty detection, which include multi-layer perceptrons (MLP), self-organizing maps, radial basis functions (RBFs), support vector machines (SVM), Hopfield networks, oscillatory networks, etc (Markou and Singh 2003ab). Among them, the MLP and radial basis functions

(RBFs) are the most widely used (Ripley 2008). The MLP is a network having several layers of adaptive weights, which allow for more general mappings in comparison to networks having a single layer of adaptive weights. In RBFs the activation of the hidden unit is determined by the distance between an input vector and prototype vector (Bishop and Hinton 1995). A review on NN methods for novelty detection is presented by Markou and Singh (2003a).

A relevant application of NN to Raman spectral data is presented by Gniadecka et al. (2004). In their study NN was able to diagnostic melanoma and basal cell carcinoma in tissues with sensitivity and specificity of 85% and 99%, and 97% and 98%, respectively. According to the authors, it is possible that Raman spectroscopy possessing the strength of the specificity of the biochemical information may reveal early signs of cancer transformation before any morphologic structure analysis is seen in histopathology. Moreover, the NN network presented had the advantages that the parameter optimization did not rely on cross-validation or tuning of parameters, and the important input features (Raman frequencies) could be estimated directly from the optimized NN. An interesting fact here is that the NN sensitivity map showed that the computer system had used identical spectral bands to those usually used for visual spectra differentiation. Recent applications include a number of studies where NN was successfully used to analyze laser induced breakdown spectral data. Among them, we can list the identification of materials selected to represent the sites analogues to Mars (Koujelev et al. 2010), classification of different polymer materials (Boueri et al. 2011), identification and discrimination of bacteria strains (Marcos-Martinez et al. 2011) and fast identification of archeological or paleontological biominerals (Vítková et al. 2012).

4. REGRESSION ANALYSIS

4.1. Linear Regression Analysis

In optical spectroscopy, the features of a spectrum (number, intensity and shape of bands) are directly related to the molecular structure of the sample in study. A spectrum is a physical property of a given compound or sample, as its *fingerprint*. Based on the Beer-Lambert law, the intensity of the spectral bands are proportional to the concentration of a specific compound (Eq. 12). Therefore, it is possible to perform quantitative analyzes using methods based on the intensity of the bands, or preferably integrated intensities (Hof 2005).

The common practice of relating spectroscopic response to concentration has been accomplished for a number of applications using Beer-Lambert law combined with *C*-matrix, *K*-matrix, multiple linear regression (MLR), principal components regression (PCR), and partial least-squares regression (PLSR) (Workman et al. 1996).

$$A = \varepsilon cl \quad (\text{Eq. 12})$$

where A is the measured absorbance, ε is the molar absorption coefficient, c the concentration of the absorbing compound, and l is the path length of light within the absorbing medium.

MLR is the oldest of the usually employed methods and it is becoming less and less used due to improvements in computation resources. This regression allows for the establishment of a link between a reduced number of wavelengths (or wavenumbers) and a given property of the samples. The prediction y_j of the search property can be described as shown below (Eq. 13). Each wavelength is studied one after another and correlated to the studied property. Wavelength selection is based on its predictive ability. The three modes of selection are: forward, backward, and stepwise. When the correlation reaches a value fixed by the operator, the corresponding wavelength is kept as a part of the calibration model. The model is then computed between this set of calibration wavelengths and the reference values of the studied property (Roggo et al. 2007).

$$y_j = b_0 + \sum_{i=1}^k b_i x_i + e_{ij} \quad (\text{Eq. 13})$$

where b_i is the computed coefficient, x_i the absorbance at each considered wavelength, and e_{ij} the error.

As MLR involves large numbers of independent variables, there is often extensive collinearity or correlation between these variables. This collinearity adds redundancy to the regression model, since more variables may be included in the model than is necessary for adequate predictive performance. To reduce collinearity, the regression coefficients should be orthogonal (Adams 1995). Thereby, a combination of PC's, initially calculated from the standard spectra by PCA, with concentration information in a linear regression can resolve this issue. The goal of Principal Components Regression (PCR) is simply relating the scores from a selected number of PCs from PCA to the concentrations using calibration coefficients.

Another important regression analysis is PLSR. One way of thinking of PLSR is that it forms “new x -variables” (LV estimates), t_a , from linear combinations of the old x ’s, and therefore uses these new t ’s as predictors of Y . PLSR is a generalization of MLR, but unlike the latter, it can analyze data with strongly collinear, noisy, and numerous X -variables, and also simultaneously model several response variables, Y (Wold, Sjostrom, and Eriksson 2001). Other advantage is that the algorithm is a one step process. The spectral and concentration information are included in the calculation of the factors and scores, and several Y ’s can be analyzed together, which provides the benefit of giving a simpler overall picture than one separate model for each Y -variable. Hence, when Y ’s are correlated, they should be analyzed together (Clark and Cramer 1993; Wold, Sjostrom, and Eriksson 2001; Smith 2002). In the case of spectral data, there are also numerous and correlated X -variables, that represent a substantial risk for “over-fitting”, i.e., getting a well fitting model with little or no predictive power. Then, the number of PLSR components must be carefully chosen, based on the predictive significance of each PLSR component, in order not to include components when that are non-significant (Mantanus et al. 2009; Wold, Sjostrom, and Eriksson 2001). This predictive significance can be reliably accessed by cross-validation, a standard procedure in PLSR discussed in the studies by Clark and Cramer (1993) and Wakeling and Morris (1993).

The power of PLSR applied to spectral data becomes evident when we take into account the large number of validated methods published. As an example, Mantanus et al. (2009) validated a method based on NIRS for determination of moisture content of pharmaceutical pellets with acceptable precision, trueness and accuracy. Subsequently, Botelho, Mendes, and Sena (2012) developed and validated a robust method for quality inspection control of mozzarella cheese. This study explored the state of art of how to estimate the figures of merit in multivariate calibration, such as trueness, precision, linearity, working range, selectivity, sensitivity, analytical sensitivity, ruggedness, bias, and residual prediction deviation. It is worth noting that the authors developed this method under real conditions of routine analysis in a laboratory of food quality inspection control, and the method was monitored for approximately one year through control charts. The establishment and implementation of spectroscopic methods in such situations present several advantages over current methods, e.g. low cost, simplified procedure, no need for reagents, and no chemical waste generation.

Other applications include the quantification of fructan concentration in grasses using NIRS (Shetty and Gislum 2011), quantification of relative

concentrations of native ribonuclease (RNase) A protein and RNase B glycoprotein within mixtures using Raman spectroscopy (Brewster, Ashton, and Goodacre 2011), prediction of carbon contents in soil using visible/near infrared diffuse reflectance spectroscopy and FTIR (Kamau-Rewe et al. 2011; Sarkhot et al. 2011; Haiqing et al. 2012) and prediction of sensory attributes values of pork by Raman spectroscopy (Wang, Lonergan, and Yu 2012).

4.2. Non-Linear Regression Analysis

The models presented in the previous section are based on the assumption of the existence of a linear relationship between the variables and the physical, chemical or biological attribute to be predicted. For a long time the contribution of nonlinearity to the error of spectroscopic calibrations was not generally recognized by the spectroscopic or the chemometric communities. Trying to improve calibration performance, much attention and concern was given to issues like random noise, choice of factors (for PCR and PLS) and wavelengths (for MLR), and investigations into the best data transform (Mark and Workman Jr 2007ab). Nowadays it is well known that in certain applications linearity may never be approximated, leading to a need for using non-linear models (Benoudjit et al. 2004). Noticeably non-linear regression methods that have been extensively used for spectral data analysis are based on NN and support vector machines (SVMs). The fundamentals of these methods are briefly discussed in sections 3.2.4 and 5.2.1.

5. FUTURE TRENDS

Following the recent breakthrough in powerful and fast growing spectroscopic technologies, new challenges in pattern recognition are emerging. In this section we present an overview of the new and promising developments in pattern recognition methods for dimensionality reduction and variable selection, hyperspectral imaging analysis, with special attention to support vector machines (SVMs).

5.1. Sparse Learning Dimensionality Reduction Algorithms

One of the primary focuses in pattern recognition applied to spectral data is finding a succinct and effective representation for original high dimensional samples. Due to the typical “ $p \gg n$ ” feature in spectral data, where n is the number of observations and p is the number of variables, and the fact that these variables are correlated, statistical analysis and results interpretation of spectral data is still challenging.

The conventional algorithms applied for spectroscopy data, e.g., PCA, LDA, are categorized into linear dimensionality reduction algorithms, assuming that samples are drawn from different Gaussians. The unsupervised PCA maximizes the mutual information between original high-dimensional Gaussian distributed samples and projects low-dimensional samples. Meanwhile, LDA finds a projection matrix that maximizes the trace of the between-class scatter matrix and minimizes the trace of the within-class scatter matrix in the projection subspace simultaneously. These algorithms produce a low dimensional subspace and each basis of the subspace is a linear combination of all the original bases (i.e. variables) used for high dimensional sample representation. Since each of the new bases is a linear combination of the original ones, it is reasonable to consider each new basis as the response of several variables, representing a problem in terms of variable selection and coefficients shrinkage (Zhou, Tao and Wu 2011).

Sparse learning dimensionality reduction algorithms, e.g. lasso (Tibshirani 1996; Tibshirani 2011) and elastic net (Zou and Hastie 2005), were developed not only to achieve dimensionality reduction, but also to reduce the number of explicitly used variables. In the last years these algorithms have become popular, according to Zhou et al. (2011) because:

- Sparsity can make the data more succinct and simpler, so the calculation of the low dimensional representation and the subsequent processing, e.g. classification and regression, becomes more efficient;
- Sparsity can control the weights of original variables and decrease the variance brought by possible over-fitting with the least increment of the bias; and
- Sparsity provides a good interpretation of a model, revealing an explicit relationship between the objective of the model and the given variables. This is important for understanding practical problems, especially when the number of variables is larger than the number of samples.

Due to these advantages, lasso and elastic net regression are frequently used in domains with very large datasets, such as genomics and web analysis (Zhu and Hastie 2004; Friedman, Hastie and Tibshirani 2010). Recent studies also demonstrate the applicability of these methods to spectroscopy data analysis, as discussed below.

Lasso regression is a popular penalized least squares method that imposes an L_1 -penalty on the regression coefficients. The L_1 -penalty corresponds to a Laplace prior, which expects many predictors to be close to zero and a small subset to be larger and nonzero. This way, lasso regression provides both continuous shrinkage and automatic variable selection simultaneously. The accuracy of Lasso regression was compared to PLSR and also ordinary least squares (OLS) for the reconstruction of glucose levels from 150 multisensor channels measured with dielectric spectroscopy and optical sensors, in a continuous glucose monitoring (CGM) sensor approach. Although OLS outperformed PLSR and Lasso in estimating glucose profiles in the training set, Lasso provided better generalization performances in predicting “unseen” data from the validation set. The prediction was improved because Lasso forced a sparse solution and estimated parameters with low absolute values, trading off decreased variance for increased bias. In addition, although PLSR provided good predictions, it was too sensitive to noisy channels, as noisy channels containing glucose information might be used to building the new latent variables. On the other hand, Lasso selected only original variables that were likely to be less sensitive to noise (Zanon et al. 2011).

In another study, Dyar and coworkers (2012) compared PLSR and Lasso regression techniques to determine the elemental composition of igneous and highly-metamorphosed rocks based on the spectra (intensity at each of 6144 spectrometer channels) obtained by a remote laser-induced breakdown spectrometer (LIBS). Despite the results of both techniques being comparable in terms of accuracy, the interpretability differed greatly in terms of fundamental understanding. While PLSR generated principal components projected into the original feature space of the spectra, resulting in 6144 correlation coefficients, Lasso required a much smaller number (<24) of non-zero correlation coefficients to determine the concentration of each of the rock elements. Thus, Lasso could directly provide an understanding of the underlying physical processes that gave rise to LIBS emissions by determining which coefficients could best represent concentration and which ones were causing matrix effects.

Although Lasso has shown success in many situations, it presents some limitations in the following scenarios: (a) in the $p > n$ case, Lasso selects at

most n variables before it saturates due to the nature of the convex optimization problem, and (b) if there is a group of variables among which the pairwise correlations are very high, Lasso tends to select only one variable from the group and does not care which one is selected. The regularization technique called Elastic net was proposed to fix these problems (Zou and Hastie 2005). Elastic net is a version of penalized least squares that combines both Ridge and Lasso regression. Ridge regression shrinks (toward zero) the least square coefficients, while Lasso not only shrinks the coefficients but also provides model selection. Unlike Lasso penalty, Ridge penalty (L_2 -penalty), drawn from a Gaussian distribution, is ideal if there are many predictors and all have non-zero coefficients. Therefore, in elastic net the penalty is a compromise between Ridge-regression penalty ($\alpha = 0$) and Lasso penalty ($\alpha = 1$) (Friedman, Hastie, and Tibshirani 2010).

Fu and coworkers (2011) proposed a multi-component spectral data analysis, called elastic net grouping variable selection combined with partial least squares regression (EN-PLSR) that can be seen as a two-step variable shrinkage. First elastic net eliminates uninformative variables. Second, the recursive leave-one-group-out strategy further shrinks the variables in terms of PLSR in the *root-mean-square error of cross-validation* (RMSECV) sense. The algorithm was applied to near infrared (NIR) spectroscopy data sets and provided competitive results with full-spectrum PLS regression method. Stephen et al. (2012) applied Elastic net to SERS spectra of a single bacterial strain grown on solid media culture with three different chromate levels. Elastic net allowed the classification and visualization of discrete points or wavelengths that discriminated environmentally induced cell surface composition. In order to predict anticancer drug sensitivity, Barretina et al. (2012) presented a collection of gene expression, chromosomal copy number and massively parallel sequencing data from 947 human cancer cell lines. When coupled with pharmacological profiles for 24 anticancer drugs across 479 of the cell lines, this collection allowed identification of genetic, lineage, and gene-expression-based predictor of drug sensitivity. Mutation information was obtained by using massively parallel sequencing of >1600 genes and by mass spectrometric genotyping. Elastic net was used to derive predictive models that explained the drug profiles based on genetic features of cell lines. The obtained predictors revealed both known and novel candidate biomarkers of response. Even within genetically defined sub-populations, or when agents were broadly active without clear genetic targets, elastic net modeling studies identified key predictors or mechanistic effectors of drug response.

5.2. Hyperspectral Analysis

In images, colors of pixels are usually represented as a combination of channels in the visible part of the electromagnetic spectrum, and the combination of the red, green and blue (RGB) colors is the most common approach used. Thus, looking at color images it is possible to get a crude view of the amount of absorption taking place on the surface of the sample at the three selected wavelengths. To obtain more spectral information at each pixel of the image, it is necessary to use a much broader range of wavelengths. This can be obtained using a hyperspectral camera (Campbell 2002). In hyperspectral imaging, each pixel is represented by a whole spectrum rather than three wavelengths, and the spectrum can cover a portion of the electromagnetic spectrum anywhere from ultraviolet to infrared (200 nm to 12 μm), depending on the application and requirements (Buttingsrud and Alsberg 2006; Ariana and Lu 2010ab).

The obtained data is a three-dimensional data cube, also called hypercube, containing two spatial dimensions and one spectral dimension, as represented by Figure 7. Thus, hyperspectral imaging is able to provide physical and geometrical features as well as chemical composition of the sample (Elmasry et al. 2012). In recent years hyperspectral imaging technique has been regarded as a smart and promising analytical tool in many fields such as agriculture, ecology, geology, meteorology, remote sensing, and so on, enabling direct identification of different components and their spatial distribution in the sample.

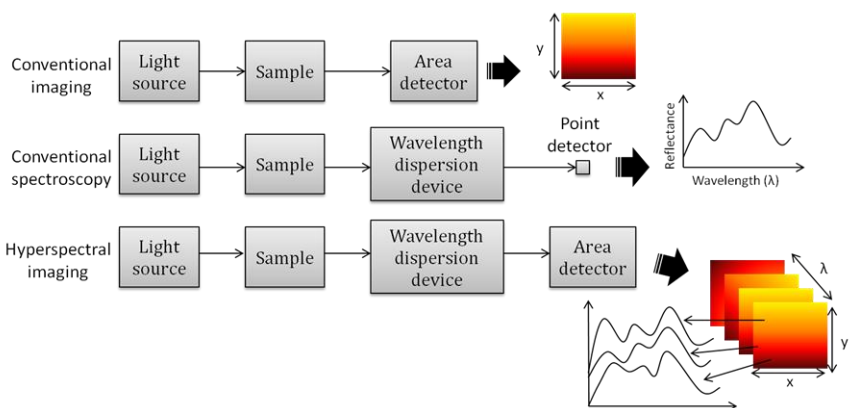


Figure 7. General system configurations for conventional imaging and spectroscopy and hyperspectral imaging.

One of the most successful applications of hyperspectral imaging is in food quality and safety assessment, where Vis/NIR and NIR spectral imaging combination provide direct identification of different components and their spatial distribution in the sample.

Conventional multivariate methods, such as PCA, LDA, PLS-DA and PLS, have been extensively employed to the analysis of food hyperspectral imaging. Some of the applications are: prediction of beef tenderness (Naganathan et al. 2008), detection of green citrus fruit in individual trees (Okamoto and Lee 2009), evaluation of internal defect and surface color of pickles (Ariana and Lu 2010ab), detection of the cooking front in potatoes (Trong et al. 2011), quality evaluation and detection of damaged tissue on mushrooms (Taghizadeh, Gowen and O'Donnell 2011ab) and prediction of some quality attributes of lamb meat (Kamruzzaman et al. 2012).

A major limiting factor hindering direct commercial application of hyperspectral technology for online use is the speed needed for acquisition, processing and analysis of hyperspectral image data (Gowen et al. 2007). One way of overcoming this problem is to implement hyperspectral imaging technology in multispectral imaging mode (Chao, Yang and Kim 2010), which means, reduce the number of spectral bands to normally less than 10 bands (Elmasry et al. 2012). Chao and coworkers (2010) developed a spectral line-scan imaging system for automated online wholesomeness inspection of broilers. A 140 bird-per-minute processing line was analyzed to optimize the region of interest (ROI) size and location, and to determine key wavebands by which to implement online high-speed multispectral inspection. In order to detect internal defect in pickling cucumbers and whole pickles, Ariana and Lu (2010ab) applied branch and bound algorithm with Mahalanobis distance for wavebands selection, reducing the number of wavebands for analysis from 101 to 4. The selected wavebands were then used for pixel classification by k -nearest neighbor (kNN) classifier. The results demonstrated that it is possible to obtain satisfactory classification accuracy even when using only a few wavebands from the original hypercube data.

In the case of remote sensing data, although the feature of linearity in a data may be transformed mathematically, in some complex data nonlinearity may never be approximated (Jensen, Qiu and Ji 1999). In this context, artificial NN have been used in many applications such as estimation of surface water quality (Zhang et al. 2002), forest cover estimation (Boyd, Foody and Ripple 2002), identification of weed stress and nitrogen status of corn (Goel et al. 2003), mapping nitrogen concentration in grass (Mutanga and Skidmore 2004) and corn yield prediction (Uno et al. 2005). Even though NN

have achieved success in the classification of complex data sets, the technique is slow during the training phase.

Some studies indicate that NN classifiers have problems in setting various parameters during training, which may be a limitation in the case of hyperspectral datasets classification, since the complexity of the network architecture increases manifolds (Varshney and Arora 2004).

5.2.1. Support Vector Machines

Support Vector Machines (SVM) is a relatively new generation of techniques for classification and regression problems based on the principle of statistical learning theory. The aim of this method is to find a linear separating hyperplane, which is a multidimensional space that separates classes of interest.

The hyperplane is placed between classes in such a way that it satisfies two conditions. First, all the data vectors that belong to the same class are placed on the same side of the hyperplane. Second, the distance between the closest data vectors in both the classes is maximized. For each class, the data vectors forming the boundary of the classes are located on supporting hyperplanes. Thus, these vectors are called the support vectors (Figure 8). However, it can be the case that no hyperplane exists to separate the input data without error. As an alternative, the data is transformed to a higher dimensional space by means of a non-linear kernel transformation that spreads the data apart such that a linear separating hyperplane may be found (Varshney and Arora 2004).

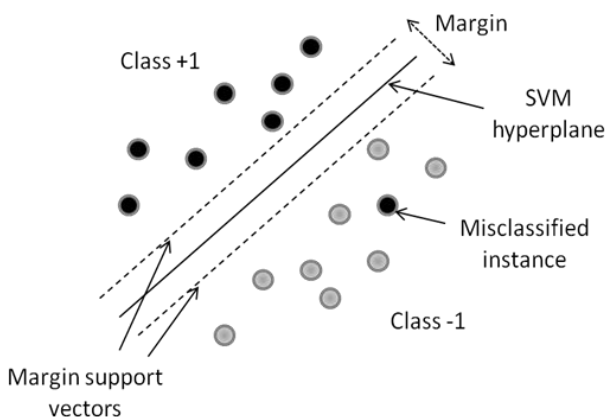


Figure 8. Optimal separating hyperplane in SVMs.

Due to the following presented advantages, SVMs have been indicated as an appropriate classifier even in high dimensional hyperspectral data as remote sensing images (Varshney and Arora 2004):

- SVMs produce accurate classifications from high dimensional data with limited number of training samples, overcoming the Hughes phenomenon. The Hughes phenomenon states that the ratio of the number of pixels with known class identity and the number of bands must be maintained at or above some minimum value in order to achieve statistical confidence;
- Unlike NN, SVMs are robust to the overfitting problem. SVMs do not minimize the empirical training error, but aim to maximize the margin between two classes of interest by placing a linear separating hyperplane between them. While doing so, the upper bound on the generalization error is minimized, providing higher generalization capability than NN;
- SVMs have a less complex structure even with high dimensional data.

In order to classify remote sensing images, Melgani and Bruzzone (2004) compared the performance of SVMs with other two non-parametric methods: radial basis function NN and kNN classifier. Results indicated that (a) SVMs were much more effective in terms of classification accuracy, computational time, and stability to parameter setting, (b) SVMs seemed more effective than the traditional pattern recognition approach, based on the combination of a feature extraction/selection procedure, and (c) SVMs exhibited low sensitivity to the Hughes phenomenon, resulting in an excellent approach to avoid the usually time-consuming phase required by any feature-reduction method. In a subsequent work, Oommen and coworkers (2008) presented a comparative analysis between SVMs and the maximum likelihood classification (MLC) method, the most conventional supervised classification method for remote sensing data. In their study, SVMs overperformed MLC in classification accuracy, robustness, and did not suffer from Hughes effect, in both multispectral and hyperspectral cases.

Other studies on the application of SVMs to the classification of remote sensing images are available in the literature (Melgani and Bruzzone 2004; Plaza, Plaza, and Barra 2009; Tarabalka, Chanussot, and Benediktsson 2010). It is important to mention that besides the employment in remote sensing imaging analysis, SVMs have been also used in a wide range of spectral

datasets. Du, Jeong, and Kong (2007) used a SVM to automatic detect and classify poultry skin tumors based on hyperspectral fluorescence imaging. A novel approach to analyze the tongue surface information based on hyperspectral medical tongue images with SVM was proposed by Zhi et al. (2007).

A simplified version of standard SVMs, least squares support vector machine (LS-SVM), has also been reported as a powerful tool for classification problems. It uses a linear set of equations instead of a quadratic programming problem to obtain the support vectors (SVs) (Tao, Chen, and Zhao 2009). Some applications involve acidity prediction in grapes by NIRS (Chauchard et al. 2004), localization of embedded inclusions in tissues using fluorescence (Chauchard et al. 2008) and quantification of protein in milk powder by FT-NIR and FT-MIR (Wu et al. 2008). In the recent study by Akbari et al. (2012), LS-SVM was used to classify hyperspectral images of prostate cancer in tumor-bearing mice and on pathology slides versus those of normal tissue. Preliminary results with 11 mice showed that the sensitivity and specificity of the hyperspectral image classification method are 92.8% to 2.0% and 96.9% to 1.3%, respectively. The proposed method may lead to advances in the optical diagnosis of prostate cancer, helping physicians to dissect malignant regions with a safe margin.

SVM was also used for detection and identification of colonies of multiple pathogens in a novel and remarkable label-free light-scattering system. The BARDOT (Bacteria Rapid Detection using Optical scattering Technology) system is a new optical sensor for detection and identification of multiple pathogens colonies that has shown great promise for distinguishing bacterial cultures on Petri dish in real time and without destroying the colony. Since the proposed technique relies on the biophysical properties of the bacterial colonies, rather than on genetic or biochemical markers, it can be readily adapted to recognize any new forms of the pathogens of interest, as some infectious agents are characterized by a high mutation rate, by simply retraining the classifier on a newest of scatter patterns (Rajwa et al. 2010; Huff et al. 2012). In the studies using BARDOT, first, the visual features were quantified using pseudo-Zernike moments and Haralick texture features. This already represents an advantage, considering that the traditional approaches to colony recognition based on the implementation of automated image-analysis systems remain limited to only few specific cases, and it cannot be easily generalized. Then, the selected numerical features were classified using support vector machine with linear kernel (SVM-L) and support vector machine with radial-basis function kernel (SVM-RBF) Successful

classifications have been demonstrated for cultures of *Listeria*, *Staphylococcus*, *Salmonella*, *Vibrio*, and *Escherichia* at genus and species level (Banada et al. 2009), different serotypes of *Salmonella enterica* (Rajwa et al. 2010) and three species of *Vibrio* (Huff et al. 2012).

CONCLUDING REMARKS

Because of their appeal as rapid, reliable and non-destructive analyses, spectroscopy techniques have been considered to be primary research tools and dominant devices in many fields such as agriculture, ecology, geology, medicine, meteorology, and so on. Due to the high complexity of spectral data, which contain a large number of variables, multivariate statistical analysis is required to recognize patterns from samples. The traditional pattern recognition techniques applied to spectral data, such as PCA, HCA, LDA, PLS-DA, PLSR, are still being extensively used and in most cases providing highly accurate results.

However, there is an increasing demand for techniques that, besides being accurate, can reduce the dimension of the data and provide some interpretation of a model, revealing an explicit relationship between the objective of the model and the given variables. This is important for understanding practical problems, especially in the case of spectral data, where the number of variables is usually larger than the number of samples. One possibility is the use of sparse learning dimensionality reduction algorithms such as Lasso regression and Elastic net.

Furthermore, in terms of applications, there is also a tendency to change from linear to non-linear models. For many years the contribution of nonlinearity to the error of spectroscopic calibrations was not generally recognized.

Trying to improve calibration performance, much attention and concern was given to issues like random noise, choice of factors and variables, and investigations into the best data transform. Nowadays it is notorious that in certain applications, such as remote sensing, linearity may not be approximated, thus requiring non-linear models. In this context, artificial NN and, above all, SVMs have been more and more explored. SVMs offers the benefits of producing accurate classifications even from high dimensional data with limited number of training samples and being robust to the overfitting problem. With the advances and higher applicability of imaging spectroscopy, the previously mentioned techniques will be important tools in the future.

ACKNOWLEDGMENTS

Authors Craig and Franca gratefully acknowledge financial support from CNPq: Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brasil.

REFERENCES

- Abbas, O., J.A. Fernandez Pierna, R. Codony, C. von Holst, and V. Baeten. 2009. Assessment of the discrimination of animal fat by FT-Raman spectroscopy. *Journal of Molecular Structure* 924-26:294-300.
- Adams, M.J. 1995. *Chemometrics in analytical spectroscopy*, RSC analytical spectroscopy monographs. Cambridge: Royal Society of Chemistry.
- Akbari, H., L.V. Halig, D.M. Schuster, A. Osunkoya, V. Master, P.T. Nieh, G.Z. Chen, and B. Fei. 2012. Hyperspectral imaging and quantitative analysis for prostate cancer detection. *Journal of Biomedical Optics* 17 (7):076005.
- Arenas-García, J. and K.B. Petersen. 2009. Kernel Multivariate Analysis in Remote Sensing Feature Extraction. In *Kernel Methods for Remote Sensing Data Analysis*: John Wiley and Sons, Ltd.
- Ariana, D.P. and R. Lu. 2010a. Evaluation of internal defect and surface color of whole pickles using hyperspectral imaging. *Journal of Food Engineering* 96 (4):583-590.
- Ariana, D.P. and R. Lu. 2010b. Hyperspectral waveband selection for internal defect detection of pickling cucumbers and whole pickles. *Computers and Electronics in Agriculture* 74 (1):137-144.
- Arzberger, U. and D.W. Lachenmeier. 2008. Fourier Transform Infrared Spectroscopy with Multivariate Analysis as a Novel Method for Characterizing Alcoholic Strength, Density, and Total Dry Extract in Spirits and Liqueurs. *Food Analytical Methods* 1 (1):18-22.
- Banada, P.P., K. Huff, E. Bae, B. Rajwa, A. Aroonnu, B. Bayraktar, A. Adil, J.P. Robinson, E.D. Hirleman and A.K. Bhunia. 2009. Label-free detection of multiple bacterial pathogens using light-scattering sensor. *Biosensors and Bioelectronics* 24 (6):1685-1692.
- Barretina, J., G. Caponigro, N. Stransky, K. Venkatesan, A.A. Margolin, S. Kim, C.J. Wilson, J. Lehar, G.V. Kryukov, D. Sonkin, A. Reddy, M. Liu, L. Murray, M.F. Berger, J.E. Monahan, P. Morais, J. Meltzer, A. Korejwa,

- J. Jane-Valbuena, F.A. Mapa, J. Thibault, E. Bric-Furlong, P. Raman, A. Shipway, I.H. Engels, J. Cheng, G.K. Yu, J. Yu, P. Aspesi, Jr., M. de Silva, K. Jagtap, M.D. Jones, L. Wang, C. Hatton, E. Palescandolo, S. Gupta, S. Mahan, C. Sougnez, R.C. Onofrio, T. Liefeld, L. MacConaill, W. Winckler, M. Reich, N. Li, J.P. Mesirov, S.B. Gabriel, G. Getz, K. Ardlie, V. Chan, V.E. Myer, B.L. Weber, J. Porter, M. Warmuth, P. Finan, J.L. Harris, M. Meyerson, T.R. Golub, M.P. Morrissey, W.R. Sellers, R. Schlegel and L.A. Garraway. 2012. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* 483 (7391):603-607.
- Beckwith-Hall, B.M., J.T. Brindle, R.H. Barton, M. Coen, E. Holmes, J.K. Nicholson and H. Antti. 2002. Application of orthogonal signal correction to minimise the effects of physical and biological variation in high resolution ¹H NMR spectra of biofluids. *Analyst* 127 (10):1283-1288.
- Benoudjit, N., E. Cools, M. Meurens and M. Verleysen. 2004. Chemometric calibration of infrared spectrometers: selection and validation of variables by non-linear models. *Chemometrics and Intelligent Laboratory Systems* 70 (1):47-53.
- Bishop, C.M. and G. Hinton. 1995. *Neural Networks for Pattern Recognition*: Clarendon Press.
- Botelho, B.G, B.A.P. Mendes and M.M. Sena. 2012. Development and Analytical Validation of Robust Near-Infrared Multivariate Calibration Models for the Quality Inspection Control of Mozzarella Cheese. *Food Analytical Methods*:1-11.
- Boueri, M., V. Motto-Ros, W.-Q. Lei, L. Qain, L.-J. Zheng, H.-P. Zeng and J. Yu. 2011. Identification of Polymer Materials Using Laser-Induced Breakdown Spectroscopy Combined with Artificial Neural Networks. *Applied Spectroscopy* 65 (3):307-314.
- Boyd, D.S., G.M. Foody and W.J. Ripple. 2002. Evaluation of approaches for forest cover estimation in the Pacific Northwest, USA, using remote sensing. *Applied Geography* 22 (4):375-392.
- Brewster, V. L., L. Ashton and R. Goodacre. 2011. Monitoring the Glycosylation Status of Proteins Using Raman Spectroscopy. *Analytical Chemistry* 83 (15):6074-6081.
- Buttingsrud, B. and B.K. Alsberg. 2006. Supperresolution of hyperspectral images. *Chemometrics and Intelligent Laboratory Systems* 84 (1-2):62-68.
- Campbell, J.B. 2002. *Introduction to Remote Sensing*: Taylor and Francis.

- Chao, K., C.-C. Yang, and M.S. Kim. 2010. Spectral line-scan imaging system for high-speed non-destructive wholesomeness inspection of broilers. *Trends in Food Science and Technology* 21 (3):129-137.
- Chauchard, F., R. Cogdill, S. Roussel, J.M. Roger and V. Bellon-Maurel. 2004. Application of LS-SVM to non-linear phenomena in NIR spectroscopy: development of a robust and portable sensor for acidity prediction in grapes. *Chemometrics and Intelligent Laboratory Systems* 71 (2):141-150.
- Chauchard, F., J. Svensson, J. Axelsson, S. Andersson-Engels and S. Roussel. 2008. Localization of embedded inclusions using detection of fluorescence: Feasibility study based on simulation data, LS-SVM modeling and EPO pre-processing. *Chemometrics and Intelligent Laboratory Systems* 91 (1):34-42.
- Chen, H., and M. Han. 2011. Raman spectroscopic study of the effects of microbial transglutaminase on heat-induced gelation of pork myofibrillar proteins and its relationship with textural characteristics. *Food Research International* 44 (5):1514-1520.
- Chen, X.W., X.B. Zhou and S. T. C. Wong. 2006. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *Ieee Transactions on Biomedical Engineering* 53 (4):762-766.
- Chevallier, S., D. Bertrand, A. Kohler and P. Courcoux. 2006. Application of PLS-DA in multivariate image analysis. *Journal of Chemometrics* 20 (5):221-229.
- Clark, M. and R.D. Cramer. 1993. The Probability of Chance Correlation Using Partial Least Squares (PLS). *Quantitative Structure-Activity Relationships* 12 (2):137-145.
- Clemmensen, L., T. Hastie, D. Witten and B. Ersbøll. 2011. Sparse Discriminant Analysis. *Technometrics* 53 (4):406-413.
- Cozzolino, D., W.U. Cynkar, N. Shah and P.A. Smith. 2011. Can spectroscopy geographically classify Sauvignon Blanc wines from Australia and New Zealand? *Food Chemistry* 126 (2):673-678.
- Craig, A.P., A.S. Franca and L.S. Oliveira. 2011a. Discrimination between Immature and Mature Green Coffees by Attenuated Total Reflectance and Diffuse Reflectance Fourier Transform Infrared Spectroscopy. *Journal of Food Science* 76 (8):C1162-C1168.
- Craig, A.P., A.S. Franca and L.S. Oliveira. 2011b. Discrimination between defective and non-defective green coffees by Diffuse Reflectance Fourier Transform Spectroscopy. In: *First International Congress on Cocoa*

- Coffee and Tea*, 2011b, Novara. Proceedings First International Congress on Cocoa Coffee and Tea 191-194.
- Craig, A.P., A.S. Franca and L.S. Oliveira. 2012a. Discrimination between defective and non-defective roasted coffees by diffuse reflectance infrared Fourier transform spectroscopy. *LWT-Food Science and Technology* 47 (2):505-511.
- Craig, A.P., A.S. Franca and L.S. Oliveira. 2012b. Evaluation of the potential of FTIR and chemometrics for separation between defective and non-defective coffees. *Food Chemistry* 132 (3):1368-1374.
- Das, R.S. and Y.K. Agrawal. 2011. Raman spectroscopy: Recent advancements, techniques and applications. *Vibrational Spectroscopy* 57 (2):163-176.
- De Maesschalck, R., D. Jouan-Rimbaud, and D.L. Massart. 2000. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems* 50 (1):1-18.
- De Veij, M., A. Deneckere, P. Vandenabeele, D. de Kaste and L. Moens. 2008. Detection of counterfeit Viagra® with Raman spectroscopy. *Journal of Pharmaceutical and Biomedical Analysis* 46 (2):303-309.
- Dochow, S., C. Krafft, U. Neugebauer, T. Bocklitz, T. Henkel, G. Mayer, J. Albert and J. Popp. 2011. Tumour cell identification by means of Raman spectroscopy in combination with optical traps and microfluidic environments. *Lab on a Chip* 11 (8):1484-1490.
- Du, Z., M.K. Jeong and S.G. Kong. 2007. Band selection of hyperspectral images for automatic detection of poultry skin tumors. *IEEE Transactions on Automation Science and Engineering* 4 (3):332-339.
- Dyar, M.D., M.L. Carmosino, E.A. Breves, M.V. Ozanne, S.M. Clegg and R.C. Wiens. 2012. Comparison of partial least squares and lasso regression techniques as applied to laser-induced breakdown spectroscopy of geological samples. *Spectrochimica Acta Part B-Atomic Spectroscopy* 70:51-67.
- El-Abassy, R. M., P. Donfack and A. Materny. 2011. Discrimination between Arabica and Robusta green coffee using visible micro Raman spectroscopy and chemometric analysis. *Food Chemistry* 126 (3):1443-1448.
- Elmasry, G., M. Kamruzzaman, D.-W. Sun and P. Allen. 2012. Principles and applications of hyperspectral imaging in quality evaluation of agro-food products: a review. *Critical reviews in food science and nutrition* 52 (11):999-1023.

- Friedman, J., Trevor H. and R. Tibshirani. 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* 33 (1):1-22.
- Fu, G.H., Q.S. Xu, H.D. Li, D.S. Cao and Y.Z. Liang. 2011. Elastic Net Grouping Variable Selection Combined with Partial Least Squares Regression (EN-PLSR) for the Analysis of Strongly Multi-collinear Spectroscopic Data. *Applied Spectroscopy* 65 (4):402-408.
- Gaus, K., P. Rösch, R. Petry, K.D. Peschke, O. Ronneberger, H. Burkhardt, K. Baumann and J. Popp. 2006. Classification of lactic acid bacteria with UV-resonance Raman spectroscopy. *Biopolymers* 82 (4):286-290.
- Gavaghan, C.L., I.D. Wilson and J.K. Nicholson. 2002. Physiological variation in metabolic phenotyping and functional genomic studies: use of orthogonal signal correction and PLS-DA. *FEBS Letters* 530 (1-3):191-196.
- Gniadecka, M., P.A. Philipsen, S. Sigurdsson, S. Wessel, O.F. Nielsen, D.H. Christensen, J. Hercogova, K. Rossen, H.K. Thomsen, R. Gniadecki, L.K. Hansen and H.C. Wulf. 2004. Melanoma diagnosis by Raman spectroscopy and neural networks: structure alterations in proteins and lipids in intact cancer tissue. *The Journal of investigative dermatology* 122 (2):443-449.
- Goel, P.K., S.O. Prasher, R.M. Patel, J.A. Landry, R.B. Bonnell and A.A. Viau. 2003. Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn. *Computers and Electronics in Agriculture* 39 (2):67-93.
- Gowen, A.A., C.P. O'Donnell, P.J. Cullen, G. Downey and J.M. Frias. 2007. Hyperspectral imaging - an emerging process analytical tool for food quality and safety control. *Trends in Food Science and Technology* 18 (12):590-598.
- Gurdeniz, G. and B. Ozen. 2009. Detection of adulteration of extra-virgin olive oil by chemometric analysis of mid-infrared spectral data. *Food Chemistry* 116 (2):519-525.
- Haiqing, Y., L. Weiqiang, X. Ning and A.M. Mouazen. 2012. Prediction of organic and inorganic carbon contents in soil: Vis-NIR vs. MIR spectroscopy. Paper read at Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on, 21-23 April 2012.
- Hastie, T., R. Tibshirani and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Second Edition): Springer.

- Hof, M. 2005. Basics of Optical Spectroscopy. In *Handbook of Spectroscopy*: Wiley-VCH Verlag GmbH and Co. KGaA.
- Hollas, J.M. 2004. *Modern Spectroscopy*: John Wiley and Sons.
- Holmes, E. and H. Antti. 2002. Chemometric contributions to the evolution of metabonomics: mathematical solutions to characterising and interpreting complex biological NMR spectra. *Analyst* 127 (12):1549-1557.
- Hudson, S.D. and G. Chumanov. 2009. Bioanalytical applications of SERS (surface-enhanced Raman spectroscopy). *Analytical and Bioanalytical Chemistry* 394 (3):679-686.
- Huff, K., A. Aroonual, A.E.F. Littlejohn, B. Rajwa, E. Bae, P.P. Banada, V. Patsekin, E. D. Hirleman, J.P. Robinson, G.P. Richards and A.K. Bhunia. 2012. Light-scattering sensor for real-time identification of *Vibrio parahaemolyticus*, *Vibrio vulnificus* and *Vibrio cholerae* colonies on solid agar plate. *Microbial Biotechnology* 5 (5):607-620.
- Indahl, U.G., N.S. Sahni, B. Kirkhus, and T. Næs. 1999. Multivariate strategies for classification based on NIR-spectra—with application to mayonnaise. *Chemometrics and Intelligent Laboratory Systems* 49 (1):19-31.
- Jensen, J.R., F. Qiu, and M.H. Ji. 1999. Predictive modelling of coniferous forest age using statistical and artificial neural network approaches applied to remote sensor data. *International Journal of Remote Sensing* 20 (14):2805-2822.
- Johnson, R. and D. Wichern. 2007. *Applied Multivariate Statistical Analysis* (6th Edition): Prentice Hall.
- Kamau-Rewe, M., F. Rasche, J. G. Cobo, G. Dercon, K.D. Shepherd and G. Cadisch. 2011. Generic Prediction of Soil Organic Carbon in Alfisols Using Diffuse Reflectance Fourier-Transform Mid-Infrared Spectroscopy. *Soil Sci. Soc. Am. J.* 75 (6):2358-2360.
- Kamruzzaman, M., G. ElMasry, D.-W. Sun, and P. Allen. 2012. Prediction of some quality attributes of lamb meat using near-infrared hyperspectral imaging and multivariate analysis. *Analytica Chimica Acta* 714:57-67.
- Koujelev, A., M. Sabsabi, V. Motto-Ros, S. Laville and S. L. Lui. 2010. Laser-induced breakdown spectroscopy with artificial neural network processing for material identification. *Planetary and Space Science* 58 (4):682-690.
- Krafft, C., K. Thümmel, S.B. Sobottka, G. Schackert and R. Salzer. 2006. Classification of malignant gliomas by infrared spectroscopy and linear discriminant analysis. *Biopolymers* 82 (4):301-305.
- Lakowicz, J.R. 2009. *Principles of Fluorescence Spectroscopy*: Springer London, Limited.

- Lasch, P. 2012. Spectral pre-processing for biomedical vibrational spectroscopy and microspectroscopic imaging. *Chemometrics and Intelligent Laboratory Systems* 117 (0):100-114.
- Le C., Kim-Anh, S. Boitard, and P. Besse. 2011. Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics* 12 (1):253.
- Li, B., P. Beveridge, W.T. O'Hare and M. Islam. 2011. The estimation of the age of a blood stain using reflectance spectroscopy with a microspectrophotometer, spectral pre-processing and linear discriminant analysis. *Forensic Science International* 212 (1-3):198-204.
- Liu, L., D. Cozzolino, W.U. Cynkar, R.G. Damberg, L. Janik, B.K. O'Neill, C.B. Colby and M. Gishen. 2008. Preliminary study on the application of visible-near infrared spectroscopy and chemometrics to classify Riesling wines from different countries. *Food Chemistry* 106 (2):781-786.
- MacIntyre, D.A., B. Jimenez, E. Jantus Lewintre, C. Reinoso Martin, H. Schaefer, C. Garcia Ballesteros, J. Ramon Mayans, M. Spraul, J. Garcia-Conde, and A. Pineda-Lucena. 2010. Serum metabolome analysis by H-1-NMR reveals differences between chronic lymphocytic leukaemia molecular subgroups. *Leukemia* 24 (4):788-797.
- Mantanus, J., E. Ziémons, P. Lebrun, E. Rozet, R. Klinkenberg, B. Streel, B. Evrard and Ph Hubert. 2009. Moisture content determination of pharmaceutical pellets by near infrared spectroscopy: Method development and validation. *Analytica Chimica Acta* 642 (1-2):186-192.
- Marcos-Martinez, D., J.A. Ayala, R.C. Izquierdo-Hornillos, F.J. Manuel de Villena and J. O. Caceres. 2011. Identification and discrimination of bacterial strains by laser induced breakdown spectroscopy and neural networks. *Talanta* 84 (3):730-737.
- Mark, H. and J. Workman Jr. 2007a. Linearity in Calibration: Act III Scene I: The Importance of Nonlinearity. In *Chemometrics in Spectroscopy*. Amsterdam: Academic Press.
- Mark, H. and J. Workman Jr. 2007b. The Statistics of Spectral Searches. In *Chemometrics in Spectroscopy*. Amsterdam: Academic Press.
- Markou, M. and S. Singh. 2003a. Novelty detection: a review - part 2: neural network based approaches. *Signal Processing* 83 (12):2499-2521.
- Markou, M. and S. Singh. 2003b. Novelty detection: a review—part 1: statistical approaches. *Signal Processing* 83 (12):2481-2497.
- Massart, D.L. 1988. *Chemometrics: A Textbook*: Elsevier.

- Melgani, F. and L. Bruzzone. 2004. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing* 42 (8):1778-1790.
- Mireei, S.A. and M. Sadeghi. 2013. Detecting bunch withering disorder in date fruit by near infrared spectroscopy. *Journal of Food Engineering* 114 (3):397-403.
- Mutanga, O. and A.K. Skidmore. 2004. Integrating imaging spectroscopy and neural networks to map grass quality in the Kruger National Park, South Africa. *Remote Sensing of Environment* 90 (1):104-115.
- Naganathan, G.K., L.M. Grimes, J. Subbiah, C.R. Calkins, A.Samal and G.E. Meyer. 2008. Visible/near-infrared hyperspectral imaging for beef tenderness prediction. *Computers and Electronics in Agriculture* 64 (2):225-233.
- Narlikar, A.V. and Y.Y. Fu. 2010. *Oxford Handbook of Nanoscience and Technology, Volume 1: Basic Aspects*: OUP Oxford.
- Narlikar, A.V. 2010. *Oxford Handbook of Nanoscience and Technology, Volume 3: Applications*: OUP Oxford.
- Okamoto, H. and W.S. Lee. 2009. Green citrus detection using hyperspectral imaging. *Computers and Electronics in Agriculture* 66 (2):201-208.
- Oommen, T., D. Misra, N.K.C. Twarakavi, A. Prakash, B. Sahoo and S. Bandopadhyay. 2008. An objective analysis of Support Vector Machine based classification for remote sensing. *Mathematical Geosciences* 40 (4):409-424.
- Özbalci, B., İ.H. Boyacı, A. Topcu, C. Kadılar and U. Tamer. 2013. Rapid analysis of sugars in honey by processing Raman spectrum using chemometric methods and artificial neural networks. *Food Chemistry* 136 (3-4):1444-1452.
- Pavia, D.L. 2009. *Introduction to Spectroscopy*: Brooks/Cole.
- Pears, M.R., J.D. Cooper, H.M. Mitchison, R.J. Mortishire-Smith, D.A. Pearce and J.L. Griffin. 2005. High Resolution ¹H NMR-based Metabolomics Indicates a Neurotransmitter Cycling Deficit in Cerebral Tissue from a Mouse Model of Batten Disease. *Journal of Biological Chemistry* 280 (52):42508-42514.
- Pereira, R.C.C., V.L. Skrobot, E.V.R. Castro, I.C.P. Fortes and V.M.D. Pasa. 2006. Determination of Gasoline Adulteration by Principal Components Analysis-Linear Discriminant Analysis Applied to FTIR Spectra. *Energy and Fuels* 20 (3):1097-1102.

- Pirro, V., L.S. Eberlin, P. Oliveri and R.G. Cooks. 2012. Interactive hyperspectral approach for exploring and interpreting DESI-MS images of cancerous and normal tissue sections. *Analyst* 137 (10):2374-2380.
- Pizarro, C., S. Rodríguez-Tecedor, N. Pérez-del-Notario, I. Esteban-Díez, and J.M. González-Sáiz. 2012. Classification of spanish extra virgin olive oils by data fusion of visible spectroscopic fingerprints and chemical descriptors. *Food Chemistry* 138 (2-3): 915-922.
- Plaza, J., A.J. Plaza and C. Barra. 2009. Multi-Channel Morphological Profiles for Classification of Hyperspectral Images Using Support Vector Machines. *Sensors* 9 (1):196-218.
- Poulli, K.I., G.A. Mousdis and C.A. Georgiou. 2005. Classification of edible and lampante virgin olive oil based on synchronous fluorescence and total luminescence spectroscopy. *Analytica Chimica Acta* 542 (2):151-156.
- Rajwa, B., M.M. Dundar, F. Akova, A. Bettasso, V. Patsekin, E.D. Hirleman, A.K. Bhunia, and J.P. Robinson. 2010. Discovering the unknown: Detection of emerging pathogens using a label-free light-scattering system. *Cytometry Part A* 77A (12):1103-1112.
- Reis N., A.S. Franca and L.S. Oliveira. 2013a. Detection of Coffee Husks and Roasted Corn in Admixtures with Roasted Coffees By Fourier Transform Infrared Spectroscopy (FTIR). In: *Proceedings of The 24th International Conference on Coffee Science - ASIC 2012*, San José, Costa Rica, Manuscript PC416, 4p.
- Reis N., A.S. Franca and L.S. Oliveira. 2013b. Discrimination between roasted coffee, roasted corn and coffee husks by Diffuse Reflectance Infrared Fourier Transform Spectroscopy. *LWT - Food Science and Technology* 50 (2):715-722.
- Reis N., A.S. Franca and L.S. Oliveira. 2013c. Attenuated Total Reflectance Fourier Transform Infrared Spectroscopy (ATR-FITR) for Discrimination between Roasted Coffee and Adulterants. In: *Proceedings of The 24th International Conference on Coffee Science - ASIC 2012*, San José, Costa Rica, Manuscript PC415, 4p.
- Rinnan, A., F. van den Berg and S.B. Engelsen. 2009. Review of the most common pre-processing techniques for near-infrared spectra. *TrAC-Trends in Analytical Chemistry* 28 (10):1201-1222.
- Ripley, B.D. 2008. *Pattern Recognition and Neural Networks*: Cambridge University Press.
- Roggo, Y., P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond and N. Jent. 2007. A review of near infrared spectroscopy and chemometrics in

- pharmaceutical technologies. *Journal of Pharmaceutical and Biomedical Analysis* 44 (3):683-700.
- Roussel, S., V. Bellon-Maurel, J.-M. Roger and P. Grenier. 2003. Authenticating white grape must variety with classification models based on aroma sensors, FT-IR and UV spectrometry. *Journal of Food Engineering* 60 (4):407-419.
- Rubio-Diaz, D.E., T. De Nardo, A. Santos, S. de Jesus, D. Francis and L.E. Rodriguez-Saona. 2010. Profiling of nutritionally important carotenoids from genetically-diverse tomatoes by infrared spectroscopy. *Food Chemistry* 120 (1):282-289.
- Said, M.M., S. Gibbons, A.C. Moffat and M. Zloh. 2011. Near-infrared spectroscopy (NIRS) and chemometric analysis of Malaysian and UK paracetamol tablets: A spectral database study. *International Journal of Pharmaceutics* 415 (1-2):102-109.
- Santos, J.R., M.C. Sarraguça, A.O.S.S. Rangel and J.A. Lopes. 2012. Evaluation of green coffee beans quality using near infrared spectroscopy: A quantitative approach. *Food Chemistry* 135 (3):1828-1835.
- Sarkhot, D.V., S. Grunwald, Y. Ge and C.L.S. Morgan. 2011. Comparison and detection of total and available soil carbon fractions using visible/near infrared diffuse reflectance spectroscopy. *Geoderma* 164 (1-2):22-32.
- Schmid, U., P. Roesch, M. Krause, M. Harz, J. Popp and K. Baumann. 2009. Gaussian mixture discriminant analysis for the single-cell differentiation of bacteria using micro-Raman spectroscopy. *Chemometrics and Intelligent Laboratory Systems* 96 (2):159-171.
- Shanmukh, S., L. Jones, Y.P. Zhao, J.D. Driskell, R.A. Tripp and R.A. Dluhy. 2008. Identification and classification of respiratory syncytial virus (RSV) strains by surface-enhanced Raman spectroscopy and multivariate statistical techniques. *Analytical and Bioanalytical Chemistry* 390 (6):1551-1555.
- Shao, J., Y. Wang, X. Deng and Sijian Wang. 2011. Sparse linear discriminant analysis by thresholding for high dimensional data. *Annals of Statistics* 39 (2):1241-1265.
- Sharaf, M.A., D.L. Illman and B.R. Kowalski. 1986. *Chemometrics*: John Wiley and Sons.
- Shetty, Nisha, and Rene Gislum. 2011. Quantification of fructan concentration in grasses using NIR spectroscopy and PLSR. *Field Crops Research* 120 (1):31-37.
- Smith, B.C. 2002. *Quantitative Spectroscopy: Theory and Practice*: Elsevier Science.

- Stephen, K.E., D. Homrighausen, G. Depalma, C.H. Nakatsu and J. Irudayaraj. 2012. Surface enhanced Raman spectroscopy (SERS) for the discrimination of *Arthrobacter* strains based on variations in cell surface composition. *The Analyst* 137 (18):4280-6.
- Sun, L. and J. Irudayaraj. 2009. Quantitative Surface-Enhanced Raman for Gene Expression Estimation. *Biophysical Journal* 96 (11):4709-4716.
- Taghizadeh, M., A.A. Gowen, and C.P. O'Donnell. 2011a. Comparison of hyperspectral imaging with conventional RGB imaging for quality evaluation of *Agaricus bisporus* mushrooms. *Biosystems Engineering* 108 (2):191-194.
- Taghizadeh, M., A.A. Gowen and C.P. O'Donnell. 2011b. The potential of visible-near infrared hyperspectral imaging to discriminate between casing soil, enzymatic browning and undamaged tissue on mushroom (*Agaricus bisporus*) surfaces. *Computers and Electronics in Agriculture* 77 (1):74-80.
- Tao, S., D. Chen and W. Zhao. 2009. Fast pruning algorithm for multi-output LS-SVM and its application in chemical pattern classification. *Chemometrics and Intelligent Laboratory Systems* 96 (1):63-69.
- Tarabalka, Y., J. Chanussot and J.A. Benediktsson. 2010. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition* 43 (7):2367-2379.
- Thygesen, L.G., M.M. Løkke, E. Micklander and S.B. Engelsen. 2003. Vibrational microspectroscopy of food. Raman vs. FT-IR. *Trends in Food Science and Technology* 14 (1-2):50-57.
- Tibshirani, R. 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B-Methodological* 58 (1):267-288.
- Tibshirani, R. 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society Series B-Statistical Methodology* 73:273-282.
- Trong, N., N. Do, M. Tsuta, B.M. Nicolai, J. De Baerdemaeker and W. Saeys. 2011. Prediction of optimal cooking time for boiled potatoes by hyperspectral imaging. *Journal of Food Engineering* 105 (4):617-624.
- Uno, Y., S.O. Prasher, R. Lacroix, P.K. Goel, Y. Karimi, A. Viau, and R.M. Patel. 2005. Artificial neural networks to predict corn yield from Compact Airborne Spectrographic Imager data. *Computers and Electronics in Agriculture* 47 (2):149-161.
- Varshney, P.K. and M.K. Arora. 2004. *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data*: Springer.

- Vítková, G., K. Novotný, L. Prokeš, A. Hrdlička, J. Kaiser, J. Novotný, R. Malina and D. Prochazka. 2012. Fast identification of biominerals by means of stand-off laser-induced breakdown spectroscopy using linear discriminant analysis and artificial neural networks. *Spectrochimica Acta Part B: Atomic Spectroscopy* 73:1-6.
- Vlckova, B., I. Pavel, M. Sladkova, K. Siskova and M. Slouf. 2007. Single molecule SERS: Perspectives of analytical applications. *Journal of Molecular Structure* 834–836: 42-47.
- Wakeling, I.N. and J.J. Morris. 1993. A test of significance for partial least squares regression. *Journal of Chemometrics* 7 (4):291-304.
- Wang, L. and B. Mizaikoff. 2008. Application of multivariate data-analysis techniques to biomedical diagnostics based on mid-infrared spectroscopy. *Analytical and Bioanalytical Chemistry* 391 (5):1641-1654.
- Wang, Q., S.M. Lonergan and C. Yu. 2012. Rapid determination of pork sensory quality using Raman spectroscopy. *Meat Science* 91 (3):232-239.
- Wold, S., M. Sjostrom and L. Eriksson. 2001. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* 58 (2):109-130.
- Workman, J.J., P.R. Mobley, B.R. Kowalski and R. Bro. 1996. Review of Chemometrics Applied to Spectroscopy: 1985-95, Part I. *Applied Spectroscopy Reviews* 31 (1-2):73-124.
- Workman Jr, J. 2001. Review of Chemometrics Applied to Spectroscopy: Data Preprocessing. In *The Handbook of Organic Compounds*. Burlington: Academic Press.
- Wu, D., Y. He, S. Feng and D.-W. Sun. 2008. Study on infrared spectroscopy technique for fast measurement of protein content in milk powder based on LS-SVM. *Journal of Food Engineering* 84 (1):124-131.
- Yang, H., J. Irudayaraj and M.M. Paradkar. 2005. Discriminant analysis of edible oils and fats by FTIR, FT-NIR and FT-Raman spectroscopy. *Food Chemistry* 93 (1):25-32.
- Zanon, M., M. Riz, G. Sparacino, A. Facchinetti, R.E. Suri, M.S. Talary and C. Cobelli. 2011. Assessment of linear regression techniques for modeling multisensor data for non-invasive continuous glucose monitoring. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference* 2011:2538-41.

-
- Zhang, Y.Z., J. Pulliainen, S. Koponen and M. Hallikainen. 2002. Application of an empirical neural network to surface water quality estimation in the Gulf of Finland using combined optical data and microwave data. *Remote Sensing of Environment* 81 (2-3):327-336.
- Zhi, L., D. Zhang, J.-q. Yan, Q.-L. Li and Q.-l. Tang. 2007. Classification of hyperspectral medical tongue images for tongue diagnosis. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society* 31 (8):672-8.
- Zhou, T., D. Tao and X. Wu. 2011. Manifold elastic net: a unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery* 22 (3):340-371.
- Zhu, J. and T. Hastie. 2004. Classification of gene microarrays by penalized logistic regression. *Biostatistics* 5 (3):427-443.
- Zou, H. and T. Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B-Statistical Methodology* 67:301-320.

Chapter 2

**OPTIMIZATION OF AN EMBEDDED
SIMPLIFIED FUZZY ARTMAP
IMPLEMENTED ON A MICROCONTROLLER
USING MATLAB GUI ENVIRONMENT**

***Eduardo Garcia-Breijo*, Jose Garrigues
and Luis Gil-Sanchez***

Centro de Reconocimiento Molecular y Desarrollo Tecnológico,
Unidad Mixta UPV-UV
Universitat Politècnica de València, Valencia, Spain

ABSTRACT

In the present work, a portable system based on a microcontroller has been developed to classify different kinds of honeys. In order to do this classification, a Simplified Fuzzy ARTMAP network (SFA) implemented in a microcontroller has been used. Due to the memory limits when working with microcontrollers, it is necessary to optimize the use of both program and data memory. In order to optimize the necessary parameters to programme the SFA in a microcontroller, a Graphical User Interface (GUI) for Matlab has been developed. The measures have been carried out by potentiometry techniques using a multielectrode of 7 different metals. With the information obtained in the experimental phase, the

* Email: egarciab@eln.upv.es, Telephone: +34.963877608, Fax: +34.963877609

neural network has been trained in a PC by means of the GUI in MATLAB, with the obtained parameters the microcontroller has been programmed and later new samples have been analyzed using the portable system. An important aspect in this work is that the training algorithm has been implemented to the equipment in such a way that it can add the information of the new samples to the network in order to optimize it.

1. INTRODUCTION

Nose and tongue electronic systems are electronic equipment that performs the measurement of electrical signals from a set of multiple sensors. Usually, these sensors are not specific because they are not sensitive to a specific physical, chemical or biological parameter, but they are sensitive to a global variation of the environment. Thus, qualitative analysis of different samples of complex composition can be carried out[1]. Sensors can be very diverse in nature, emphasizing those that use electrochemical analysis techniques, as potentiometry[2], voltammetry [3], impedance spectroscopy [4], etc. The analysis of the measures is performed with the data set, so we can achieve a classification of samples according to several categories of interest.

Some of the most commonly used methods to make a sample classification are those based on artificial neural networks [5], named in this way by their analogy with biological neural systems because they consist of a set of neurons linked together.

Neural networks are implemented by mathematical algorithms in order to develop equations that relate outputs with inputs. When analyzing data with neural networks, two stages must be applied: a first stage for training in which each output is related with all the inputs to establish the weights for each neuron, and a second stage consisting of verifying the network to determine its ability to carry out appropriate classification of new samples.

There are several types of neural networks, they are usually classified based on learning methods: *supervised*, *unsupervised* and *reinforced* but they can also be classified based on architectures: *single-layer feed forward*, *multilayer feed forward* and *recurrent neural network*. A recurrent neural network with unsupervised learning methods is the Adaptive Resonance Theory (ART). This theory was developed in 1976 by Grossberg[6]and it proposes a model for artificial neural networks whose operation is based on the way the human brain processes the information, describing a series of neural network models using supervised and not supervised learning methods

to tackle recognition problems and pattern recognition. In 1987, Carpenter and Grossberg developed the named ART or ART1 network[7], which tried to solve the stability-plasticity dilemma. This first network, ART1, only worked with vectors of digital inputs, and so it did not have much success. In 1987, they developed the ART2 network[8] to work with analogical information. In 1990, the ART3 was developed by the same authors[9] and it included chemical transmitters to control the searching category processes into the network. In 1991, ART2-A was published. It was a faster version than ART2. Next, ARTMAP network [10] was published the same year. ARTMAP is a supervised version of ART network. In 1991, Fuzzy ART [11][12] was also published as a synthesis of Fuzzy Logic Theory. Finally, in 1982, the Fuzzy ARTMAP [13] was published as a supervised version of Fuzzy ART. As the application of these networks was complicated, authors developed later (in 1991 and 1992) their respective algorithmic versions [14]-[15].

Fuzzy ARTMAP and Fuzzy ARTMAP Modified Algorithms have been applied to a large number of applications such as nose electronic systems [16][17] and electronic tongues [18][19]. These applications have several advantages: ease of use, good results for a limited number of samples, low computational cost, and transparency and relative simplicity of the implemented algorithms. These algorithms usually work in computer systems based on a PC and they are able to analyze the data of the measures obtained before. But one of the lines to improve the electronic tongue systems is the development of electronic systems capable to perform analyses of the samples in situ. Due to this, autonomous equipment is interesting because it is flexible and easy to use so it can be used by non-specialized personnel. In order to develop the autonomous equipment, the incorporation of the neural network in a standalone digital electronic system must be done. The easiest way to create a digital electronic system with these characteristics is using microcontroller devices because they are cheap, relatively easy to program, information-rich, easy handling and they have low power consumption. The limited memory is one of the main features that set microprocessors apart from PC-like systems. Because of this, minimization of the required memory is fundamental when tackling the task to embed a neural network into a microprocessor. This is one of the challenges of this task [19].

Another feature that differs microcontrollers to the PC is the processing speed. Microcontroller devices are usually slower than PC microprocessors, but this limitation is not usually important for electronic tongue systems because most of the applications are related to industrial control, specifically, to the monitoring of food properties, and computational speed doesn't seem to

be a critical condition in these applications. But the microprocessor memory minimization must be limited to the neural network. In this way, it obtains the best possible results for the analyzed sample classification. Attending to this, a variation of the network parameters to determine the accuracy in sample classification has been made. The neural network used in this work is a simplified version of the Fuzzy ARTMAP network and so, it is named Simplified Fuzzy ARTMAP[21]. This version simplifies the original algorithm created by Carpenter maintaining good performance. These algorithms are developed to run on mathematical calculation programs such as MATLAB, through various scripts that facilitate their use [22].

In order to perform the Fuzzy ARTMAP network analysis and its incorporation into a microcontroller, data of an experience in the field of food and agriculture have been used, specifically, the measurements obtained in the monitoring of different kinds of honey. The data were obtained by an electronic tongue system using potentiometric techniques and it consisted of a set of seven metallic electrodes of different materials. The analyses consisted of measuring four different botanic-origin honey samples. Three of these samples were monofloral honey (citric, rosemary and honeydew) and the fourth one was a mix of different origin honeys (polyfloral). In addition to the botanic origin, three physical treatments that are usually made to the honey have been taken into count: raw, liquation and pasteurization.

Companies that produce and commercialize honey must indicate its floral origin. This is due to the origin dependence of certain physical and organoleptic characteristics that are of interest to the consumer. The interest of our application is to determine the origin of any honey sample. In general, monofloral honeys have higher commercial value to the producers because certain properties (color, consistency, odor, taste, etc.) are guaranteed. For this reason, companies are increasing their quality control standards and are looking forward to the development of innovative systems to identify the botanical origin of honey.

Classical methods to determine the botanical origin of honey is based on melissopalynology analysis. This analytical technique is based on the identification and quantification of the percentage of one type of pollen by means of a microscopic examination[23]. However, this technique is very tedious and requires highly skilled analysts and specialized equipment, so the analysis of 100% of the samples is unrealistic. Due to this, searching for new analytical methods, such as electronic tongue systems, is an interesting task.

Moreover, the honey is usually treated with some physical processes such as liquefaction and pasteurization that facilitates its storage and consumption.

In the first case, a warming up to 45 °C and 55 °C for approximately two days is performed. In the second case, pasteurization, honey is subjected to a high thermal shock (75-85°C) for a short time (2-6 minutes). In this way, the initial crystalline structures that promote full or partial crystallization of honey are destroyed and it allows honey to remain liquid for longer. But at the same time, these processes must not constitute a variation of the intrinsic characteristics of honey. Details of these samples and measurements have been published in a previous work[24] showing the measurements made with honey samples from the four different floral origins described above, as well as with three types of heat treatments (oil, liquefied and pasteurized) and four replicates with each of the 12 analyzed samples. Measurements were performed using a potentiometric electronic tongue system. As described before, it consisted of a set of seven metal electrodes of different materials. We had already used metal electrodes as potentiometric sensors for various applications of food quality control [25][26]. The used device is a PIC18F4550. This kind of PIC is widely used and has the appropriate characteristics to implement the proposed algorithms.

Based on the explained above, the goal of this work is the creation of algorithms to develop Fuzzy ARTMAP simplified artificial neural networks to be implemented in a microcontroller. In order to do this, MATLAB software programs by graphical GUI have been developed to allow the modification of the network characteristics, and check the size of the memory. The analysis was conducted based on data obtained from the measurements of the different floral origin honey and the heat treatments. These data have been obtained with a potentiometric electronic tongue system with the described seven electrodes. In addition, an analysis for the electrodes election has been performed, in order to determine the lower number of electrodes to have similar hit rate values than the analyses with the whole electrode array. In addition, net characteristics are improved since the decrease of inputs reduces the memory size.

2. FUZZY ARTMAP: A BRIEF REVIEW

As explained above, Fuzzy ARTMAP (FAM) network was developed by Carpenter and Grossberg in 1991. FAM is an adaptation of the Adaptive Resonance Theory and Fuzzy Logic Theory. FAM network is a learning rule that of joint form minimizes predictive errors and maximizes the generalization. This is achieved by means of a searching process that increases

the vigilance parameter in the minimal necessary quantity to correct the predictive error.

In short, FAM network is a generalization of ARTMAP, simply substituting the ART modules (ART1) for Fuzzy ART modules that are described by means of operations of fuzzy logic.

FAM operates in two different phases:

- Supervised Phase or Training Phase. During this phase a list of input vectors pairs $\{a, b\}$ is given. Where $\{b\}$ vector is the expected output for the corresponding input vector $\{a\}$.
- Unsupervised Phase or Test Phase. A list of input vectors $\{a\}$ and a group of output vectors $\{b\}$ are obtained in this phase.

As shown in Figure 1, FAM network is formed by two Fuzzy ART networks, called ART_a and ART_b . These two networks are related by an associative map called inter-ART map (F^{ab}). This module is used in order to predict associations between categories and to develop the track rule. Architecture is complemented with a control mode called RESET.

The network processes the input I^a and selects the appropriate category in the layer F_2^a based on the setting of the vigilance parameter ρ_a . The pattern associated with the winning F_2^a category is presented to the mapfield, which is labeled on the diagram as inter ART module.

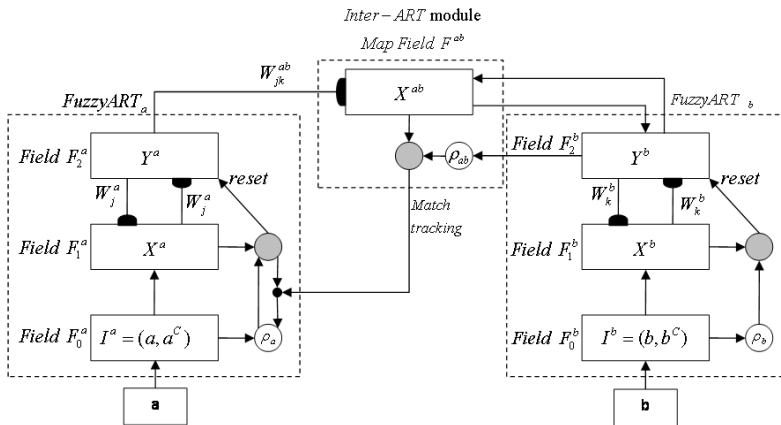


Figure 1. Fuzzy ARTMAP architecture.

Similarly, the paired output vector I^b associated with the input vector is applied to the input of the fuzzy ART_b network. The ART network then determines an appropriate output category for the fuzzy ART_b input. The pattern associated with the winning F_2^b category is also presented to the mapfield. The two patterns are then compared with each other in the mapfield and held up against the inter-ART vigilance parameter (ρ_{ab}). If the match between the fuzzy ART_a and fuzzy ART_b output vector is suitable, then the weights (W_{jk}^{ab}) between the layer F_2^a and the mapfield are adjusted to match the pattern presented by layer F_2^b . Simultaneously, the fuzzy ART_a network resonates and learn its input pattern.

When the patterns at the mapfield do not meet the vigilance criterion, an inter-ART reset is issued. During the inter ART reset, the vigilance parameter (ρ_a) of the fuzzy ART_a network is raised just far enough so that the winning neuron of fuzzy ART_a no longer wins the competition. This causes the fuzzy ART_a network to seek or create a new category in layer F_2^a . This particular feedback ensures that a new category is selected for data that does not fit the current pattern set. By dynamically adjusting the ART_a vigilance (ρ_a), the ARTMAP network ensures that there will be just enough categories created to cover all possible input-output pairs.

The reader is referred to [13][15] for a detailed review of this architecture.

There are several properties that make Fuzzy ARTMAP a promising pattern recognition method for EN systems [16].

Exhibits fast learning of rare events: Many traditional learning strategies use forms of slow learning that average over the occurrence of similar events. Fuzzy ARTMAP can rapidly learn a rare event that predicts different consequences than a cloud of similar events in which it is embedded.

Suitable for non-stationary environments: In a non-stationary environment, traditional algorithms tend to lose the memory of old, but still useful knowledge. Fuzzy ARTMAP contains a self-stabilizing memory that allows for the accumulation of knowledge in response to a non-stationary environment; until the memory capacity is full memory can be chosen arbitrarily large.

Ability to adjust the scale of generalization: In many environments some information may be coarsely defined, whereas other information may be precisely characterized. Fuzzy ARTMAP is able to automatically adjust its scale of generalization to match the morphological variability of the data. It conjointly maximizes generalization and minimizes predictive error using only information that is locally available under incremental learning conditions.

Ability to learn many-to-one relationships: Many-tone learning combines categorization of many exemplars into one category, and labeling of many categories with the same name. Individual recognition categories play the role of hidden units in the back-propagation model. Unlike the back-propagation model, Fuzzy ARTMAP discovers, on its own, the number of categorical ‘hidden units’ that it needs for a specific problem.

Ability to deal with uncertainty: A key element in any measurement system is uncertainty and the fuzzy approach is one way of dealing with it.

Due to its properties, this network has become very widely used in different applications. Some of its most extended applications are in noses and electronic tongues: noses related with food industry[17],[18],[27],[28], noses related with chemical systems[29]-[33], tongues related with food[19],[20], [34]-[37] and tongues related with nervous agents[38]. In addition, it is also applied in biomedicine[39]-[41], bioengineering[42],[43], communications[44],[45], power energy and industry[46]-[49], micro-electronic[50],[51], image processing[52],[53], materials[54][56], security[57], insurances[58] and meteorology[59].

2.1. Simplified Fuzzy ARTMAP

In spite of the several applications of Fuzzy ARTMAP network, its algorithm can be complex and redundant. Due to this, some difficulties can appear in applications with some memory restriction. Most of aforementioned algorithm applications are implemented in a PC. The memory used in a PC is usually big enough in order that the algorithm works properly. Problems appear when the algorithm is used in portable systems because low-cost microcontrollers are used in its fabrication and they usually have limited memory. In this type of systems, it is necessary to look for the algorithms that fit well in the limited memory.

In 1993, Kasuba[21] develops a simplified version of Fuzzy ARTMAP (Simplified Fuzzy ARTMAP or SFAM).). In 2003, Rajasekaran[22] explains the SFAM algorithm based on Kasuba’s paper. That year, Aaron Garret (Jacksonville State University) develops a Matlab toolbox based on SFAM. The GUI presented in this paper has been developed using this toolbox.

The network is a step forward for Fuzzy ARTMAP in reducing the computational overhead and architectural redundancy of Fuzzy. The model employs simple learning equations with a single user selectable parameter and can learn every single training pattern within a small number of training

iterations. SFAM is faster than FAM and it's easier to program. Figure 2 shows the architecture of SFAM. Vectors $\{a\}$, with a number of d features, are introduced in the *Complement Code*. There, they are stretched to double the size by adding their complements. The complement code inputs are called $\{I^a\}$ and they have $2d$ size. These inputs are introduced into the *Input Layer*. Next, weights (W_j) from each of the output category nodes O_N or subclasses are associated with the input layer vectors. This is the reason why they are called *Top-Down Weights*. Aaron Garret designates the *Output Category Layer* as *Mapfield* because of its similar function to the FAM mapfield. The *Category Layer* (C_M) contains the labels for the M categories or classes that the network has to learn for each one of the input vectors. Vigilance Parameter, Match Tracking and Reset are mechanisms of the network that are used during the training.

SFAM network is very sensitive to the absolute magnitudes of the inputs and their fluctuations and it could cause a malfunction of the network. Therefore, it is necessary to normalize the inputs into the same value range. On the other hand, all the inputs values must be into $[0,1]$ range. Complement coding is an input normalization process that preserves the input range. In other words, it shows the presence of a particular feature in the input vector or its absence.

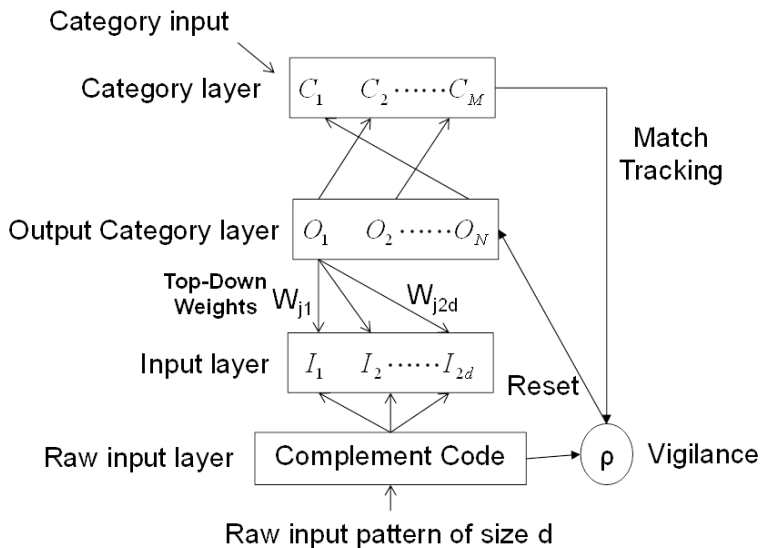


Figure 2. Architecture of SFAM network.

If the given input vector is composed by d features, the complement vector \bar{a} represents the absence of each feature. Therefore, the input vector I^a will be the complement-code input (Equation (1)).

$$I^a = (a, a^c) = (a_1, \dots, a_d, a_1^c, \dots, a_d^c) \quad (1)$$

Whereas $C_i = (1 - a_i)$

An interesting effect of complement coding is the fast normalization of complement-code input (Equation (2)).

$$|I^a| = |(a, a^c)| = \sum_{i=1}^d a_i + (d - \sum_{i=1}^d a_i) = d \quad (2)$$

where the norm $|\cdot|$ is defined by Equation(3).

$$|p| \equiv \sum_{i=1}^M |p_i| \quad (3)$$

The network requires a mechanism to form an activation at the output layer in response to input to the network. When SFAM is presented an input pattern whose complemented-coded representation is I^a , all output nodes became active to some degree. This output activation is denoted as $T_j(I^a)$ of the j^{th} output node and its weights W_j . The function to produce this activation is defined by Equation (4).

$$T_j(I^a) = \frac{|I_j^a \wedge W_j|}{\alpha + |W_j|} \quad (4)$$

Where $W_j = (W_{j1}, W_{j2}, \dots, W_{j2d})$, operator \wedge is defined by Equation (5) and α is called the biasing parameter (α is kept as a small value close to 0). Increasing the value of α will increase the number of subclasses. And where the norm $|\cdot|$ is defined by Equation (3).

$$(p \wedge q)_i \equiv \min(p_i, q_i) \quad (5)$$

The winning output category node is the node with the highest activation function T_j . If more than one T_j is maximal, the output node j with the smallest index is arbitrarily chosen. The category choice is indexed by J , Equation(6).

$$T_j = \max\{T_j: j = 1..N\} \quad (6)$$

The match function is used to compare the complement-coded input features and a particular output node's weights to help determine if learning should occur. The match function is defined by Equation(7) .

$$\frac{|I_j^a \Delta W_j|}{|I^a|} \quad (7)$$

and by Equation (2), the match function is also defined by Equation(8).

$$\frac{|I_j^a \Delta W_j|}{d} \quad (8)$$

When used in conjunction with the vigilance parameter ρ , the match function value states whether the current input is a good enough match to a particular output node (O_j) to be encoded by that node or instead whether a new output node should be formed to encode the input pattern. If the match function value is greater than the vigilance parameter, the network is said to be in a state of resonance (Equation(9))

$$\frac{|I_j^a \Delta W_j|}{d} \geq \rho \quad (9)$$

If the output category node of the winner matches with the category input ($O_j = C_i$), it must update weight vector W_j according to the Equation(10).

$$W_j^{\text{new}} = \beta(I \wedge W_j^{\text{old}}) + (1 - \beta)W_j^{\text{old}} \quad (10)$$

where the learning parameter β (between 0 and 1) determines the speed of network learning; high values of β result in high learning speed (fast-learning mode) while low values causes low learning speed (slow-learning mode). Additionally it contributes robustness to the classification algorithm, especially when it comes to categorizing data that may have some noise in their values (by using slow-learning mode).

If the value of the Vigilance parameter is higher than the Match function, the network is Mismatch Reset. This condition indicates that the node of the output category does not fit sufficiently with the input category ($O_j \neq C_i$). It must temporarily increase the vigilance parameter ρ so as to violate the condition of Equation (9). Set ρ by Equation(11).

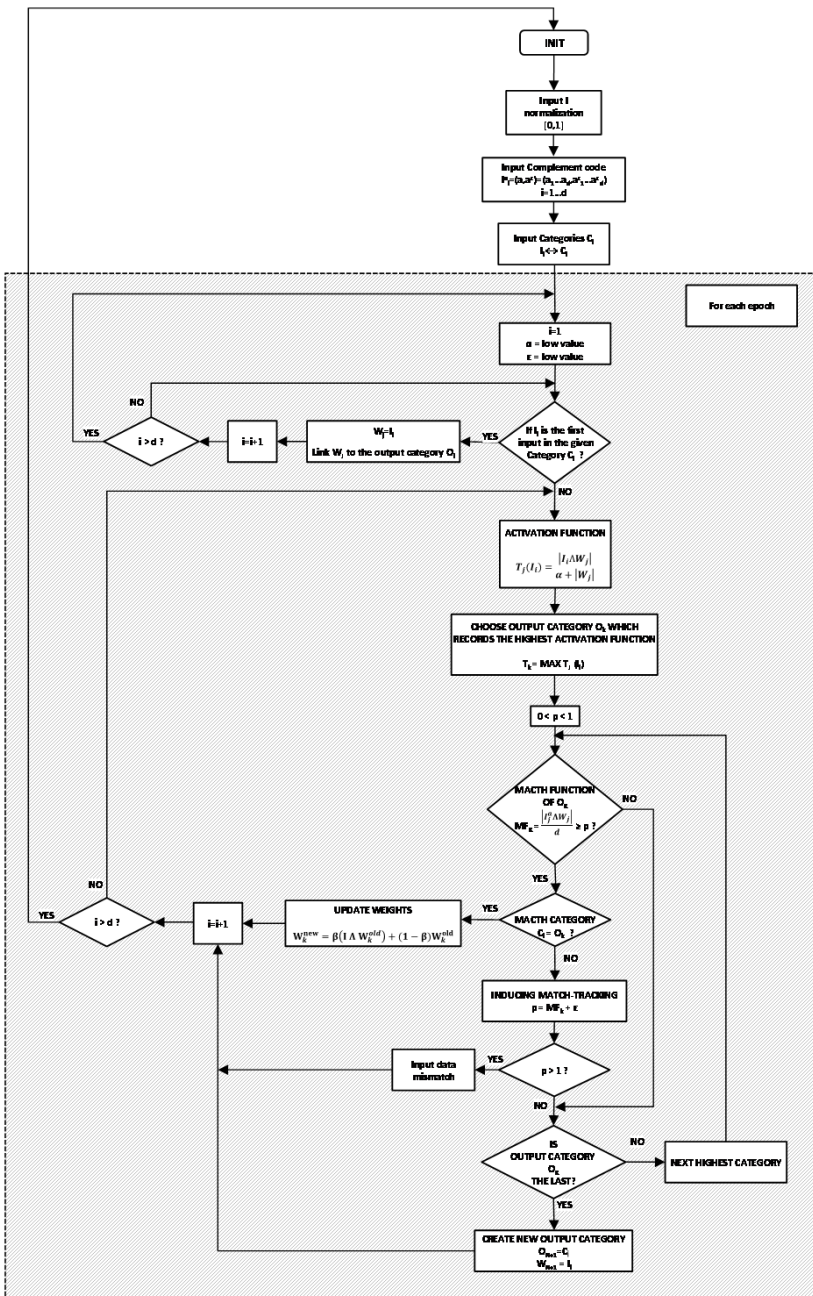


Figure 3. Flow chart of SFAM training algorithm.

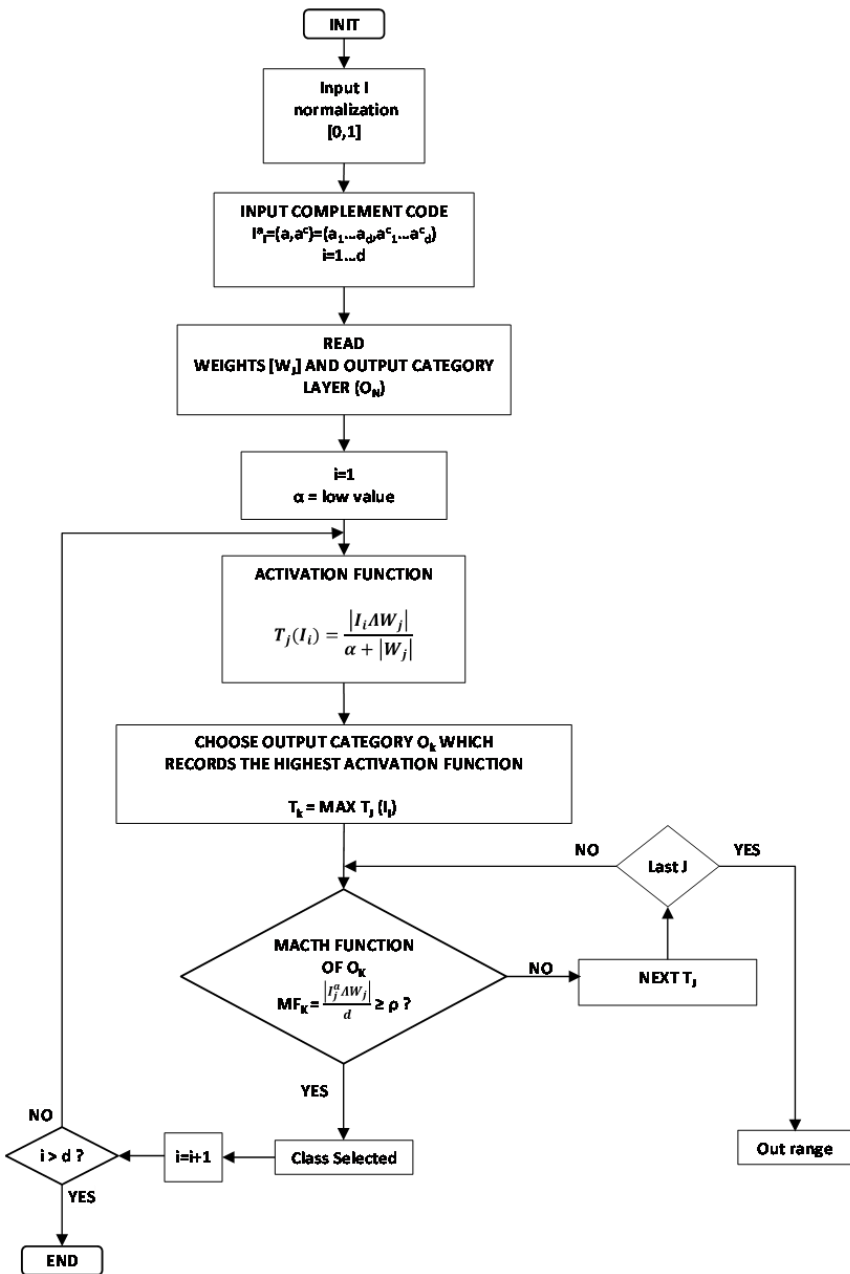


Figure 4. Flow chart of SFAM algorithm in Unsupervised Phase (Test Phase).

$$\rho = \frac{|I_j^a \Delta W_j|}{d} + \epsilon \quad (11)$$

where ϵ is a small positive number, i.e. $\epsilon \approx 0.001$. A large value of ϵ could trigger a false data mismatch alarm, by causing the value of ρ to go beyond. In the cases where some real mismatch exists, larger value of ϵ will increase the error.

SFAM algorithm to training phase can be defined as the flow chart in Figure 3.

Algorithm for the Test Phase (Figure 4) is simpler since it is focused on determining the class of the input vector.

The category choice is indexed by K . This index K points to the subclass in the output category layer or mapfield. If the match function is less than the vigilance criterion a lesser choice function is selected and the resonance is checked again. Finally if there is no choice function whose match function is greater than the vigilance criterion, the input vector is classified as out of range. If there is resonance then the input vector is classified.

In 2003, Vakil-Baghmisheh et al. [60] published a SFAM model different from Kasuba's model pointing some criticism to Kasuba's model. Based on the original SFAM, several versions have been published later, i.e. Probabilistic Simplified Fuzzy ARTMAP (PSFAM)[61]. Another improved version was published by Vuskovic[62] in 2002 and it was called SFAM based on Mahalanobis Distance.

There are more applications of SFAM in image processing[63],[64], face recognition[65]-[67], power[68],[69], biomedicine[62],[70]-[72] and electronic tongues[20].

2.2. Simplified Fuzzy ARTMAP Graphical User Interface

An aim of the present work is to implement a SFAM neural network in a low-cost 8 bit microcontroller. The idea is to develop a portable system that could be applied in different fields of the industry. In our case, it is going to be applied in a Food industry, specifically in Honey production.

Systems based on microcontrollers have a premise that must be considered: the limit in the use of memory. In low-cost systems, the used microcontrollers usually have a limited size of memory. In these cases it is necessary to optimize the data processing algorithms. In this case, the SFAM contributes with a series of advantages: less memory requirements, rapidity

and facility of programming. The information that must be programmed in the microcontroller memory are the Weight Matrix, the Mapfield (*Output Category Layer*) and the maximum and minimum values of the input vector in order to use them in the normalization of these vectors.

One of the problems when working with FAM and also SFAM is the size of the mapfield and the Weight matrix that depends on the values chosen for the parameters β y ρ . The size of the weight matrix and the mapfield grows if the number of inputs data increases. Table I shows the size variation of the weight matrix, RAM memory size and Program memory size used according to the number of training samples (i.e. between 8 and 18 samples, the program memory hardly changes but the RAM memory changes up to 40%). In this case, it would mean that the microcontroller should be changed due to the memory limit was exceeded.

Usually there is not an initial criterion to establish the values of these parameters, and different trainings must be realized changing its value in order to find an ideal recognition rate. The *Output Category Layer* (mapfield) size and the weight matrix size are not usually taken into account when a suitable recognition rate is found because memory size in PC systems is not significant. On the contrary, in microcontroller based systems it is necessary to verify the sizes of these two data and they should be the minimum.

Table I.

Number of training samples	Weight Matrix Size	RAM Memory	Programme Memory
8	12x16	1635	12726
10	12x17	1700	12812
12	12x20	1895	13064
14	12x22	2025	13224
16	12x23	2090	13304
18	12x26	2285	13356

It can be done by training with different β and ρ values and looking for the best recognition rates in order to select the smallest weight matrix and mapfield sizes.

In order to obtain the maximum recognition rate in the classification for the minimum weight matrix and mapfield, a Graphical User Interface (GUI) has been developed in Matlab.

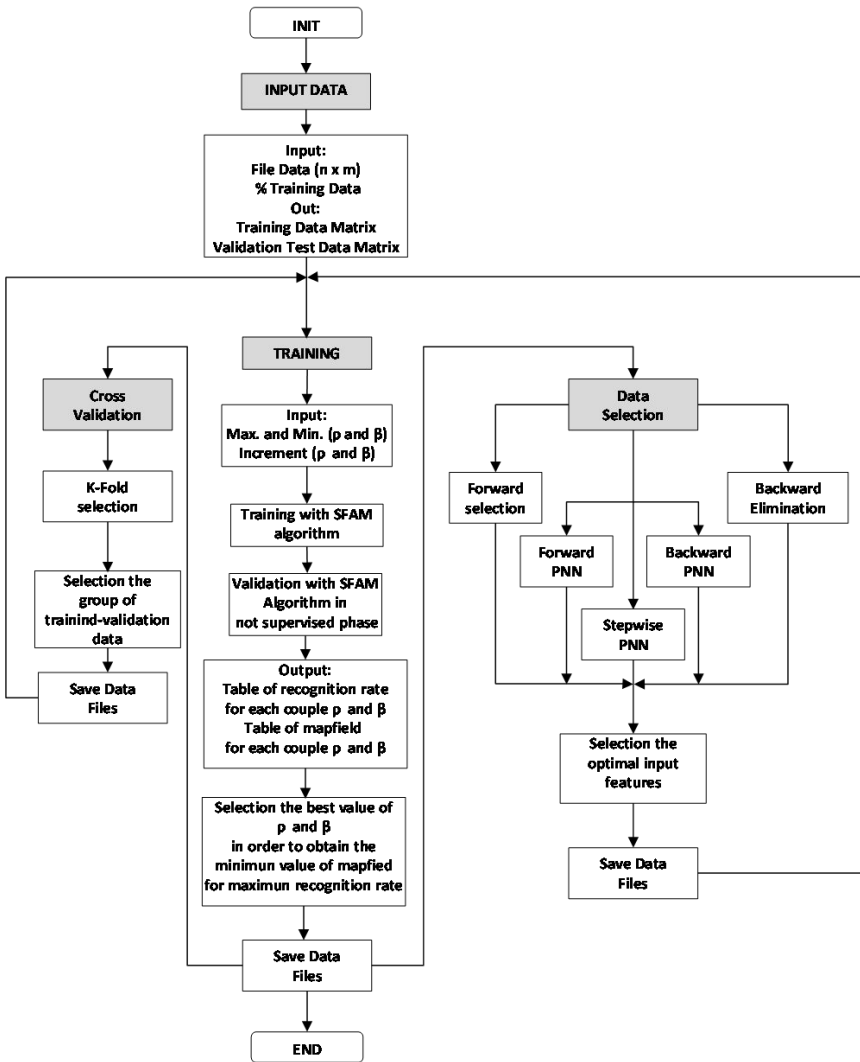


Figure 5. Flow chart of GUI.

It is based on the toolbox developed by Aaron Garret. By means of this GUI, it is possible to carry out the training considering all the possible values for β and p in order to determine the best classifications and look for the one with the smallest mapfield and data weights size. The idea is to improve the recognition rate in order to implement the network in the microcontroller. In

addition, the option of doing a cross validation and a variable selection have been implemented into the GUI. Figure 5 shows the flow chart for the developed GUI.

2.2.1. Input Data

Data are given in $n \times m$ matrix. In this matrix, $n-1$ is the input vector for each sample plus a row for the category layer and m is the number of samples for both training and validation. On the other hand, the percentage of data for training must be specified and introduced in the program.

The program calculates the following parameters attending to the introduced data matrix: number of categories or classes, number of data for validation, number of data for training. It also separates the input vector from the expected outputs (category layer). In the next step, one can choose among selection of variables, cross validation and training.

2.2.2. Training

The goal of this step is to obtain the appropriate β and ρ values to achieve the maximum recognition rate for the minimum mapfield. In order to meet these conditions several of the following programs have been designed and applied: *ARTMAP_parameters*, *ARTMAP_files* y *ARTMAP_robeta*.

ARTMAP_parameters: allow us to configure the initial, final and the increment for parameters β and ρ . Once they have been configured, the training and testing steps are carried out. After these calculations, recognition rates depending on β and ρ tables are shown, as well as mapfields, maximum and minimum input values and the non-classified training inputs.

ARTMAP_files: allows us to select the appropriate information to be stored in .xls or .txt files.

ARTMAP_robeta: in case of having two cases with a maximum recognition rate for a minimum mapfield value, this program is able to determine the sensitivity and specificity of these cases by using Plot Confusion and Plot ROC (Receiver Operating Characteristic). It is also able to determine the optimum case by studying the Area Under Class (AUC).

2.2.3. Cross Validation

Cross validation is used to identify the training-validation data set that achieves the best recognition rate. In a cross validation of K iterations or K -fold cross-validation, the data of the samples are divided into K subsets. One of these subsets is used as a test and the rest ($K-1$) are used as training data.

The cross validation process is repeated for K iterations with each one of the potential test data sets. Finally, the arithmetical mean of the overall results for each iteration is used to get a final result. Attending to this aim, three programs have been developed: *ARTMAP_cross*, *ARTMAP_validation* and *ARTMAP_samples*.

ARTMAP_Cross: this program allows us to introduce the initial, final and increment values for ρ y β parameters in order to carry out the SFAM training and validation.

The d input data are divided into m samples, so that each sample contains one data of each class. If c is the number of classes, the number of samples will be defined by Equation (12). Care must be taken with the m number of samples that must be an integer number and not a prime.

$$m \text{ (samples)} = \frac{d \text{ (data)}}{c \text{ (classes)}} \quad (12)$$

Next, cross validation is done. Then, the program calculates the dividends for the number of samples and it presents them in a dropdown menu for the user to choose the K -iterations or K -fold cross-validation. As written before, the sample data are divided into K subsets. One of the subsets is used as a test data and the rest ($K-1$) are used as training data. The process is repeated for K iterations with each one of the potential training subsets. Next, the program calculates the number of samples and data for training and validation as well as the overall data and it shows them on the screen. At last, initial, final and increment values for both β and ρ are selected. Attending to all these variables, the subprogram calculates the matrices or goals for training and validation.

ARTMAP_validation: This program has been designed to do the cross-validation and perform the iterations selected in the previous subprogram depending on β and ρ parameters. Once the calculation has been done, it shows a series of windows with the results for Mapfield values and recognition rates as function of β and ρ .

ARTMAP_samples: this program allows us to save the obtained data in .xls and/or .txt format.

2.2.4. Data Selection

The objective of every variable selection problem is to find the subset of variables that best explain the class for each pattern. In order to solve the paradigm of variable selection, several methods have been designed: *Forward*

Selection, Backward Elimination, Forward Probabilistic Neural network (PNN), Backward PNN and Stepwise PNN.

Forward Selection: the number of output variables (1 to n variables) is selected by the user using a dropdown menu. Operation is easy: first, we start with a model without variables. Next, the most statistically significant variables are chosen one by one until the maximum predetermined number of variables is reached or some previous condition is satisfied. The main advantages of this method are the following: Easy implementation and short calculation time. The disadvantages are: the number of variables to be chosen is arbitrary and variables cannot be eliminated once they have been chosen.

Backward Elimination: the number of output variables (1 to n variables) is selected by the user using a dropdown menu. This method starts considering all the selected variables and the ones that are not statistically significant are removed. As well as in the method explained before, implementation is easy and the calculation time is short. In this method, the number of variables is selected by the user but variables cannot be removed once they have been selected.

Forward PNN: this method is a *Forward* type one. In this case, a Probabilistic Neural Network has been used for validation and output variables are chosen depending on the recognition rate (98%). In addition, the runtime has been displayed on the screen.

Backward PNN: this method is a *Backward* type one. In this case, a Probabilistic Neural Network has been used for validation and output variables are chosen depending on the recognition rate (98%). In addition, the runtime has been displayed on the screen.

Stepwise selection: this method is a mix of the two methods explained before. First, a Forward selection method is carried out. This allows us to select the variables. Then, a backward elimination method lets us remove variables. The Stepwise selection method is executed as many times as the user wants and next it is validated by a Probabilistic Neural Network. Therefore, the user just has to enter the number of iterations to be done. In this method, output variables are decided in function of the recognition rate (98%) and, as done before, runtime has been included and has been displayed on the screen.

At the end of the study with all these subprograms, a file can be saved containing the new optimized data matrix in order to do a new training.

3. MATERIALS AND METHODS[24]-[73]

3.1. Electrodes

Measurements were carried out by using an electronic measuring system with an array of sensors consisting in a set of metallic electrodes made with different materials. This array has been designed to be immersed in the honey samples with a reference electrode and measure the electrical potential generated spontaneously between each electrode and the reference one. In this specific case, the electrodes act as non-specific sensors because they do not respond to any particular chemical parameter.

In this experience, seven different metallic electrodes were used. Some were pure as gold, silver and copper ones but other electrodes were chemically treated by electrolysis (AgO_2 , CuO_2 , AgCl y Ag_2CO_3) in order to determine if they had different properties from the pure metal ones. Some of the electrodes were repeated in the array. In this case, the average values of their measurements were used. Electrodes were made of 0.8mm in diameter and 5 cm in length wires, connected to a ribbon cable (Figure 6).

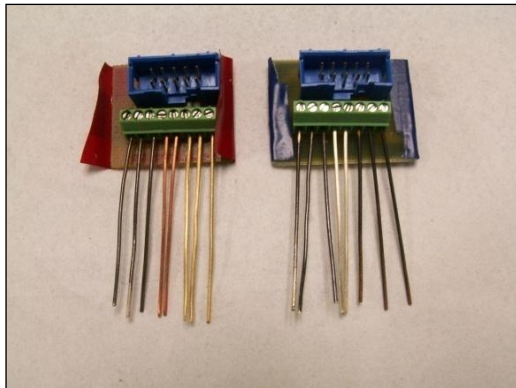


Figure 6. Arrays of different metallic electrodes.

3.2. Electronic System

Two boards with 8 electrodes on each were used. Therefore 16 channels could be measured simultaneously. The external reference electrode employed was an Ag/AgCl device (supplied by CRISON).

Measurements were carried out using an own design portable data logger. The output signals of the multi-electrode were acquired using a 36:1 multiplexer architecture, which was formed by two 18:1 channel MOS analog multiplexers (MAX306, MAXIM) and one 8:1 channel analog multiplexer (MAX308, MAXIM). The selection of each channel in the multiplexer was controlled by the microcontroller.

A precision CMOS quad micro power Operational Amplifier (LMC646, NATIONAL SMC), was connected to the output multiplexer. This operational amplifier (AO) has very high input impedance (ultra low input bias current of less than 16 fA) and hence is suited to the signal impedance generated by the potentiometric multi-electrode.

An analog to digital converter (A/D) (MAX128, MAXIM) has been used because the AD of the microcontroller is merely of 10 bits resolution and it only accepts positive voltage. This A/D has a resolution of 12-bits and can work with unipolar or bipolar input signals. It uses an external or internal reference voltage in order to obtain different full scale ranges. In this case a 2.5V external reference and a bipolar input signal were used. With this configuration the resolution (equivalent to 1 Least Significant Bit) is 1.22 mV. The PIC18F4550 microcontroller gathered the data from the A/D converter using an I2C bus. PIC18F4550 was selected for its low power consumption (sleep mode currents down to 0.1 μ A typical), 32K of memory program and 2K of RAM and USB port. The software for the PIC18F4550 microcontroller has been designed to obtain the average value for each channel. Seven input vectors are calculated using the 16 channels of data. These seven input vectors correspond to the seven types of electrodes (there were some of them with the same material). The process of measurement has been divided in two stages: the training period and the test period. In the training period, the data were sent to the PC via an RS232 serial communications link in order to use them in the training algorithm with MATLAB® R2010b. The acquisition software was developed using Visual Basic® 6.0 and Microsoft Excel® 2003 software. In the test period, the data were measured and they were stored directly into the microcontroller in order to be used in the embedded neural network. A block diagram of the measurement system is shown in Figure 7.

The training stage is performed with some of the available measures. At this stage the network categories are set out. The data from electrodes for each measurement are applied as an input vector. With these data the coefficients of the algorithm that configures the network are calculated. In the verification stage, the data from new measures are applied to the inputs, checking whether the output of the active network is correct or not.

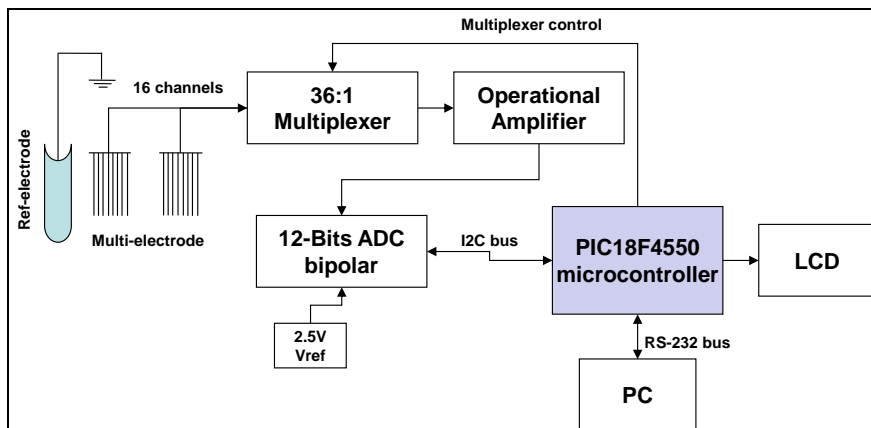


Figure 7. Diagram of the measurement system.

The GUI Matlab 2010b® running on a PC computer has been used to train the networks. The computer to be used is determined by its computing power and ease of implementing the algorithms of the neural networks. By contrast, the verification stage is performed entirely in a microcontroller. To this end, the results obtained in the training stage are used as the coefficients of the algorithms that are incorporated into the microcontroller program. Through this way of working, once the training stage has been accomplished, the developed system can work independently of a PC. This is one of the key features of the equipment presented in this paper.

3.3. Measurement Process

In order to determine the repetitiveness of the measuring system, four measurements were carried out in each honey sample. In this way, the total number of measurements was 48 (4 floral origins x 3 physical treatments x 4 repetitions). The electrochemical response to the voltammetric analysis changes depending on the measuring electrode and the specific honey sample. This variation is not decisive because there are specific cases in which the response of an electrode changes with some kind of sample but it doesn't change with some others and vice versa. It's also important to consider that voltammetric measurements with such complex chemical samples as ours are not usually very repetitive.

The sampling rate for the 16 channels was one electrode every 100 ms in periods of 10 s for approximately 5 min or until the signal became established because the electrochemical equilibrium was reached between the electrode and the sample.

Once the equilibrium was reached, the values of the last ten samples were taken in order to reduce the effect of electrical noise. In this way, a single value was obtained for each one of the seven working electrodes.

3.4. Data Analysis

In order to obtain quantitative and complete conclusions from the measurement results, we decided to work with SFAM networks. In our specific case, two SFAM networks were used.

The first one tried to determine the floral origin of the honey samples so the network has four outputs (Figure 8-1): one output for each honey group. The second neural network consists of three outputs, one for each physical treatment (Figure 8-2).

With the initial data, a matrix of 7 columns and 48 rows was created. Table II shows all the data for each analyzed sample: Citrus (C), Rosemary (R), Polyfloral (PF) and Forest (F), and each one of the treatments: Raw (R), liquid (L) and Pasteurized (P).

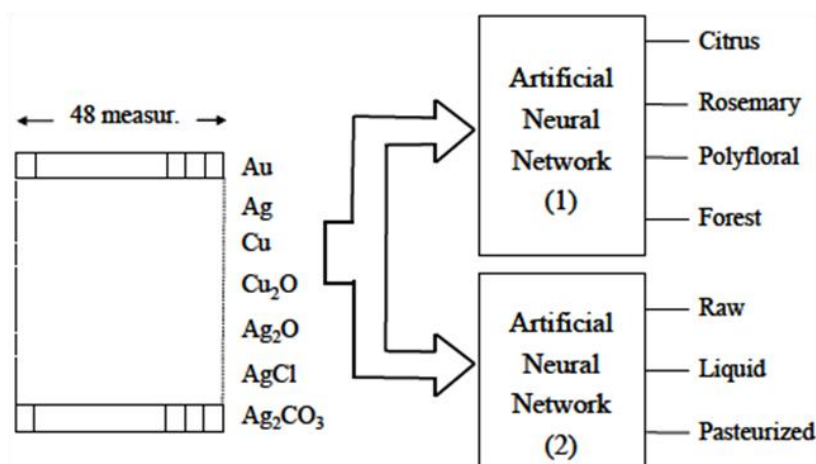


Figure 8. Data classification by using a SFAM network.

Table II. Training Data

Honey	Au	Ag	Cu	Ag ₂ O	Cu ₂ O	AgCl	Ag ₂ CO ₃	Class	Sample
C-R	0.1108	0.1022	-0.0949	0.1164	-0.0422	0.0503	0.0761	1	M1
R-R	0.0768	0.0959	-0.1065	0.1124	-0.0771	0.0249	0.0927	2	
PF-R	0.0240	0.0810	-0.1324	0.1048	-0.0978	0.0272	0.0715	3	
F-R	0.0277	0.0749	-0.1528	0.0934	-0.0976	0.0268	0.0922	4	
C-R	0.1150	0.1355	-0.0522	0.1405	-0.0256	0.0499	0.1231	1	M2
R-R	0.0870	0.0884	-0.0798	0.1080	-0.0391	0.0198	0.0861	2	
PF-R	0.0168	0.0880	-0.1040	0.1040	-0.1001	0.0326	0.0849	3	
F-R	0.0109	0.0547	-0.1415	0.0855	-0.1005	0.0186	0.0676	4	
C-R	0.1000	0.1120	-0.0500	0.1240	-0.0126	0.0339	0.0859	1	M3
R-R	0.0762	0.0939	-0.1020	0.1147	-0.0580	0.0259	0.1030	2	
PF-R	0.0248	0.0984	-0.0871	0.1097	-0.0835	0.0317	0.0951	3	
F-R	0.0099	0.0728	-0.1522	0.1058	-0.1075	0.0215	0.0774	4	
C-L	0.1187	0.1232	-0.1066	0.1450	-0.0674	0.0735	0.0937	1	M4
RL	0.1076	0.0819	-0.1228	0.0877	-0.0654	0.0580	0.0727	2	
PF-L	0.0173	0.0762	-0.0912	0.1022	-0.0555	0.0118	0.0742	3	
F-L	0.0325	0.0581	-0.1635	0.0917	-0.1422	0.0093	0.0899	4	
C-L	0.1194	0.1075	-0.1006	0.1301	-0.0146	0.0594	0.0990	1	M5
R-L	0.1053	0.0837	-0.1362	0.0823	-0.0880	0.0286	0.0504	2	
PF-L	0.0230	0.0897	-0.1311	0.1114	-0.0767	0.0309	0.0822	3	
F-L	0.0142	0.0710	-0.1497	0.0937	-0.0972	0.0316	0.0468	4	
C-L	0.0850	0.1119	-0.0979	0.1282	-0.0415	0.0461	0.1158	1	M6
R-L	0.0846	0.0963	-0.1092	0.0945	-0.0756	0.0424	0.0781	2	
PF-L	0.0329	0.0818	-0.1289	0.1028	-0.0900	0.0070	0.0751	3	
F-L	0.0274	0.0784	-0.1429	0.1140	-0.1422	-0.0070	0.0615	4	
C-P	0.1211	0.1153	-0.0536	0.1310	-0.0297	0.0419	0.0868	1	M7
R-P	0.0740	0.0858	-0.0827	0.0953	-0.0431	0.0403	0.0431	2	
PF-P	0.0417	0.0982	-0.0848	0.1150	-0.0603	0.0148	0.0926	3	
F-P	-0.0003	0.0617	-0.1454	0.0723	-0.0974	-0.0008	0.0442	4	
C-P	0.1307	0.1314	-0.0665	0.1359	-0.0305	0.0387	0.1162	1	M8
R-P	0.0819	0.0982	-0.1181	0.1200	-0.0676	0.0202	0.0852	2	
PF-P	0.0489	0.1024	-0.0988	0.1161	-0.0598	0.0298	0.1009	3	
F-P	-0.0124	0.0721	-0.1688	0.0902	-0.0981	-0.0016	0.0772	4	

3.4.1. Training and Validation with GUI

Floral Origin Network

Input and output data files as well as the Validation Percentage are defined in the main window of the GUI software. In our specific case, the Validation Percentage was 87.5% (28 data for training and 4 for validation). The software chooses the training and validation data for the expected output and shows them on the screen.

The number of categories is also shown. In the end, binary training and test matrices are obtained. They are necessary to get the Plot ROC (Receiver Operating Characteristic) and the Plot confusion.

Next, β and ρ data are introduced in the SFAM in order to do the training. In the beginning a wide range must be given to these parameters due to the fact that we have to determine who the best ones are in order to obtain the optimal recognition rate for the lower validation mapfield. Table III shows the results for the two variables scanning. Next, when running the model with ρ between 0.1 and 0.5 and β between 0.4 and 1, 100% recognition rates are obtained with a mapfield size of 1 x 6.

Next, a GUI cross validation is done. In this specific case, the chosen order is 4. The order 4 cross validation is equal to use 75% of the data for training and 25% of the data to do the test (The complete data matrix is divided into two groups, 24 data for training and 8 for validation). This process is repeated four times in order to obtain the data with the best recognition rate and the smallest mapfield size. The number of samples and the data for training, validation/test and the total data are calculated and shown in the screen.

Target matrices must be generated in order to carry out training and validation and obtain both hit rates and their corresponding Mapfield as functions of β and ρ .

In our case, the software runs the training with the SFAM network by using the whole range for β and ρ . The best result is obtained with samples M1, M2 and M5 to M8 for training, and M3 and M4 for validation. In this case, 100% recognition rates and very small mapfield sizes (1x4) are obtained in the zone with ρ in the range of [0.1 – 0.3] and β in the range of [0.7 – 0.8]. See Table IV.

Finally, the study to select the variables can be done by GUI. A number is given to each electrode according with Table V.

Table III. Recognition rates and mapfield sizes depending on β and ρ for 87,5% training data rate

Mapfield (1xO)											Recognition rate %									
ρ / β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	100	100	7	6	6	6	6	6	6	6	100	75	100	100	100	100	100	100	100	100
0.2	100	100	7	6	6	6	6	6	6	6	100	75	100	100	100	100	100	100	100	100
0.3	100	100	7	6	6	6	6	6	6	6	100	75	100	100	100	100	100	100	100	100
0.4	100	100	7	6	6	6	6	6	6	6	100	75	100	100	100	100	100	100	100	100
0.5	100	100	7	6	6	6	6	6	6	6	100	75	100	100	100	100	100	100	100	100
0.6	100	100	9	7	8	9	9	7	7	7	75	75	75	50	75	50	75	75	100	100
0.7	100	100	15	14	14	15	14	14	9	8	25	25	25	50	50	25	25	75	25	25
0.8	61	100	28	21	24	21	19	20	15	12	75	25	50	75	50	50	50	75	50	50
0.9	100	100	31	26	27	27	26	24	23	21	25	25	25	25	50	25	50	50	25	25

Table IV. Recognition rates and mapfield depending on β and ρ for validation with samples M3 and M4

Mapfield (1xO)											Recognition rate %									
ρ/β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	17	100	6	5	5	5	4	4	6	6	100	75	100	100	100	100	100	100	100	100
0.2	17	100	6	5	5	5	4	4	6	6	100	75	100	100	100	100	100	100	100	100
0.3	17	100	6	5	5	5	4	4	6	6	100	75	100	100	100	100	100	100	100	100
0.4	17	100	6	5	5	5	4	4	6	6	62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5
0.5	17	100	6	5	5	5	4	4	6	6	62.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5	87.5
0.6	37	100	10	10	9	9	7	8	7	7	75	75	75	75	62.5	75	75	75	87.5	87.5
0.7	31	100	14	14	14	14	13	13	10	8	25	37.5	62.5	50	37.5	50	50	50	50	75
0.8	100	100	20	19	18	21	18	16	15	13	37.5	37.5	75	37.5	50	62.5	62.5	75	50	75
0.9	100	100	23	23	23	23	22	22	22	21	37.5	50	25	37.5	50	37.5	25	25	25	37.5

Table V. Number given to each electrode

Electrode	Au	Ag	Cu	Ag ₂ O	Cu ₂ O	AgCl	Ag ₂ CO ₃
Variable number	1	2	3	4	5	6	7

The obtained results by different methods are shown in Table VI.

Table VI. Data obtained

Method	Variables selected	Rate
<i>Forward</i> (3 variables)	1 3 6	-----
<i>Forward</i> (4 variables)	1 2 3 6	-----
<i>Backward</i> (3 variables)	1 5 6	-----
<i>Backward</i> (4 variables)	1 3 5 6	-----
<i>Forward PNN</i>	1 2 3	96,875%
<i>Backward PNN</i>	1 2 3	96,875%
<i>Stepwise PNN</i>	1 2 3 5	100%

**Table VII. Recognition rates and mapfields as a function of ρ y β .
Validation with samples M3 y M4 and Au, Ag, Cu and Cu₂O electrodes**

	Validation with M3 y M4						Validation with M8					
	Recognition rate %			Mapfield (1xO)			Recognition Rate %			Mapfield (1xO)		
ρ/β	0.7	0.75	0.8	0.7	0.75	0.8	0.7	0.75	0.8	0.7	0.75	0.8
0.1	100	100	100	6	6	6	100	100	100	4	4	4
0.15	100	100	100	6	6	6	100	100	100	4	4	4
0.2	100	100	100	6	6	6	100	100	100	4	4	4
0.25	100	100	100	6	6	6	100	100	100	4	4	4
0.3	100	100	100	6	6	6	100	100	100	4	4	4

Cross validation with 7 electrodes has a 100% recognition rate when validating with samples M3 and M4, with a mapfield of 1x4. A 100% recognition rate has also been obtained with the Stepwise PPN method when working with Au, Ag, Cu and Cu₂O electrodes. In this way, our decision was to combine the training with samples M1, M2 and M5 to M8, and validating/testing with samples M3 and M by using only variables {1,2,3,5}; then, the values for the new training are [0.1 - 0.3] for ρ and [0.7 – 0.8] for β with increments of 0.05 for both parameters.

Results are shown on the left side of Table VII. As shown, 100% recognition rates have been obtained in all cases with a very small mapfield size. If we compare these results with the ones obtained for the same values for ρ and β but with only one validation sample (results shown at the right side of Table VII), it can be seen that there are the same successful 100% recognition rates but smaller mapfield sizes. This could induce us to believe that these results are better but not in this case because the number of input variables is bigger (7 vs 4).

In this way, values for ρ and β parameters can be chosen (E.g. $\rho=0.3$ and $\beta=0.8$). With these values, training is done to obtain the definitive values of the weight matrix (Table VIII), the mapfield (Table IX), and maximum/minimum input values (Table X) in order to program the microcontroller.

Table VIII. Weight Matrixs (W_{ij})

0.651145	0.567176	0.130534	0.000000	0.185496	0.720916
0.000000	0.176336	0.679389	0.749618	0.746565	0.479084
0.619296	0.354628	0.434159	0.000000	0.342894	0.796610
0.000000	0.432855	0.430248	0.691004	0.646675	0.403390
0.501322	0.039648	0.475771	0.000000	0.274009	0.662379
0.000000	0.262555	0.306608	0.806167	0.695154	0.537621
0.577160	0.513889	0.324846	0.000000	0.342593	0.821605
0.000000	0.204475	0.368056	0.652778	0.597222	0.378395

Table IX. Mapfield

1	2	3	4	3	1
---	---	---	---	---	---

Table X. Maximum and minimum

0.1307	-0.0003
0.1314	0.0547
-0.05	-0.1635
-0.0126	-0.1422

GUI allows us to study the best case if there were different cases with the same recognition rate and minimum mapfield value. In our specific case there is no sense for this study because 100% recognition rate has been obtained but, attending to the interest of this section, two case studies are proposed:

- Case 1: 83.35% recognition rate, 1x26 mapfield, $\rho=0.75$ and $\beta=0.6$
- Case 2: 83.35% recognition rate, 1x28 mapfield, $\rho=0.80$ and $\beta=0.7$

In these cases, mapfields are different and it is easy to believe that the first case is the best one due to its smaller mapfield size but specificity and sensitivity for each case can be obtained by studying the AUC (Area Under Curve) by means of a ROC graphic. GUI calculates the different AUC values for each class and it gives the average AUC values for each class (Table XI). In comparison, it can be seen that case 2 is the best one because sensitivity and specificity are optimal due to its minimum AUC values and overall area.

Table XI. AUC for two different ρ and β values

	Area 1	Area 2	Area 3	Area 4	Total	ρ	β
Case 1	0.889	0.667	1.000	1.000	3.556	0.750	0.600
Case 2	0.944	0.833	1.000	0.833	3.611	0.800	0.700

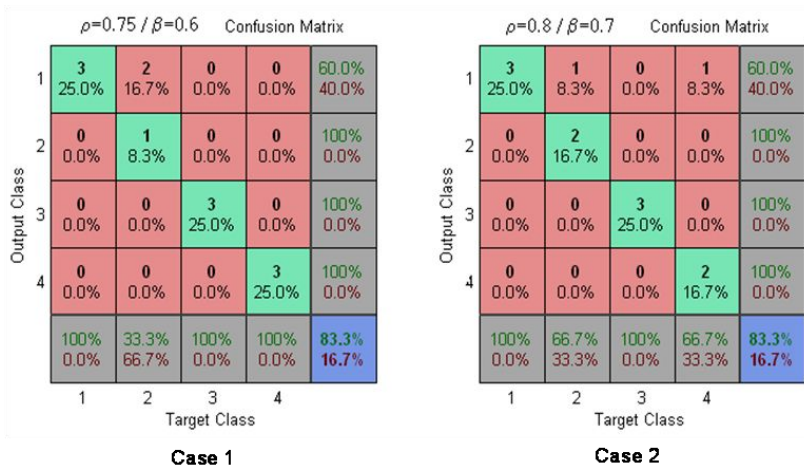


Figure 9. PlotConfusions for the studied cases.

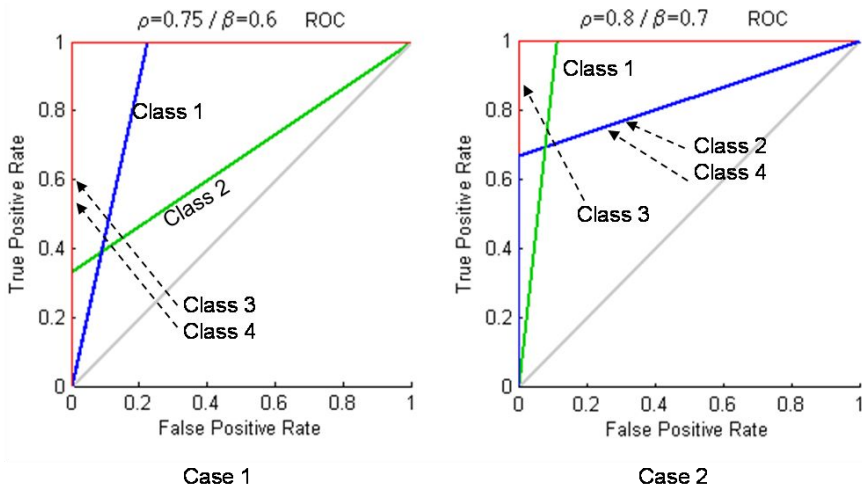


Figure 10. ROC curves.

Figure 9 shows Plot Confusions for the studied cases. It can be observed that case 1 has 2 false positives in class 1 in a total of 10 data ($fpr=0.2$). For Class 2, there is one right in a total of 3 data ($tpr=0.33$). Finally, Classes 3 and 4 are correctly predicted. Figure 10 shows the ROC curve and it can be seen the AUC for each classes. In case 2, 3 data have been correctly predicted for Class 1 but there are 2 false positives in Classes 2 and 4 ($fpr=0.1$). In addition, there are 2 right ones in a total of 3 data for Classes 2 and 4 ($tpr=0.66$). Finally, Class 3 is correctly predicted.

Physical Treatment Network

When developing the network to classify samples by physical treatment, a maximum of 83.3% recognition rate has been obtained for $p=0.7$ and $\beta=0.3$, with a 1×15 mapfield and using samples M1 and M2 for validation/test. It means that there are nearly no groups in the studied data. This fact suggests that thermal treatments don't affect the honey intrinsic properties.

3.5. Implementation of SFAM in the Microcontroller

The embedded system is built around a Microchip PIC18F4550 microcontroller. The PIC18F4550 is a PIC18/8-bit family microcontroller and has 2KB of RAM and 32KB of reprogrammable flash memory.

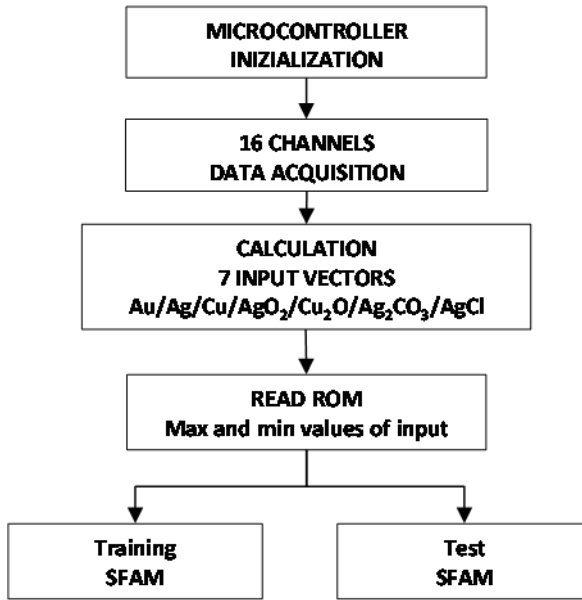


Figure 11. Routines of programme.

The software was coded in C language for the microcontroller and consists of two main routines (Figure 11):

- Data acquisition system where the microcontroller reads the data from the A/D converter and processes them in order to obtain the average of each channel.
- Implementation of the SFAM (Training and Test algorithm).

The algorithm shown in (Figure 4) is used in the microcontroller in order to run the Test phase. In this routine the seven input vectors I^a are calculated using the 16 channels. The data from 16 channels are acquired and they are also normalized to set their range to $[0, 1]$ using the same function of MATLAB®, Equation (14).

$$I = \frac{(Y_{\max} - Y_{\min})(I^a - I_{\min})}{(I_{\max} - I_{\min})} + Y_{\min} \quad (14)$$

where I^a is the input value, I is the normalized input value, Y_{\max} and Y_{\min} are maximum and minimum values respectively of interval $[0,1]$ and finally I_{\max} and I_{\min} are the maximum and minimum values of inputs obtained during the

training period. The I_{\max} and I_{\min} values can be change by the microcontroller algorithm depending on the new inputs.

In order to preserve the amplitude information the data is complemented by using Equation (2). For each input I , the choice function is defined by Equation (4). The category choice is indexed by K (Equation (6)). This index K points to the subclass in the output category layer or mapfield. If the match function is less than the vigilance criterion a lesser choice function is selected and the resonance is checked again. Finally if there is no choice function whose match function is greater than the vigilance criterion, the input vector is classified as out of range. If there is resonance then the input vector is classified. The class is displayed on the LCD panel.

This routine was coded in the C language (CCS C) and was converted to HEX code using a cross compiler. The HEX file is downloaded into the flash memory of the microcontroller. The SFAM neural network has been programmed in 14.745 bytes of program memory (45% ROM) and 1.820 bytes of data memory (88.8% RAM).

4. RESULTS AND DISCUSSION

During the test phase, four more sampling campaigns were carried out. The obtained results by the microcontroller are shown in Table XII.

Table XII. Training Data

Honey	Au	Ag	Cu	Ag ₂ O	Cu ₂ O	AgCl	Ag ₂ CO ₃	Class	Sample
C-P	0.1105	0.1287	-0.0578	0.1388	-0.0460	0.0367	0.0946	1	M9
R-P	0.0902	0.0952	-0.1590	0.1030	-0.0737	0.0049	0.0725	2	
PF-P	0.0351	0.0893	-0.1095	0.1033	-0.0789	0.0226	0.0894	3	
F-P	-0.0085	0.0640	-0.1879	0.0927	-0.1308	-0.0022	0.0447	4	
C-R	0.0872	0.1112	-0.0553	0.1328	-0.0196	0.0246	0.1084	1	M10
R-R	0.0870	0.1053	-0.1041	0.1228	-0.0634	0.0345	0.1041	2	
PF-R	0.0256	0.0736	-0.0733	0.0956	-0.0589	0.0015	0.0724	3	
F-R	0.0213	0.0920	-0.1539	0.1095	-0.1123	0.0276	0.0879	4	
C-L	0.0969	0.1194	-0.0934	0.1377	-0.0490	0.0463	0.1051	1	M11
R-L	0.0761	0.0986	-0.0790	0.1206	-0.0401	0.0212	0.1052	2	
PF-L	0.0286	0.0823	-0.1145	0.1079	-0.0972	0.0102	0.0767	3	
F-L	0.0029	0.0666	-0.1426	0.0815	-0.1091	-0.0061	0.0693	4	
C-P	0.1217	0.1367	-0.0760	0.1400	-0.0456	0.0374	0.0835	1	M12
R-P	0.1016	0.1132	-0.1341	0.1263	-0.0706	0.0347	0.1008	2	
PF-P	0.0279	0.0925	-0.1077	0.1076	-0.0810	0.0167	0.0923	3	
B-P	-0.0077	0.0317	-0.1681	0.0648	-0.1440	-0.0224	0.0335	4	

Table XIII shows the obtained results by the programmed microcontroller in the SFAM test phase for $\rho=0.3$ and 7 electrodes. Figure 12 and Figure 13 show the confusion matrix and the Receiver Operating Characteristic (ROC) for fuzzy ARTMAP, it is observed a recognition rate of 68.8% in this case.

Table XIII. Outputs obtained by the Microcontroller with 7 electrodes

Sample	Class				Rate
	1	2	3	4	
M9	1	1	3	4	75%
M10	1	1	3	3	50%
M11	1	1	3	4	75%
M12	1	1	3	4	75%

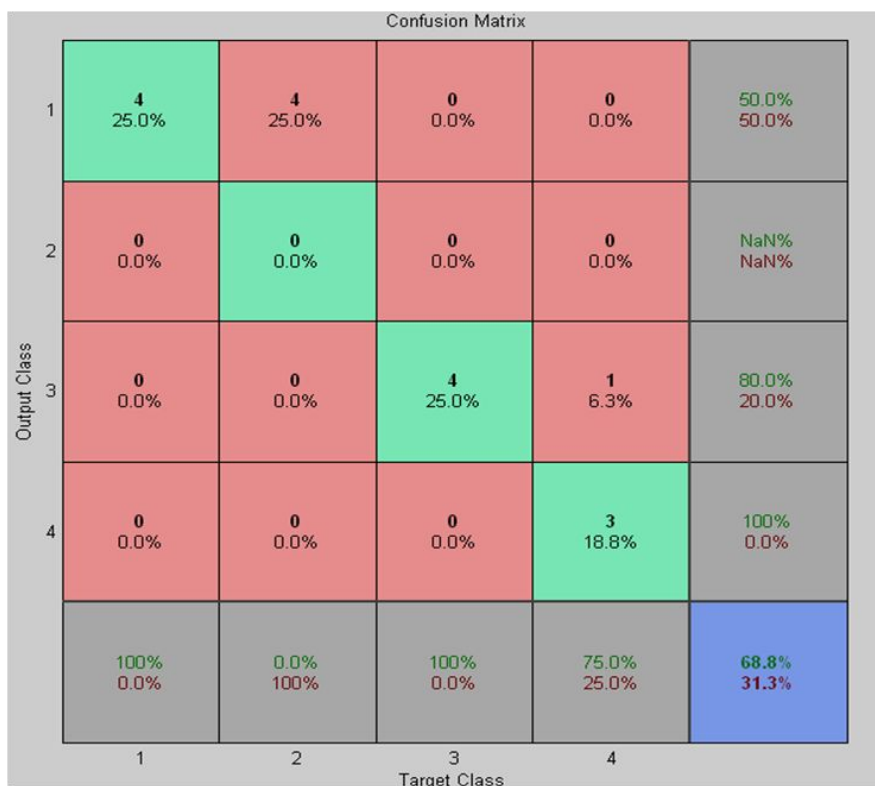


Figure 12. Plot confusion.

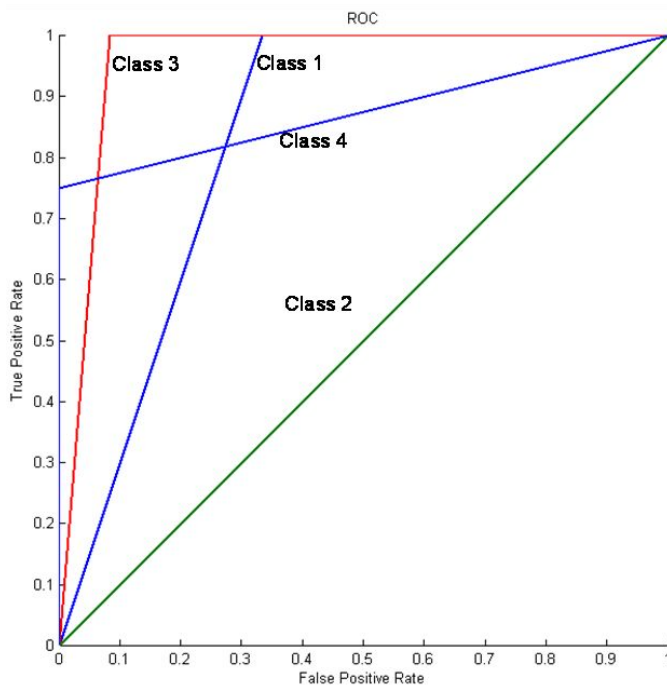


Figure 13. ROC.

Table XIV shows the results when working only with 4 electrodes. Figure 14 and Figure 15 show the confusion matrix and the Receiver Operating Characteristic (ROC) for fuzzy ARTMAP, it is observed a recognition rate of 75% in this case. The Microcontroller has obtained a higher recognition rate when working with a lower number of input variables.

Thanks to the implementation of the SFAM training algorithm into the microcontroller, if the correct sample classification is known, it is possible to use new vectors to do an “in situ” training.

Table XIV. Outputs obtained by the Microcontroller with 4 electrodes

Sampled	Class				rate
	1	2	3	4	
M9	1	2	3	4	100%
M10	1	1	3	3	50%
M11	1	1	3	4	75%
M12	1	1	3	4	75%

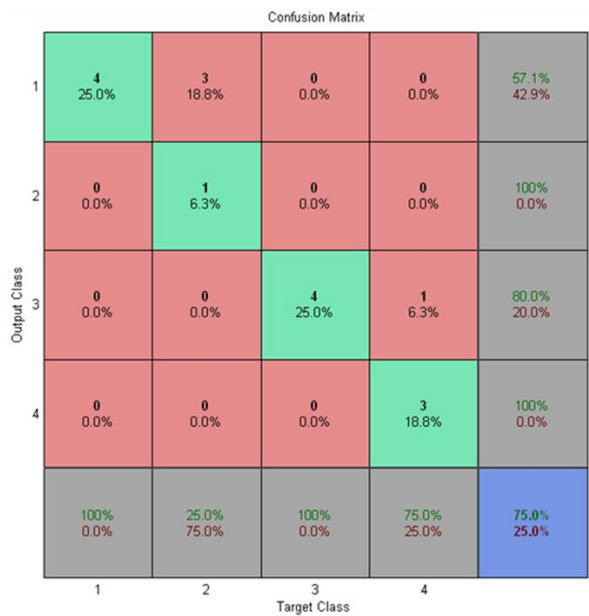


Figure 14. Plot confusion.

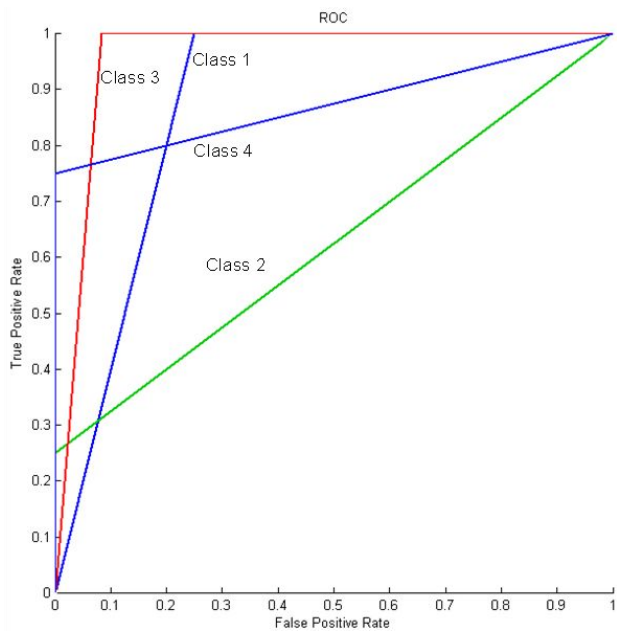


Figure 15. ROC.

Table XV.

Samples		Number of electrodes	
Training	Validation	4	7
8 Samples	2 (M11,M12)	87.5%	75%
9 Samples	1 (M12)	100%	75%

Data introduced in the microcontroller are obtained by training the model with samples M1, M2 and M5 to M8. Next, training is done with 2 or 3 new samples (M9-M10 o M9-M10-M11). The rest will be used to validate the model. β and ρ parameters are kept at 0.8 and 0.3 respectively. Finally, Table XV shows the results and it's seen that the recognition rate has been improved from 75% up to 87.5% when working with 4 electrodes and 2 validation samples. When working with 7 electrodes, the recognition rate was increased from 68.85% up to 75%.

CONCLUSION

A GUI for Matlab has been developed in order to optimize the design parameters of an SFAM algorithm in a microcontroller. The classification SFAM algorithm has been implemented both in training and in its not supervised phase in order to classify honeys. The implementation has been carried out in a portable system based on an 8-bit microcontroller. With the information obtained in the experimental phase, the SFAM has been trained in the microcontroller to obtain the best implementation parameters. The recognition rate in the training phase has been 100 %. Likewise, a simplification of 7 input variables into 4 has been carried out. With the implemented network in the microcontroller, a test with new samples has been made achieving a recognition rate of 68.8 %. This rate has been increased up to 87.5 % by using a part of the information of the test phase as new information for training in the microcontroller.

ACKNOWLEDGMENTS

Financial support from the Spanish Government project MAT2009-14564-C04-02.

REFERENCES

- [1] Legin, A. Rudnitskaya, Y. Vlasov, Electronic Tongues: Sensors, Systems, Applications, *Sensors Update* 10 (2002) 143–188.
- [2] J. Gallardo, S. Alegret, M. del Valle, Application of a potentiometric electronic tongue as a classification tool in food analysis, *Talanta*, 66, June 2005, 1303-1309.
- [3] F. Winquist, P. Wide, I. Lundström, electronic tongue based on voltammetry, *Analytica Chimica Acta*, 357, December 1997, 21-31.
- [4] R. Masot, M. Alcañiz, A. Fuentes, F. C. Schmidt, J. M. Barat, L. Gil, D. Baigts, R. Martínez-Máñez, J. Soto. Design of a low-cost non-destructive system for punctual measurements of salt levels in food products using impedance spectroscopy. *Sensors and Actuators A: Physical*, 158, March 2010, 217-223.
- [5] M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [6] S. Grossberg, Adaptive pattern classification and universal recoding, II: feedback, expectation, olfaction and illusions, *Biological Cybernetics* 23 (1976) 187–202.
- [7] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self organizing neural pattern recognition machine, *Computer Vision, Graphics and Image Processing* 37(1987) 54–115.
- [8] G.A. Carpenter, S. Grossberg, ART2: Self-organization of stable category recognition codes for analog input patterns, *Applied Optics* 26(23) (1987) 4919–4930.
- [9] G.A. Carpenter, S. Grossberg, ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architecture, *Neural Networks* 3 (1990) 129–152.
- [10] G.A. Carpenter, S. Grossberg, J.H. Reynolds, ARTMAP: Supervised real-time learning and classification of non-stationary data by a self-organizing neural network, *Neural Networks* 4 (1991) 565–588.
- [11] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks* 4(1991) 759–771.
- [12] J. Huang, M. Georciopoulos, G. Heileman, Fuzzy ART Properties. *Neural Networks*, 8 (2) (1995) 203-213.
- [13] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, Fuzzy ARTMAP: A neural network architecture for incremental

- supervised learning of analog multidimensional maps, *IEEE Trans. On Neural Networks* 3(5) (1992) 698–713.
- [14] G.A. Carpenter, S. Grossberg, D.B. Rosen, ART2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition, *Neural Networks* 4 (1991) 493–504.
- [15] G.A. Carpenter, S. Grossberg, A self-organizing neural network for supervised learning, recognition, and prediction, *IEEE Communications Mag.* (1992) 38–49.
- [16] E. Llobet, E.L. Hines, J.W. Gardner, P.N. Bartlett, T.T. Mottram, Fuzzy ARTMAP based electronic nose data analysis, *Sensors and Actuators B-Chemical* 61(1-3) (1999) 183-190.
- [17] J. Brezmes, P. Cabre, S. Rojo, E. Llobet, X. Vilanova, X. Correig, Discrimination between different samples of olive oil using variable selection techniques and modified fuzzy artmap neural networks, *Sensors Journal IEEE* , 5(3) (2005) 463- 470.
- [18] L. Gil, J.M. Barat, D. Baigts, R. Martínez-Máñez, J. Soto, E. Garcia-Breijo, E. Llobet, A potentiometric electronic tongue to monitor meat freshness, *IEEE International Symposium on Industrial Electronics*, (2010) 390-395.
- [19] L. Gil, J. M. Barat, D. Baigts, R. Martínez-Máñez, J. Soto, E. Garcia-Breijo, M-C. Aristoy, F. Toldrá, E. Llobet, Monitoring of physical–chemical and microbiological changes in fresh pork meat under cold storage by means of a potentiometric electronic tongue, *Food Chemistry*, 126 (3) (2011) 1261-1268.
- [20] E. Garcia-Breijo, J. Atkinson, L. Gil-Sanchez, R. Masot, J. Ibañez, J. Garrigues, M. Glanc, N. Laguarda-Miro, C. Olguin, A comparison study of pattern recognition algorithms implemented on a microcontroller for use in an electronic tongue for monitoring drinking waters, *Sensors and Actuators A: Physical*, 172(2) (2011) 570-582.
- [21] T. Kasuba, Simplified Fuzzy ARTMAP, *AI Expert* 8(11) (1993) 18–25.
- [22] S. Rajasekaran, G. A. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis and Applications*. Ed. Prentice Hall.
- [23] A.C. Soria, M. González, C. de Lorenzo, I. Martínez-Castro, J. Sanz. Characterization of artisanal honeys from Madrid (Central Spain) on the basis of their melissopalynological, physicochemical and volatile composition data. *Food Chemistry* 85 (2004) 121–130.
- [24] Escriche, M. Kadar, E. Domenech, L. Gil-Sánchez, A potentiometric electronic tongue for the discrimination of honey according to the botanical origin. Comparison with traditional methodologies:

- Physicochemical parameters and volatile profile, *Journal of Food Engineering* 109(3) (2012) 449-456.
- [25] J. M. Barat, L. Gil, E. García-Breijo, M. C. Aristoy, F. Toldrá, R. Martínez-Máñez, J. Soto. Freshness monitoring of sea bream (*Sparus aurata*) with a potentiometric sensor. *Food Chemistry*, 108, May 2008, 681-688.
- [26] L. Gil, J. M. Barat, E. Garcia-Breijo, J. Ibañez, R. Martínez-Máñez, J. Soto, E. Llobet, J. Brezmes, M. C. Aristoy, F. Toldrá. Fish freshness analysis using metallic potentiometric electrodes. *Sensors and Actuators B: Chemical*, 131, May 2008, 362-370.
- [27] M. Vinaixa, E. Llobet, J. Brezmes, X. Vilanova, X. Correig, A fuzzy ARTMAP- and PLS-based MS e-nose for the qualitative and quantitative assessment of rancidity in crisps, *Sensors and Actuators B: Chemical* 106(2) (2005) 677-686.
- [28] Amari, N. El Barbri, E. Llobet, N. El Bari, X. Correig, B. Bouchikhi, Monitoring the Freshness of Moroccan Sardines with a Neural-Network Based Electronic Nose, *Sensors* 6 (2006) 1209-1223.
- [29] M. Cristhian, A. Durán, G. Oscar Gualdrón, F. Adrián Carvajal, Data acquisition, analysis and processing tool for multisensory system and mass spectrometry, *Revista Colombiana de Tecnologías de Avanzada* 1(17) (2011) 16-23.
- [30] Young Wung Kim, Jung Hwan Cho, Gi Joon Jeon, An Intelligent Wireless Electronic Nose Node for Monitoring Gas Mixtures Using Neuro-Fuzzy Networks Implemented on a Microcontroller, *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, (2007) 100-104.
- [31] E. Llobet, R. Ionescu, S. Al-Khalifa, J. Brezmes, X. Vilanova, X. Correig, N. Barsan, J.W. Gardner, Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition, *Sensors Journal IEEE* 1(3)(2001) 207-213.
- [32] E. Llobet, J. Brezmes, O. Gualdrón, X. Vilanova, X. Correig, Building parsimonious fuzzy ARTMAP models by variable selection with a cascaded genetic algorithm: application to multisensor systems for gas analysis, *Sensors and Actuators B: Chemical* 99(2-3) (2004) 267-272.
- [33] E. Llobet, J. Brezmes, R. Ionescu, X. Vilanova, S. Al-Khalifa, J.W. Gardner, N. Bârsan, X. Correig, Wavelet transform and fuzzy ARTMAP-based pattern recognition for fast gas identification using a micro-hotplate gas sensor, *Sensors and Actuators B: Chemical* 83(1-3) (2002) 238-244.

-
- [34] R. Martínez-Máñez, J.Soto, E. Garcia-Breijo, L. Gil, J. Ibáñez, E. Llobet, An electronic tongue design for the qualitative analysis of natural waters, *Sensors and Actuators B: Chemical* 104(2) (2005) 302-307.
 - [35] J.M. Barat, I. Escriche, E. Garcia-Breijo, L. Gil, R. Martinez-Manez, J. Soto, A n electronic tongue for fish freshness analysis using a thick-film array of electrodes, *Microchimica Acta* 163 (1-2) (2008) 121-129.
 - [36] L. Gil, E. Garcia-Breijo, J. Ibáñez, R.H. Labrador, E. Llobet, R. Martínez-Máñez, J.Soto, Electronic Tongue for Qualitative Analysis of Aqueous Solutions of Salts Using Thick-film Technology and Metal Electrodes, *Sensors* 6(9) (2006) 1128-1138.
 - [37] L. Gil, J.M. Barat, E. Garcia-Breijo, J. Ibáñez, R. Martínez-Máñez, J. Soto, E. Llobet, J. Brezmes, M.-C. Aristoy, F. Toldrá, Fish freshness analysis using metallic potentiometric electrodes, *Sensors and Actuators B: Chemical* 131(2) (2008) 362-370.
 - [38] Campos, L. Gil, R. Martínez-Máñez, J. Soto, J.L. Vivancos, Use of a Voltammetric Electronic Tongue for Detection and Classification of Nerve Agent Mimics, *Electroanalysis*, 22 (2010) 1643–1649.
 - [39] Vickneswaran Jeyabalan, Andrews Samraj, Loo Chu Kiong, FUZZY ARTMAP Classification for motor Imagery based Brain Computer Interface, *Arab Conference on Information Technology* (2008) 1-5.
 - [40] F. Bortolozzi, N. Murshed, R. Sabourin, A fuzzy ARTMAP-based classification system for detecting cancerous cells, based on the one-class problem approach, *Proceedings of 13th International Conference on Pattern Recognition IV* (1996) 478-482.
 - [41] J. Downs, R.F. Harrison, R.L. Kennedy, S.S. Cross, Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks, *Artif Intell Med.* 8(4) (1996) 403-28.
 - [42] Lerner, B. Vigdor, An empirical study of fuzzy ARTMAP applied to cyto genetics, *Proceedings 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, (2004) 301- 304.
 - [43] S. Mohamed, D. Rubin, T. Marwala, Multi-class Protein Sequence Classification Using Fuzzy ARTMAP, *IEEE International Conference on Systems, Man and Cybernetics*, 2 (2006) 1676-1681.
 - [44] B. Raahemi, A. Kouznetsov, A. Hayajneh, P. Rabinovitch, Classification of Peer-to-Peer traffic using incremental neural networks (Fuzzy ARTMAP), Electrical and Computer Engineering, *Canadian Conference* (2008) 719-724.

-
- [45] Chralampidis, T. Kasparis, M. Georgiopoulos, Classification of noisy signals using fuzzy ARTMAP neural networks , *IEEE Transactions on Neural Networks*, 12(5) (2001) 1023-1036.
 - [46] M. Qasim, Doos, Zouhair Al-Daoud, M. Suhair, Al-Thraa, Agent Based Fuzzy ARTMAP Neural Network for Classifying the Power Plant Performance, *Jordan Journal of Mechanical and Industrial Engineering* 2(3) (2008) 123-129.
 - [47] H. Nafisi, M. Davari, M. Abedi, G.B. Gharehpetian, Using Fuzzy ART map neural network for determination of partial discharge location in power transformers, *PowerTech*, (2009).
 - [48] M. Demetgul, S. Taskin, I. Nur Tansel, Conditioning Monitoring and Fault Diagnosis for a Servo-Pneumatic System with Artificial Neural Network Algorithms, *Artificial Neural Networks, Industrial and Control Engineering Applications*, Ed. Kenji Suzuki (2001).
 - [49] M.L. Lopes, C.R. Minussi, A. Diva, Electric load forecasting using a fuzzy ART and ART MAP neural network, *Applied Soft Computing* 5(2) (2005) 235-244.
 - [50] T. Serrano-Gotarredona, B. Linares-Barranco, A.G. Andreou, Adaptive Resonance Theory Microchips, *Circuit Design Techniques. The Springer International Series in Engineering and Computer Science* 456 (1998).
 - [51] K. A. Sumithradevi, M.N. Vijayalakshmi, A. Annamma, Vasanta, Evaluation of Fuzzy ARTMAP with DBSCAN in VLSI Application, *World Academy of Science, Engineering and Technology* 36(2007) 261-264.
 - [52] L. Zhe, Fuzzy ARTMAP Based Neuro computational Spatial Uncertainty Measures, *Photogrammetric engineering and remote sensing* (2008) 1573-1584.
 - [53] M. Salah, J C Trinder, Fuzzy ARTMAP Neural Networks for Automatic Feature Extraction from Aerial Images and Lidar Data, *Science Mcmaster ca* (2001) 1-10.
 - [54] Yaffe, Y. Cohen, G. Espinosa, A. Arenas, F. Giralt, Fuzzy ARTMAP and Back-Propagation Neural Networks Based Quantitative Structure–Property Relationships (QSPRs) for Octanol–Water Partition Coefficient of Organic Compounds, *Journal of Chemical Information and Computer Sciences* 42 (2) (2002) 162-183.
 - [55] ZheXu, Xiajing Shi, Lingyan Wang, Jin Luo, Chuan-JianZhong, Susan Lu, Pattern recognition for sensor array signals using Fuzzy ARTMAP, *Sensors and Actuators B: Chemical*, 141(2) (2009) 458-464.

-
- [56] P. Ramuhalli, L. Udpa, S.S. Udpa, Use of reliability measures to improve the performance of fuzzy ARTMAP networks, *Neural Networks International Joint Conference on* , 6 (1999) 4015-4020.
 - [57] C.C. Vilakazi, T. Marwala, Application of Feature Selection and Fuzzy ARTMAP to Intrusion Detection, Systems, Man and Cybernetics, *IEEE International Conference on* ,6 (2006) 4880-4885.
 - [58] Nachev, Data mining with Fuzzy ARTMAP neural networks: prediction of profiles of potential customers, *International Conference Knowledge-Dialogue-Solutions* (2007) 1-8.
 - [59] A.C. Subhajini, T. Santhanam, Fuzzy ARTMANEURAL network achitecture for weather forecasting, *Journal of Theoretical and Applied Information Technology* 34 (1) (2011) 022 – 028.
 - [60] M. Vakil-Baghmisheh, N. Pavešić, A Fast Simplified Fuzzy ARTMAP Network, *Neural Processing Letters* 17 (3). (2003) 273-316.
 - [61] B.W. Jervis, T. Garcia, E.P. Giahnakis, Probabilistic simplified fuzzy ARTMAP (PSFAM)”, *IEE Proc., Sci. Meas. Technol.* 146 (1999) 165-169.
 - [62] M. Vuskovic, D. Sijiang, Classification of Prehensile EMG Patterns With Simplified Fuzzy ARTMAP Networks, *Proceedings of the 2002 International Joint Conference on Neural Networks* (2002) 2539-2544.
 - [63] S. Rajasekaran, G.A. VijayalakshmiPai, Simplified Fuzzy ARTMAP as Pattern Recognizer, *J. Comput. Civ. Eng.* 14(92) (2000) 92-100.
 - [64] S. Rajasekaran, G.A. VijayalakshmiPai, Image recognition using simplified fuzzy ARTMAP augmented with a moment based feature extractor, *International Journal of Pattern Recognition and Artificial Inteligence* 14(8) (2000) 1081-1095.
 - [65] A.A. Thomas, M. Wilsy, An Improved and Adaptive Face Recognition Method Using Simplified Fuzzy ARTMAP, *Advanced Computing, PT III. Book Series: Communications in Computer and Information Science.* 133(Part III) (2011) 23-34.
 - [66] T. A. Annam, M. Wilsy, A Comparative Study of Feed forward Neural Network and Simplified Fuzzy ARTMAP in the Context of Face Recognition, 1st International Conference on Computer Science and Information Technology Location. *Advances in network and communications, PT II. Book Series: Communications in Computer and Information Science* 132(Part 2) (2011) 277-289.
 - [67] T. Annam, M. Wilsy, Face recognition using simplified fuzzy ARTMA, *Signal and Image Processing : An International Journal* 1(2) (2011) 134-146.

- [68] S. Boonpoke, B. Marungsri, Pattern Recognition of Partial Discharge by using Simplified Fuzzy ARTMAP, *World Academy of Science, Engineering and Technology* 65 (2010) 212-219.
- [69] B. Marungsri, S. Boonpoke, Applications of simplified fuzzy ARTMAP to partial discharge classification and pattern recognition, *WTOS* 10(3) (2011)69-80.
- [70] C.K. Loo, M.V.C. Rao, Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP, *Knowledge and Data Engineering, IEEE Transactions on* , 17(11) (2005) 1589- 1593.
- [71] K.V.R. Ravi, Ramaswamy Palaniappan, S.H. Hen, Simplified fuzzy ARTMAP classification of individuals using optimal VEP channels, *International Journal of Knowledge-Based and Intelligent Engineering Systems*. 10 (6) (2006) 445-452.
- [72] P. Venkatesan, M.L. Suresh, Classification of Renal Failure Using Simplified Fuzzy Adaptive Resonance Theory Map, *International Journal of Computer Science and Network Security* 9(11) (2009) 129-134.
- [73] L. Gil-Sanchez, E. Garcia-Breijo, J. Garrigues, M. Alcaniz, I. Escriche, M. Kadar, Classification of honeys of different floral origins by artificial neural networks, *IEEE Sensors*, (2011) 1780-1783.

Chapter 3

APPLICATION OF PATTERN RECOGNITION IN OPTIMIZATION-SIMULATION TECHNIQUE

G. M. Antonova*

Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia

ABSTRACT

The method of approximate optimization of dynamic stochastic systems widely known as the optimization-simulation method is considered. It is realized by means of the grid's method of uniform probing of the space of parameters called LP_{τ} -search with averaging.

The statements of optimization problems, discussion of the algorithms for solving them and application of methods of pattern recognition theory for evaluation the efficiency region of dynamic stochastic systems are involved in this chapter. The efficiency region is defined as the region in the space of parameters where the system quality indices are better than in other regions. Pattern recognition methods allow accelerate the evaluation of efficiency regions by ensuring better quality of processing the results of simulation experiments.

* Phone.: (095) 334 90 50, E-mail: gmant@ipu.ru.

INTRODUCTION

A grid's methods are widely used in modern computing algorithms. It deals with procedures for calculating of significances of functions in points with special coordinates. These coordinates correspond to grid's centers in multidimensional space of parameters. In multidimensional domain of parameters or at the surface difference grid is defined algorithmically as ensemble of grid's meshes in the domain or at the surface and components of lateral bound of meshes, i.e. vertexes, edges and others. Analysis of features of grid's methods using finished with creation of theory of difference schemes (A.N. Tikhonov, A.A. Samarskiy and others). This theory allows find solving by means of computing for wide category of problems, which represented only with the aid of differential operator, initial and boundary conditions.

The procedure of creation of difference algorithm involves following stages: choice of type of equation with differential operator, development of difference scheme, making of algorithm for calculating of coordinates of grid's centers, proof of convergence of difference task solving to differential task solving, evaluation of speed of convergence and error of accepted decision. On the base of this scheme both accurate and approximate algorithms may be created. A scheme for investigation for complicated systems with analytic expression contains a few stages. First of all descriptive model is created for complicated systems, having mathematical expression. It must take into account physical and others features of system. Then accurate mathematical description is chosen. If there are a few accurate models, a few parallel investigations are conducted. Among results those with best precision, speed of computing and others characteristics are selected. Secondly computing algorithm in the form of series of arithmetical and logical operations, giving numerical decision of problem, is created. Sometimes additional investigation of mathematical model for testing of precision of problems definition, complement and consistency of initial date, existence and oneness of decision, value of decision error and others is conducted.

Programming system and computer are chosen on third stage. Program is created and coded. Computing experiment is organized on forth stage. Series of calculation with computers program is fulfillment. Every calculation finished with fixation procedure for future analysis and comparison with other results. Detailed analysis, model correction, choice of next model in the case of a few models and recommendations preparation for functioning's advance of complicated system are concluded investigation. Unfortunately this scheme is not evaluable always. An application of grid's methods for investigation of

complicated systems, represented in form of imitation statistical model, is problem of today. Formalized description for them is selected only after decomposition. Separate parts of system don't formalize. They are described with heuristic procedures. New optimization-simulations methods for solving of multiparameters multicriteria optimization problem are considered in this chapter.

1. OPTIMIZATION WITH SIMULATION MODEL USING

At initial stage of investigation the set of input values parameters and the reaction of the object to selected combinations of values of input parameters are accumulated empirically. This state of model is named by N. Wiener, one of the fathers of cybernetics, as "black box" model. The knowledge about object is close to zero in such moment. A preliminary description is formed when information is analyzed. The "black box" model becomes lighter at this moment. When the object description converts to formalized and detailed the "black box" is turning first to "gray box" model and then to "transparent box" model or "white box" model.

If object has a high level of complexity or large scale and enormous quantity of internal or external ties imitation model may be created at any stage of description. The first imitation models were realized at computers in the 20th century [1, 2]. The costs to development of simulation models are justified by the fact that they can be used to solve all kinds of problems. These may be predicting of object's behavior, estimating the influence of changes in the object or in its environment to quality of object's functioning, checking of performability for different projects of production of new unprecedented items, optimizing the structure or parameters of object by checking them in the process of imitation experiments on simulation models and choosing the best variant from many evaluating variants. All well known optimization methods that require define values of derivatives or their estimates can not be applied to simulation models. The method of combinatorial optimization is the only way to decide nondifferentiable optimization problems [3]. Moreover, the solution of optimization problem can be called "rational," but not optimal, since simulation experiments study the behavior of the object for particular values of parameters, yet this region lacks the comprehensive description of the object's behavior for continuously changing input parameters. This is the so-called optimization-simulation method [4, 5].

The optimization using simulation models includes the following sequence of actions.

- 1) Creating the simulation model of the object and computer implementation of the simulation model.
- 2) Simulation experiments realization and defining the space of variants to solve the optimization problem.
- 3) Applying the model of choosing a “rational” solution.
- 4) Implementing the obtained solution and checking its quality

Various grid's methods may be used to find the approximate solution of optimization problem. It is the combinatorial method, where in order to reduce the number of experiments with the imitation statistical model, the input parameters are chosen equivalent to the coordinates of the points of the grid uniform in the multidimensional space of parameters, in particular LP_τ -search with averaging. Various uniform grids are widely used in practice [6, 7]. In the one-dimensional case, for the continuous function $f(x)$ given on the interval (a, b) and bounded below on it and satisfying the Lipschitz condition [8], the following values for enumeration can be used:

$$x_k^l = a + (b - a)(2k - 1) / (2l), \quad k = 1, \dots, l,$$

where l is the number of enumeration points. Such grid ceases to be uniform after new points are added. So if the increase is not aliquot to l , full recalculation is required for the increasing the number of enumeration points l . If attempts to use the information about features of $f(x)$ are fulfilled, the points are also distributed nonuniformly.

In a space of larger dimension, the challenge is to choose a grid with good randomness and uniformity properties. The cubic grid $\Theta_N^{(1)}$ consists of $N = p^n$ points with the coordinates:

$$\left(\frac{i_1 + 1/2}{p}, \frac{i_2 + 1/2}{p}, \dots, \frac{i_n + 1/2}{p} \right)^T, \quad i_k = \overline{0, p-1}, \quad k = \overline{1, n},$$

where p is the base of the number system and the index T is the transposition operation of the row vector.

The rectangular grid Θ_N^2 results from generalizing the procedure of constructing the cubic grid. As the values of coordinates of the points of the rectangular grid are calculated, the lengths of the cube sides can be divided into different numbers of parts. The random grid Θ_N^3 includes N independent realizations of a multidimensional random vector distributed uniformly in the space of n variables. The Hammersley–Halton grids Θ_N^4 are formed of the first N terms of the Halton sequence [9] given by transformation of mutually coprime numbers r_1, \dots, r_n . The Halton sequence includes the sequence of points in R^n , such that their Cartesian coordinates $(p_{r_1}(i), \dots, p_{r_n}(i))$, $i = 1, 2, \dots$ for the value $i = a_m a_{m-1} \dots a_2 a_1$ represented in the number system with the base r are calculated by the formula $p_r(i) = 0, a_1 a_2 \dots a_{m-1} a_m$.

The parallelepiped grid Θ_N^5 [10, 11] consists of the sequence of points $x_j = (\{a_{1j}/N\}, \dots, \{a_{nj}/N\})^T$, $j = \overline{1, N}$, where $N > 3$ is a prime number, $\{a\}$ is the fractional part of a , and the totality of values a_1, \dots, a_n are optimal coefficients, i.e., some numbers chosen in a special way [11]. The P_τ grid Θ_N^6 is given in a more complicated way [12–15]. To form points of the LP_τ -sequence in the unit n -dimensional hypercube K^n the so-called binary parallelepipeds P_k is singled out. They are the set of points with the coordinates (x_1, x_2, \dots, x_n) such that $x_j \in l_{kj}$ for $j = \overline{1, n}$, where l_{kj} are the binary segments that can be obtained when the segment $0 \leq y \leq 1$ is divided into 2^m equal parts, $m = 0, 1, 2, \dots$

Any binary parallelepiped belongs to the unit n -dimensional hypercube K^n . The grid, involved $N = 2^V$ points of n -dimensional hypercube K^n , is chosen. If one point of the grid belongs to each binary parallelepiped P_k with the volume $V_{P_k} = 1/N$, the grid is called a P_0 -grid. The grid points are located in the hypercube K^n according property “uniformity”. They probe the space of the hypercube K^n with maximum fullness.

If each binary parallelepiped P_k with the volume $V_{P_k} = 2^\nu / N$ and $\nu > \tau$ contains 2^τ points of the grid that include $N = 2^\nu$ points of the hypercube K^n , the grid is called a P_τ -grid. For the sequence of points $q_0, q_1, \dots, q_i, \dots$ from the hypercube K^n , the definition of the binary part of this sequence is introduced as the set of points with their numbers satisfying the inequality:

$$k2^s \leq i < (k+1)2^s; \quad k = 0, 1, 2, \dots; \quad s = 1, 2, \dots$$

Then LP_τ -sequence is defined as the sequence of points $q_0, q_1, \dots, q_i, \dots$ from the hypercube K^n such that any section of its binary part that consists of no less than $2^{\tau+1}$ points represents P_τ -grid. The grids $\Theta_N^{(4)}$, $\Theta_N^{(5)}$, Θ_N^6 are called quasi-random. The Monte-Carlo method uses them as random grids. If $N = 2^{r+\tau}$, at least one element from Θ_N^6 is included in the arbitrary binary parallelepiped from $\text{Pr} \subset K^n$ with the volume 2^{-r} . So the grid Θ_N^6 is more preferable than the grids $\Theta_N^{(4)}$ and $\Theta_N^{(5)}$. Different kinds of grids are widely used in practice [9-12, 16]. Of course, there are some grids more difficult to construct [17-25], for instance, adaptive grids [26]. If uniform grid may be constructed step by step when the number of points of the grid increases, such property of the grids is named composite nature. Not all grids have this property.

The quantitative characteristics of uniformity of the grid's points may be found by means of deviation [6, 7]. The deviation is estimated by

$$D_N(\Theta_N) = \sup_B |S_N(B) - N\mu_n(B)|,$$

where $S_N(B)$ is the number of sampling points that fell into the region B :

$$B = [0, b_1] \times \dots \times [0, b_n], \quad 0 < b_j \leq 1, \quad j = \overline{1, n},$$

and

$$\mu_n(B) = \lim_{N \rightarrow \infty} N^{-1} S_N(B) \text{ is the Lebesgue measure.}$$

The spread is given by:

$$d_N(\Theta_N) = \sup_{\vec{x} \in \Pr} \min_{\vec{q}_i \in \Theta_N} \rho(\vec{x}, \vec{q}_i),$$

where ρ is the Euclidean metrics.

According to results [12, 14, 15], the inequality holds for any grid:

$$C(n)N^{-1/n} \leq d_N(\Theta_N) \leq 2\sqrt{n}[D_N(\Theta_N)/N]^{-1/n},$$

where $C(n) > 0$ is some constant.

By [27] the inequality holds:

$$\sqrt{n/2e}N^{-1/n} \leq d_N(\Theta_N) \leq \sqrt{n}[D_N(\Theta_N)/N]^{-1/n}.$$

The following expressions hold:

$$D_N(\Theta_N^{(1)}) = (0.5)N^{1-1/n}, \quad d_N(\Theta_N^{(1)}) = (\sqrt{n}/2)N^{-1/n};$$

$$D_N(\Theta_N^{(3)}) = O(N^{-1/2}), \quad d_N(\Theta_N^{(3)}) = O(N^{1/(2n)}), \quad N \rightarrow \infty;$$

$$D_N(\Theta_N^{(i)}) = O(N^{-1} \ln^n N)$$

$$d_N(\Theta_N^{(i)}) = O(N^{-1/n} \ln N), \quad N \rightarrow \infty, \quad i = 4, 5, 6$$

The grids $\Theta_N^{(4)}$, $\Theta_N^{(5)}$, $\Theta_N^{(6)}$ are best in terms of the deviation value.

The grid $\Theta_N^{(1)}$ is optimal for $n=1$, but it is worse than the grid $\Theta_N^{(3)}$ as

early as $n \geq 3$. The cubic grid is optimal by the order for any n with respect to the spread value. The grids $\Theta_N^{(4)}$, $\Theta_N^{(5)}$, $\Theta_N^{(6)}$ are almost optimal with respect to the order. It is shown in [6, 7] that P_τ -grid $\Theta_N^{(6)}$ is optimal by the order when the dependence of d_N on the set of grid's points is taken in account. The grid $\Theta_N^{(3)}$ is widely used in practice, the grids $\Theta_N^{(1)}$, $\Theta_N^{(6)}$ are used more rarely, and the grid Θ_N^4 is used seldom [28]. The lack of composite nature limits an application of the grids $\Theta_N^{(1)}$ и $\Theta_N^{(5)}$.

The choice of the grid is a separate difficult problem. There should be an algorithm for calculating the coordinate points with acceptable complexity for the computer model realization. LP_τ -sequence has very useful property. The quantity of grid points may be increased after preliminary analysis of the obtained results by means of growth density of point's placement. The values obtained earlier in the previous experiments will keep in the set of data.

I.M. Sobol' analyzed in detail in [29] the existing grids and imposed the requirements on sequences of points distributed uniformly in the multidimensional space of parameters. These requirements assume the fastest decrease of values of D_N or φ_∞ as the number of grid's points increases, sufficiently small values of constants for estimating deviation and nonuniformity:

$$D_N = O(N^{-1} \ln^n N),$$

$$\varphi_\infty(\mathbf{q}_1, \dots, \mathbf{q}_N) \leq O(n, \tau),$$

where \mathbf{q}_i - are the points of the LP_τ -sequence, and the decrease of the ratios D_N/N and φ_N/N as early as for small values N .

The Halton sequence satisfies these requirements partially, while the LP_τ -sequence satisfies them to a greater degree. Attempts to create grids with better properties are still being made. For instance, in [30], a new procedure for constructing the sequence of numbers with good uniformity properties on the plane, i.e., in the two-dimensional space, is proposed.

The grids of more complicated types are applied for solving of problems that demand finding values of derivatives. The difference grid in the multidimensional region or on the plane is given as the algorithmically given set of grid cells in the region or on the surface and components of lateral faces of cells, i.e., vertices, edges, etc. The grid nodes correspond to cell vertices. Three classes of grids, i.e., structural, nonstructural, and hybrid, are used to solve computational problems in multidimensional regions. Structural grids [21, 26, 31, 32] are regular. They can be constructed by nondegenerate transformation as the mapping of nodes and cells of the uniform grid. Coordinate grids have such property that their points may be easy to numerate and the quantity of nodes can be increased fast if we need to increase accuracy or to estimate convergence. Coordinate grids have become common among structural grids.

Structural grids are extremely convenient for parallel computational algorithms realization. Structural grids are constructed by algebraic methods [33, 34] such as the stretch method [35, 36]. The equidistribution method helps create adaptive grids that eliminate oscillations and provide more accurate description of the solution in stationary and nonstationary problems in the regions of large gradients [37]. Elliptical methods adapt the grid nodes by choosing the coefficients of derivatives or absolute terms in equations. For instance, papers [38–40] may be highlight among the great variety of works that deal with application of such grids. Variation methods of constructing adaptive grids [41] are classified in detail in [31] with respect to functional given on the set of smooth or discrete transformations. Variation methods allow taking into account the entire totality of requirements on the quality of the grid that cannot be fully implemented by other methods. These requirements can include non degeneracy, smoothness, uniformity, adaptability, etc. The method of projections combined with methods of constructing quasi uniform grids allows creating programs of automated grid construction to solve problems with a complicated geometry of boundaries of the region of search for the solution and a complicated structure of the solution.

Grids nondegenerate in simply connected regions can be constructed using the theory of harmonic functions. Moving grids are created to solve nonstationary problems [42]. Nonstructural grids [43] are of irregular nature, and their cells can have an arbitrary form. The spatial coordinates of nodes of nonstructural grids are determined by complicated calculations, with special algorithms required to numerate them. Hybrid grids combine features of

structural and nonstructural grids and are applied mostly to numerically solve boundary problems in regions of complicated form [44].

2. CLASSIFICATION OF OPTIMIZATION-SIMULATION PROBLEMS

There are many different statements of the optimization problem for the object represented by a simulation model. Figure 1 shows the proposed classification with respect to the dimension and type of optimization criterion [5].

2.1. Single-Criterion Problems

The single-criterion statement of the optimization problem is as follows. Find the vector of parameters α and the structure of the object such that its operation quality is described by the functional $F(\alpha)$, which ensure

$$\text{extr } F(\alpha)$$

if

$$f_i(\alpha) \leq 0, \quad i \in I; \tag{1}$$

$$f_j(\alpha) \leq 0, \quad j \in J. \tag{2}$$

This statement divides restrictions imposed on the vector of parameters into two groups. The first group includes the restrictions given as mathematical expressions while the second group includes nonformalized restrictions represented in the algorithmic form. Thus, the second group allows taking into account partially formalized restrictions. Algorithmic restrictions are implemented in the computer model and are checked as simulation experiments are held.

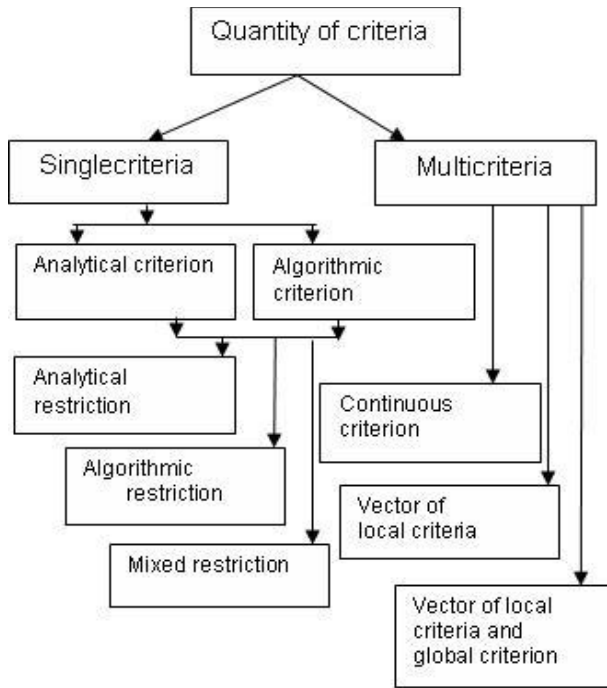


Figure 1. Classification of optimization-simulation problems.

The high performance of modern computers and computing systems allows present the functional $F(\alpha)$ in the algorithmic form too. There are six possible variants of statements for single-criterion optimization problem depending on the type of optimization criterion and the restrictions. For each variant, features of the criterion and the restrictions are taken into account and methods of solving the problem are proposed. In the simplest first case of the single-criterion statement of the optimization problem, the optimization criterion is given in analytical form and the analytical restrictions are chosen from the set I_0 . There are numerous methods designed to solve such problems. The high formalization level allows applying both classical analytical optimization methods and methods of mathematical programming. Unfortunately, completely formalized problems are extremely rare in practice. The great disadvantage of such statements is that they neglect the stochastic parameters and the influence of internal and external random noise on the object. The more complicated second case contains the optimization criterion in analytical form, with restrictions given in the algorithmic form and chosen

from the set J_0 . A simulation model is needed to describe the behavior of the object. Searching for an approximate solution of optimization problem involves constructing a computer model and performing simulation experiments on it to check whether or not the imposed restrictions are satisfied. Planning simulation experiments and applying guided simulation modeling and grid's methods of uniform probing of the space of parameters and other procedures help decrease the optimization efforts.

In the third case, for the analytical optimization criterion, the restrictions can be mixed; i.e., they can be chosen from the set $I_0 \cap J_0$. For such types of problems, part of the restrictions is formalized and given analytically while the other part of them is given algorithmically. Special procedures are needed to process the results of simulation experiments. It is better to check analytical restrictions at first stage of processing the results of simulation experiments and check algorithmic restrictions for the rest of the variants. Thus, we can decrease the quantity of simulation experiments. If the resulting solution does not suit the researcher, additional iterations are possible. If the optimization criterion is given algorithmically (the fourth variant of the single-criterion problem statement) and analytical restrictions are chosen from the set I_0 , strict classical optimization methods cannot be used. It is need to construct a simulation model, choose a simulation algorithm, and create a computer model. After processing of the significant quantity of simulation experiments a "rational variant" of parameters or the object's structure may be chosen. To speed up optimization, all existing simulation methods, all existing experiment planning methods, and all existing variants of optimization-simulation methods may be used.

In the fifth case, the optimization criterion used to optimize nonformalized objects for which minimal analytic description is given algorithmically. The restrictions are chosen from the set J_0 , i.e., they are represented algorithmically. For such a problem statement, the simulation model of the object is changed in the process of simulation experiments. The initial variant of the model may be turn out in extremely complicated since the description of the object is not adequate for real functioning of the object. Efforts to solve the optimization problem are reduced by planning simulation experiments and applying optimization-simulation methods. Various special methods, in particular LP_τ -search with averaging or guided simulation modeling, allow solving the optimization problem faster if they take into account features of the object. The last, sixth case, corresponds to the algorithmic form of the

optimization criterion. The restrictions are chosen from the set $I_0 \cap J_0$, i.e., the previous problem becomes more complicated since analytical restrictions are added. Some restrictions are still represented algorithmically. Obviously, this case preserves all obstacles of solution searching discovered in the optimization problem of the fifth type. The methods for searching of the optimization solution are similar to those proposed for the optimization problem of the fifth type. While checking whether the restrictions are satisfied, it is reasonable to check analytical restrictions and then algorithmic restrictions for the rest of the variants.

2.2. Multicriteria Problems

Multicriteria multiparametric optimization-simulation problem may be formalized as follows.

Find the vector of parameters α and the structure of the object such that the set of functionals $F_l(\alpha)$, $l = \overline{1, L_{crit}}$, choosing as quality indices, will ensure joint extremum

$$extr\{F_l(\alpha)\}, l = \overline{1, L_{crit}}$$

if

$$f_{li}(\alpha) \leq 0, i \in I; \quad (3)$$

$$f_{lj}(\alpha) \leq 0, j \in J. \quad (4)$$

Usually a unique solution satisfying all restriction and ensuring the joint extremum for all quality indices does not exist. So the solution of the problem is sought in the form of the Pareto set. The results of imitation experiments are used to find the values of quality indices. Additionally a set of ill-formalized reasoning omitted at the stage of quality indices choosing is taken in account. The examples of solving multicriteria multiparametric optimization-simulation problems for a number of technical systems and detail consideration of statement of the optimization problems are fulfilled in [45-52].

For the dynamic stochastic systems functioning under the action of external and internal noises approximate solution of multicriteria multiparametric optimization-simulation problem with a set of quality indices in the form of incomplete mean :

$$K_j = \int_0^\infty \int_G \int_{\Omega} f_j(\alpha(t), \omega) w_G(\alpha(t), \omega) d\omega d\alpha dt, \quad j = \overline{1, J}, \quad (5)$$

may be found. At the formula (5) $K_j, j = \overline{1, J}$, is the set of quality indices at the system output, G is the efficiency region found in the process of search, Ω is the range of values of stochastic parameters, $f_j(\alpha(t), \omega)$ is the function describing the j -th quality index, $\alpha(t)$ is the vector of input parameters with the dimension n_1 , ω is the vector of random external and internal action (noises) with dimension n_2 , t is time, J is total number of criteria,

$$w_G(\alpha, \omega) = w(\alpha, \omega) / \int_G \int_{\Omega} w(\alpha, \omega) d\omega d\alpha$$

is the distribution density normalized with respect to the region G , $w(\alpha, \omega)$ is the distribution density such that:

$$\int_{-\infty}^{\infty} \int_{\Omega} w(\alpha, \omega) d\omega d\alpha = 1.$$

The region in the Euclidean space of parameters G must be found where the joint extremum of quality indices (5) is realized in the sense of solving the multicriteria multiparametric optimization-simulation problem. It means that the estimate of the efficiency region or the set of estimate of the efficiency region from the space of estimates with the given metrics [52] for which the quality indices form the Pareto set must be found. In this integration region G , named the efficiency region, the averaged values of the quality indices should be better than in other regions and conditions are fulfilled:

$$\begin{aligned}
 K_j(G) &\geq K_{jz}, \quad j = \overline{1, q}, \\
 K_j(G) &\leq K_{jz}, \quad j = \overline{q+1, J}.
 \end{aligned}
 \tag{6}$$

2.3. Optimization Problem with Continuous Optimization Criterion

The design problems connect with another variant of the multicriteria optimization problem [53]. Its statement takes into account requirements on ensuring the given form of the continuous criterion. It is no possible to solve optimization problem with indefinite set of criterion but after discretization of continuous curve optimization problem may be formulated as problem with vector of criteria whose components are homogeneous and are calculated according to one common algorithm. Initial continuous curve defines restrictions imposed on all components of the vector of criteria. The modified statement of the multicriteria multiparametric optimization-simulation problem with continuous criterion may be formalized as follows.

The set of indices of quality is given in the form of incomplete mean:

$$K_j = \int_0^\infty \int_G \int_{\Omega} f_j(\alpha(t), \omega) w_G(\alpha(t), \omega) d\omega d\alpha dt, \quad j = \overline{1, J}, \tag{7}$$

where $K_j, j = \overline{1, J}$, is the set of quality indices at the system output, J is total number of criteria. Additional vector of optimization criteria, connected by the common algorithm of calculation, is given:

$$CK_j, j = \overline{J+1, J+n_c}, \tag{8}$$

where n_c is the quantity of vector components.

In addition G is the region of efficiency which found by means of searching procedure, Ω is the range of values of stochastic parameters, $f_j(\alpha(t), \omega)$ is the function describing the j -th quality index, $\alpha(t)$ is the

vector of input parameters with the dimension n_1 , ω is the vector of random external and internal action (noises) with dimension n_2 , t is time,

$$w_G(\alpha, \omega) = w(\alpha, \omega) / \int_G \int_{\Omega} w(\alpha, \omega) d\omega d\alpha$$

is the distribution density normalized with respect to the region G , $w(\alpha, \omega)$ is the distribution density satisfying the condition:

$$\int_{-\infty}^{\infty} \int_{\Omega} w(\alpha, \omega) d\omega d\alpha = 1.$$

Restrictions for point criteria are formulated as:

$$K_j(G) \geq K_{jz}, \quad j = \overline{1, q}; \quad (9)$$

$$K_j(G) \leq K_{jz}, \quad j = \overline{q+1, J},$$

where K_{jz} , $j = \overline{1, J}$, are the chosen boundary values. Restrictions for the components of the vector criterion approximating the criterion given in the form of continuous curve are formulated as:

$$CK_{bj} \leq CK_j \leq CK_{tj}, \quad j = \overline{J+1, J+n_c}, \quad (10)$$

where K_{bj} , K_{tj} , $j = \overline{1, n_c}$, are the lower and upper boundary values, respectively.

The region in the Euclidean space of parameters G must be found where the joint extremum of quality indices (7) is realized in the sense of solving the multicriteria multiparametric optimization-simulation problem. It means that the estimate of the efficiency region or the set of estimate of the efficiency region from the space of estimates with the given metrics [52] for which the indices of quality form the Pareto set must be found. In this integration region G , named the efficiency region, the averaged values of the quality indices

should be better than in other regions and conditions for the simulated values are fulfilled:

$$\tilde{K}_j \geq K_{jz}, j = \overline{1, q}; \quad (11)$$

$$\tilde{K}_j \leq K_{jz}, j = \overline{q+1, J}.$$

The components of the vector criterion (8) should fall within the region of admissible values (10), chosen in the initial statement of the problem.

3. ALGORITHM FOR THE EFFICIENCY REGION SEARCHING

The algorithm of LP_{τ} -search with averaging is considered for the problem that takes into account the continuous optimization criterion. Although its dimension is greater than that of the multicriteria optimization algorithm in its initial variant for statement of the problem with criteria (5), the algorithm is the same in terms of the principal stages.

Results of simulation experiments are placed in database to reduce the analyzing efforts. Samples of results of simulation experiments, for which conditions (9), (10) are met and that correspond to the formed hypotheses on the efficiency region, are created automatically by DBMS tools. The simulation experiments can yield one estimate of the efficiency region or a series of estimates

$$G_1, G_2, \dots, G_i, \dots, G_l,$$

corresponding to different hypotheses on the efficiency region. The proposed solving algorithm [47, 48, 52, 53] includes heuristic procedures. The solution found as a result of simulation experiments has the form of a series of subsequences of points of holding the simulation experiment

$$L_{G_1}, L_{G_2}, \dots, L_{G_i}, \dots, L_{G_l},$$

approximated the efficiency region. The solution can be described by a set of functions that correspond to some sequence of points of the LP_{τ} - sequence and its respective estimate. For instance, for the estimate G_i

$$r_{G_i}(\alpha), r_{\min G_i} \leq r_{G_i}(\alpha) \leq r_{\max G_i}.$$

If functions cannot be selected, the estimates of the efficiency region can be represented by the sequence of rules. Rule 1 is used for the first estimate, Rule 2 - for the second estimate, ..., Rule i - for the i -th estimate, and so on. Modern DBMS can help search for the approximate solution of the problem of finding the efficiency region. If selection rule (10), (11) is multiply applied and various hypotheses on the form of the efficiency region G are checked the estimates of the efficiency region may be successfully generated [54]. The simplest way to describe estimates of the efficiency region G is to construct multidimensional parallelepipeds

$$\alpha_{s \min} \leq \alpha_{si} \leq \alpha_{s \max}, \quad s = \overline{1, n_1}, \quad i = \overline{1, l},$$

where $\alpha_{s \min}, s = \overline{1, n_1}$ are the minimal values of coordinates of input parameters α , and $\alpha_{s \max}, s = \overline{1, n_1}$ are the maximal values of coordinates of input parameters α .

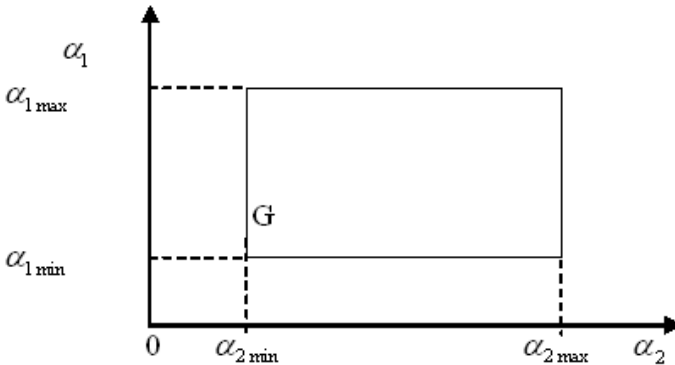


Figure 2. Efficiency region in the form of a parallelepiped for the two-dimensional space of input parameters.

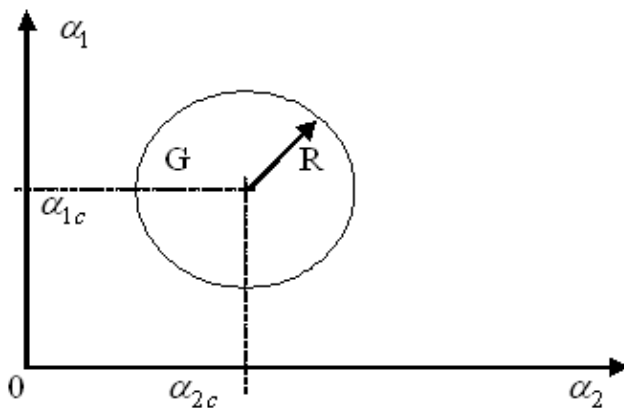


Figure 3. The efficiency region in the form of a ball for the two-dimensional space of input parameters.

The example of such solution for the two-dimensional case is shown in Figure 2.

A more complicated way of representing the solution to the problem of determining the efficiency region G is to construct a ball with a minimal radius to cover all selected points. In the two-dimensional case, the region G is shown in Figure 3. To determine the coordinates of the center of the ball α_{1c}, α_{2c} and the radius of the minimal length R_{min} it is need to solve the search problem

$$\min R$$

under L_v conditions associated with the elements of the obtained sample:

$$(\alpha_{1i} - \alpha_{1c})^2 + (\alpha_{2i} - \alpha_{2c})^2 \leq R, \quad i = \overline{1, L_v}.$$

3.1. Application Pattern Recognition Methods in the Algorithm of Efficiency Region Search

The formal instrument for selecting decision functions or discriminators and creating the decision rule that was developed within the pattern recognition theory can be applied to search the efficiency region in the

optimization simulation [51]. Pattern recognition theory [55] helps at discover differences between the initial data. Comparison is fulfilled with the aggregate of patterns rather than with particular patterns. For this purpose, attributes or invariant properties are defined on sets of objects that form particular totalities called classes of patterns. Significant attributes should be preliminarily marked out from the large amount of insignificant details that characterize the initial data. There are known three principal pattern recognition problems.

- 1) Decomposing the results of measurements of parameters characterizing the object into classes of sets that do not overlap.
- 2) Separation pattern attributes or properties. Selecting significant attributes among the general amount of characteristics of the object reduce the dimension of the pattern vectors. Usually doesn't possible obtain the full set of distinctive attributes for all established classes at once. Several successive checking and screening stages are performed until the recognition process is converted to comparing new measured elements with standard ones or table scanning. This procedure is called preliminary processing and choice of attributes.
- 3) Creating optimal decision procedures needed for identification and classification.

The latter problem is transformed to the problem of constructing boundaries of regions of solutions that correspond to the chosen classes. Boundaries can be determined by the decision functions (discriminators) that depend on the pattern α :

$$d_1(\alpha), d_2(\alpha), \dots, d_M(\alpha),$$

where M is the total number of classes of patterns. After the values of discriminators are measured and calculated, the condition

$$d_i(\alpha) > d_j(\alpha), \quad j \neq i, \quad j = \overline{1, M},$$

is checked. If it is true, the pattern α belongs to the class for which the region of solutions is singled out by the function $d_i(\alpha)$.

The search for the efficiency region within multicriteria multiparametric optimization of dynamic stochastic systems with respect to the criterion of the incomplete average using LP_{τ} -search with averaging [47, 48, 51–53] can be represented as a problem of recognizing the simulation experiments such that their parameters correspond to the points of the sequence L_{G_i} , approximated the estimate of the efficiency region G_i .

Features of the recognition problem in the search of decision functions defining the efficiency region include the following:

- 1) preliminarily setting the totality of characteristic attributes based on the statement of the optimization problem with respect to the criterion of the incomplete average,
- 2) increasing the dimension of the vector of patterns in the course of checking and correcting the set of distinctive attributes, which is due to the fact that the number of attributes in the set grows because of addition some “indicators” of the selected classes known from analytical studies,
- 3) absence of checking for recognition correctness, which is due to the features of the statement optimization problem with respect to the criterion of the incomplete average, for which an accurate analytic solution is not known.

A table of standard values of the attributes $\alpha(t)$ may be created with using results of simulation experiments. According to the heuristic algorithm of optimization with criterion of the incomplete average of stochastic systems represented by simulation models two classes may be formed. Conditions (11) are fulfilled in the first class C11, while they are not in the second class C12. Decomposition into classes is done in the process of performing simulation experiments and calculating the values of quality indices (5). It isn't possible to use information on the quality indices K_j , $j = \overline{1, J}$ in further procedures of searching for decision functions even for the known analytical descriptions. These descriptions for stochastic systems do not allow demonstrate the explicit linear and nonlinear dependence of the quality indices on the chosen parameters and establish the correspondence of conditions (11) and totalities of values of the chosen significant attributes. The problem of searching for optimal decision procedures for determining the efficiency region is

formulated as follows. Use the known values of attributes of the singled out classes C11 and C12 to create the decision function $d(\alpha)$, matching the pattern α exactly with one of the given classes. The feature of the stated problem is that a priori probabilities for objects of different classes' appearance are different. The probability of appearance for objects of the class C11 is significantly smaller than the probability of appearance for objects of the class C12. Moreover, the measured values of attributes are not distributed normally.

Choosing the algorithm of decision function searching both the type of the distribution law and the probability of appearance for objects of different classes must be taken into account. The search algorithm should

- 1) take into account, where possible, different probabilities for the objects to belong to the classes C11 and C12,
- 2) be invariant with respect to the distribution laws of attributes,
- 3) be stable to the influence of noise on values of attributes given by parameters of the dynamic stochastic system.

4. EXAMPLES OF APPLICATION LP_τ -SEARCH WITH AVERAGING

Considered examples include adaptive control of the power of a short-wave fading radio channel transmitter [45], choice of noise combating correcting code for the Viterbi decoder [46], choice of parameters of the jumping control block for a network satellite radio navigation system [49], the choice of parameters of powerful vacuum resonator tubes [50], etc. The developed methods are used to solve several problems of studying dynamic stochastic systems. Being typical examples of dynamic stochastic systems, modern radio engineering systems are typical class of dynamic stochastic systems and they were chosen as the illustration of application. Processes that occur in radio engineering systems are stochastic, with features of their behavior in time requiring special modes and methods of adaptive inspection and control to be developed. Various extremal problems that accompany the design of radio engineering systems cannot always be solved by classical methods of searching for the extremum. They require special methods to be created that allow, given no exact solution, obtaining the approximate solutions called "rational solutions". The system of data transmission with

adaptive control of the power of the transmitter is designed for data transmission with small energy costs. Its reliability is given as the integral [56, 57]

$$N(\mu_0) = \int_{\mu_0}^{\infty} w(\mu) d\mu,$$

where μ is the coefficient of data transmission in the fading radio channel, μ_0 is the threshold value of the transmission coefficient $\mu(t)$, below which it is not reasonable to transmit because of high noise, and $w(\mu)$ is the density of the probability distribution of the data transmission coefficient in the fading radio channel that has the form [57]

$$w(\mu) = \frac{\mu}{\sigma_x \sigma_y} \exp\left(-\frac{\mu}{2\sigma_x^2} - \frac{m_x^2 \sigma_y^2 + m_y^2 \sigma_x^2}{2\sigma_x^2 \sigma_y^2}\right) \times \\ \times \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \frac{(2k-2l-1)!! (\sigma_y^2 - \sigma_x^2)^k m_y^{2l} \sigma_x^{2l}}{k! (2l)! 2^k} \frac{\sigma_y^{2k+4l} m_x^{k+l}}{\sigma_y^{2k+4l} m_x^{k+l}} \mu^{k+l} I_{k+l}\left(\frac{m_x^2}{\sigma_x^2} \mu\right).$$

The control efficiency for the approximate description of the probability distribution density of the data transmission coefficient in the fading radio channel by the Nakagami law [58]

$$w(x, m) = \frac{2m^m}{\Gamma(m)} x^{2m-1} \exp(-mx^2), x \geq 0,$$

where m is the depth of fading, $x = \mu / \sqrt{\bar{\mu}^2}$, $\bar{\mu}$ is the mean value of the transmission coefficient in the communication channel, $\Gamma(m)$ is the gamma function [59, 60], has the form [56]

$$\eta_P = \frac{\Gamma(m)}{mx_0^2 \Gamma((m-1), mx_0^2) + \phi(m, mx_0^2)},$$

where $x_0 = \mu_0 / \sqrt{\bar{\mu}^2}$, ϕ is the coefficient that gives the value of the power for probing the communication channel in the time intervals when $\mu_i < \mu_0$,

$$\Gamma(m-1, mx_0^2), \Gamma(a, x) = \Gamma(a) - \gamma(a, x) = \int_x^\infty e^{-t} t^{a-1} dt,$$

$a = m-1, x = mx_0^2$ is the incomplete gamma function [59, 60], and

$$\gamma(m, mx_0^2), \gamma(a, x) = \int_0^x e^{-t} t^{a-1} dt, \operatorname{Re}(a) > 0, \quad a = m, x = mx_0^2 \quad \text{is the}$$

incomplete gamma function [59, 60].

If the considered stochastic system has been studied in detail, it is possible to find an empirical indicator or decision rule for selecting the points belonging to the efficiency region. For instance, the algorithm of functioning of the data transmission system with adaptive control of the transmitter power [45, 48, 52] is efficient in the range of values of the parameters

$$0.5 < (m_x^2 + m_y^2 + \sigma_x^2 + \sigma_y^2)^2 / \{2(\sigma_x^4 + \sigma_y^4 + 2m_x^2\sigma_x^2 + 2m_y^2\sigma_y^2)\} < m_{min}, \quad (12)$$

where m_x, m_y are mathematical expectations, σ_x, σ_y are variances of orthogonal components of the transmission coefficient of the fading short-wave radio channel, and, m_{min} is the value of the depth of fading calculated with taking into account restrictions imposed on the quality indices of the data transmission system, i.e., reliability and control efficiency.

This definition of the efficiency region is obtained for the approximate description of the distribution law of the module of the transmission coefficient of the fading short-wave radio channel by the Nakagami law [58]. The depth of fading is the ratio of the squared average power of the received signal to the variance of its instantaneous power and characterizes the properties of the random process defining the efficiency of control. The results of simulation experiments are stored in the database to reduce the time of analyzing efforts for searching of the efficiency region. Hypotheses on the efficiency region are used to successively form samples from the results of simulation experiments. After analyzing the samples and averaged values of

quality indices (5), the approximate solution of the stated multicriteria multiparametric optimization problem may be found. Samples of results of simulation experiments, for which conditions (6) corresponding to the formed hypotheses on the efficiency region are met, are created by the DBMS tools automatically. The DBMS constructor such as Access allows forming any request for sorting the complete sample of the results of simulation experiments and creating a partial sample by conditions (6) imposed on the quality indices.

To choose the convolutional code for the Viterbi decoder, the analytical estimates of the probability of erroneous decoding are formed after enumeration of all variants of positions of the erroneous symbol or a few symbols within the code combination. In channels with variable parameters, the probability of error per symbol may vary. Accurate estimates are replaced by approximate ones. Although the problem of discrete choice seems to be simple at first sight, the level of its complexity requires special methods of searching for the optimal solution to be applied [46, 48, 52]. The continuous input parameters, according to the model of continuous communication channel, include mathematical expectations m_x , m_y and the variances σ_x , σ_y of orthogonal components of the signal-to-noise ratio at the input of the receiver. The discrete parameters include the number of registers of the decoding device reg , the number of bits in the register of the decoding device k , and the number of summators n . The space of criteria includes the residual probability of the error per symbol $P_{res} = f(\mu(t), \tau_k, reg, k, n)$. From the consumer's point of view, it is the principal characteristic of the applied code or encoding method and depends, in addition, on the type of data transmission channel and the current noise environment. The data transmission speed $R = reg/n$ is introduced to characterize the properties of the code. The complexity of the encoding device and the decoding device can be represented as a discrete parameters function $f_{cod}(reg, k, n)$. This criterion can be estimated approximately by the maximal value of the order of the generating polynomial k . The free distance of the code d associated with the code structure and significantly influencing the value of the residual probability of error P_{res} also may be considered as criterion.

After change the encoding method or code parameters, it is need to create a new decoding device. In the simulation conditions, it means changing the using model with number i , i.e., changing the number of the checking variant. For choice the efficiency region in the space of continuous parameters, the

single-criterion four-parametric problem for four types of convolutional codes $\{v_i\}, i = \overline{1, I_0}$ is solved. After studying the space of parameters for the particular code with the number i the efficiency region $G_{v_i} \subset \mathfrak{S}_0$, $\mathfrak{S}_0 = \bigcup_{i=1}^{I_0} \mathfrak{S}_{v_i 0}$, $i = \overline{1, I_0}$, is determined, where $\mathfrak{S}_{v_i 0}$, $i = \overline{1, I_0}$ is the range of change of parameters of the code with the chosen number. Within G_{v_i} the inequality $P_{res\ v_i} \leq P_z$, bounding the residual probability of error by the given threshold value, should be fulfilled. Derived group of values G_{v_i} , $i = \overline{1, I_0}$ is used to determine the mutual efficiency region $G \subset \mathfrak{S}_0$, $G = \bigcap_{i=1}^{I_1} G_{v_i}$ for the subset of variants of codes or decoding methods $\{v_i\}, i = \overline{1, I_1}$, $I_1 \leq I_0$. Expertise helps to realize final choice.

Analysis of the results of simulation for the chosen set of codes shows, that encoding is efficient for the states of the communication channel, where the grouping coefficient is close to zero; i.e., the errors are independent, that is justified by the existing recommendations. The influence of the depth of fading or combinations of values of orthogonal components of the signal-to-noise ratio on the encoding efficiency is not detected. Thus, the region G_{v_i} includes vectors of parameters that correspond to such states of the communication channel, where the grouping coefficient is close to zero, i.e., the efficiency region has the form

$$r(\alpha(t)) = \frac{\left| \left(\sum_{i=1}^J l_i^2 \right) / J - (\tilde{p} T_D)^2 \right|}{\tilde{p} T_D} < \varepsilon, \quad \varepsilon \rightarrow 0, \quad (13)$$

where \tilde{p} is the mean probability of error per symbol in the obtained flow of errors, J is the number of simulated messages, and l_i is the number of errors within the interval of discretization T_D .

The third example deals with solving the problem of increasing the efficiency of phase-lock systems applied in navigation devices. This requires constructing special phase jumping control and registration devices. These are called jumping control blocks. I.A. Mymrin's simulation model of a three-stage jumping control block was used [48]. The functions of particular stages are as follows. The first stage detects that the phase-lock error signal becomes greater than the limit given preliminarily. The second stage tracks slow changes of the amplitude of the co phased component of the signal. The third stage has an alarm signal, which is the response to fast changes in the amplitude caused by external noise of the type of the reflected signal, deep fading, and additive noise. The simulation model of the jumping control block is constructed so that it gives one implementation of the random process that describes the appearance of all these types of noise. If the process is ergodic and the length of implementation is rather large, it includes comprehensive description of phenomena that cause jumping. Otherwise, we need to change the value of the fixed input variables and simulate other implementations.

The resulting alarm signal is formed by using the alarm signals of particular stages, i.e., three signals, generated by three stages in this scheme, determines the quality of operation of the jumping control block. The quality of operation of the jumping control block is estimated by probabilities of missing the jump, false alarm, correct fixation of intervals of phase jumping, and correct reception of signals when there are no jumps. In the process of simulation, statistical estimates for the above-mentioned probabilities are calculated for the implementation length of 500 intervals. Multiple simulations for different values of parameters of interferences allow estimating the average values of quality indices. The simulation statistical model takes into account the following types of noise: reflected signal, fading of the amplitude of the principal signal, fast change of the phase of the input signal at the expense of antenna dynamics, additive impulse, or continuous noise.

Based on analyzing the limit modes of jumping control block operation, the set of values of coefficients of jumping control block stages was chosen. They must be equal to 0.11, 3.33, 11.2. For it, the statistics for estimating the above listed probabilities 0, 0.414, 0.186, 0.4, respectively was obtained. By means of analyzing the algorithm of jumping control block operation and applying the methodology of studying dynamic stochastic systems using the LP_{τ} -search with averaging values of coefficients of the jumping control block stages was defined such that the probability of a false alarm was reduced.

The space of parameters for performing the LP_{τ} -search with averaging included values of the jumping control block coefficients participating in forming the threshold. Criteria for estimation its operation efficiency were chosen equivalent to statistical estimates of probabilities of missing the jump P_{loss} ; false alarm P_{false} ; correct fixation of phase jumping intervals $P_{correct}$; and the correct reception of signals without jumps P_{truth} . In the course of the LP_{τ} -search regions of change of values of parameters were found such that $P_{loss} \rightarrow 0$, $P_{false} \rightarrow 0$, $P_{correct} \rightarrow \min$, $P_{truth} \rightarrow \max$. The values of coefficients of jumping control block stages chosen from these regions ensure satisfactory operation and increase in the quality of secondary information processing. Because of low level of formalization of description for noise appearance conditions in the process of jumping control block operation, the hypothesis about the form of the function $r(\alpha, \varpi)$, that describes the efficiency region, wasn't formulated. The statistical estimates of quality indices took the values 0, 0.214, 0.186, 0.6, respectively. To attain improved values of quality indices, the new set of values of coefficients of the jumping control block stages should fall within the multidimensional parallelepiped:

$$0,62 \leq \alpha_1 \leq 2,05; \quad 2,9 \leq \alpha_2 \leq 11,38; \quad 0,038 \leq \alpha_3 \leq 0,875 .$$

The quantity of false labels is reduced almost twice as compared to the missing jumps statistics given above and named as standard.

CONCLUSION

Obviously the way of comparison to references represented as patterns stored in memory is not applicable for the problem of searching the efficiency region in the multidimensional space of continuous parameters. For discrete parameters, individual solutions may be found, but it is no possible create a universal procedure of solving the stated problem. If patterns are represented by vectors with real components, the class of patterns is treated as a cluster separation by the set of features. Simple recognition methods (such as the minimal distance principle) are used if clusters do not overlap. Otherwise, it is needed to increase the quantity and accuracy of measurements and apply complicated methods clusters separation (likelihood functions, Bayes pattern

classifiers, trained pattern classifiers, stochastic approximation methods, potential function methods, etc.) [61].

Clusters will be overlap for the problem under consideration. The most universal method is decomposition into classes based on common properties of classes. Common properties are mapped onto the set of attributes peculiar to such patterns. Difficulties in creating recognition algorithms arise when attributes ensuring the difference between the classes are selected. The complete set of differentiating attributes is not formed at once. As the procedure of comparison with respect to attributes improves as a result of accumulating statistical material, the totality of attributes may change. The quality of recognition algorithms depends on methods of choosing attributes and can be improved by additional procedures of selection from the totality of attributes. Only this method may be used to solve the stated problem. It allows checking and estimating characteristic attributes, including additional “indicators” in them, and creating a universal procedure of comparing with respect to attributes. Such procedure does not require the long deep analysis needed when empirical procedures are used and speeds up the search of the efficiency region [51].

REFERENCES

- [1] N. Metropolis and S. Ulam, “The Monte–Carlo Method,” *J. Am. Stat. Assoc.* 44 (247), 335–341 (1949).
- [2] V. A. Starosel’skii, “On Functional Optimization Set by Statistic Model,” *Ekonom. Stat. Metody* 3 (3), 460–461 (1967).
- [3] V. F. Dem’yanov and L. V. Vasil’ev, *Nondifferentiable Optimization* (Nauka, Moscow, 1981) [in Russian].
- [4] A. D. Tsvirkun, V. K. Akinfiev, and V. A. Filippov, *Imitational Simulation in Synthesis Problems for Complex Systems Structure (Optimization–Imitation Approach)* (Nauka, Moscow, 1985) [in Russian].
- [5] G. M. Antonova and A. D. Tsvirkun, “Optimization Imitation Simulation for Solving Optimization Problems of Modern Complex Control Systems,” *Probl. Upravlen.*, No. 5, 19–27 (2005).
- [6] A. A. Zhiglyavskii, *Mathematical Theory of Global Random Search* (Leningrad Univ. Leningrad, 1985) [in Russian].
- [7] A. A. Zhiglyavskii and A. G. Zhilinskias, *Methods of Searching the Global Extremum* (Nauka, Moscow, 1991) [in Russian].

-
- [8] S. M. Ermakov and G. A. Mikhailov, *Statistical Simulation*, 2nd ed. (Nauka, Moscow, 1982) [in Russian].
 - [9] J. H. Halton, "On the Efficiency of Certain Quasi Random Sequences of Points in Evaluating Multi Dimensional Integrals," *Num. Math.* 2 (2), 84–90 (1960).
 - [10] N. M. Korobov, "On Approximate Computation of Multiple Integrals," *Dokl. Akad. Nauk SSSR* 124 (6), 1207–1210 (1959).
 - [11] N. M. Korobov, *Numerical–Theoretical Methods in Approximate Analysis* (Fizmatgiz, Moscow, 1963) [in Russian].
 - [12] I. M. Sobol', *Multidimensional Quadrature Formulas and Haar Functions* (Nauka, Moscow, 1969) [in Russian].
 - [13] I. M. Sobol' and R. B. Statnikov, *How to Choose Optimal Parameters in Multicriteria Problems* (Nauka, Moscow, 1981) [in Russian].
 - [14] I. M. Sobol', "How to Estimate Accuracy of Elementary Multidimensional Search," *Dokl. Akad. Nauk SSSR* 266 (3), 569–572 (1982).
 - [15] I. M. Sobol', "On Functions That Meet Lipschitz Condition in Multidimensional Problems of Calculus Mathematics," *Dokl. Akad. Nauk SSSR* 293 (6), 1314–1319 (1987).
 - [16] I. M. Sobol', "Uniformly Distributed Sequence with Complementary Uniformity Property," *Zh. Vychisl. Mat. Mat. Fiz.* 16 (5), 1332–1337 (1976).
 - [17] N. A. Artemova, A. F. Khairullin, and O. B. Khairullina, "How to Generate Optimal Meshes in Multiply Connected Domains with Complex Topology at Multiprocessor Computers," in *Algorithms and Software for Parallel Calculations* (Yekaterinburg, 1998), Issue 2, pp. 22–38 [in Russian].
 - [18] G. S. Ganshin, "How to Calculate Maximum for a Function of Several Variables," *Kibernetika*, No. 2, 61–63 (1983).
 - [19] N. F. Zaliznyak and A. A. Ligun, "On Optimal Strategies for Searching the Function Global Maximum," *Zh. Vychisl. Mat. Mat. Fiz.* 18 (2), 314–321 (1978).
 - [20] V. V. Ivanov, S. K. Girlin, and V. A. Lyudvichenko, "Problems and Results of Global Search for Smooth Functions," *Vopr. Kibernet.* 122, 3–13 (1985).
 - [21] S. I. Martynenko, "Universal Multi–Grid Technique for Numerical Simulation of Partial Differential Equations at Structured Grids," *Vychisl. Metody Programm.* 1 (1), 85–104 (2000).

-
- [22] S. I. Martynenko, "Software for Universal Multigrid Technique: Blocks and Diagnostic Instruments," *Vychisl. Metody Programm.* 2 (2), 181–186 (2001).
 - [23] V. N. Nefedov, "How to Calculate the Global Maximum of Function of Several Variables for a Set Given by Limitations of Inequality Type," *Zh. Vychisl. Mat. Mat. Fiz.* 27 (1), 35–51 (1987).
 - [24] O. B. Khairullina, A. F. Khairullin, and N. A. Artemova, "The Way to Calculate Optimal Grids of High Dimensionality in Multiply-Connected Domains by Using MVS-100 Distributed Memory," in *Algorithms and Software for Parallel Calculations* (Yekaterinburg, 1999), Issue 3, pp. 239–251 [in Russian].
 - [25] A. F. Khairullin and O. B. Khairullina, "Automatic Generation for Initial Approximation of Curvilinear Mesh," in *Algorithms and Software for Parallel Calculations* (Yekaterinburg, 2000), Issue 4, pp. 273–286 [in Russian].
 - [26] O. V. Ushakova, "Parallel Algorithm and Program for Generating Optimal Adaptive Grids," in *Algorithms and Software for Parallel Calculations* (Yekaterinburg, 1995), Issue 1, pp. 182–192 [in Russian].
 - [27] H. Niederreiter, "Quasi-Monte-Carlo Methods and Pseudo Random Numbers," *Bull. Amer. Math. Soc.* 84 (6), 957–1041 (1978).
 - [28] H. Niederreiter and K. McCurley, "Optimization of Functions by Quasi-Random Search Methods," *Comput.* 22, 119–123 (1979).
 - [29] I. M. Sobol', *Monte-Carlo Numerical Methods* (Nauka, Moscow, 1973) [in Russian].
 - [30] V. M. Chernov, "Canonical Number Systems and Two-Dimensional Uniform Distribution," in *Proc. 7th Int. Conf. on Pattern Recognition and Image Analysis: New Information Technologies PRIA-7-2004* (St. Petersburg, Oct. 18–23, 2004), Vol. 1, pp. 185–188.
 - [31] V. D. Liseikin, "The Ways for Generating Structural Adaptive Grids: Review," *Zh. Vychisl. Mat. Mat. Fiz.* 36 (1), 3–41 (1996).
 - [32] N. A. Artemova, A. F. Khairullin, and O. B. Khairullina, "How to Generate Optimal Meshes in Multiply Connected Domains with Complex Topology at Multiprocessor Computers," in *Algorithms and Software for Parallel Calculations* (Yekaterinburg, 1998), Issue 2, pp. 22–38.
 - [33] Zh. M. Sakhabutdinov, G. A. Petrov, and S. V. Maigurova, "How to Construct and Optimize 3D Curvilinear Meshes," in *Problems of Atomic Science and Technique. Series Mathematical Simulation for Physical*

- Processes* (NII Upr. ekon. i informatsii, Moscow, 1989), Issue 1, pp. 9–18.
- [34] L.E. Eriksson, “Practical Three-Dimensional Mesh Generation Using Transfinite Interpolation,” *SIAM J. Sci. Stat. Comput.* 6 (3), 712–742 (1985).
- [35] O. A. Borovikov, “How to Optimize Nodes Distribution for Computational Mesh,” *Gazodinam. Teploobmen*, No. 8, 102–112 (1984).
- [36] Y. H. Oh, *An Analytical Transformation Technique for Generating Uniformly Spaced Computational Mesh* (WNGG–NASA, 1980).
- [37] R. D. Russel and J. Christiansen, “Adaptive Mesh Selection Strategies for Solving Boundary Value Problems,” *Numer. Anal.* 15, 59–80 (1978).
- [38] A. M. Sorokin, “How to Generate the Computational Meshes for the Problems on Flat Transonic Flows,” *Uch. Zapiski TsAGI* 17 (3), 115–120 (1986).
- [39] A. M. Winslow, “Adaptive Mesh Zoning by the Equipotential Method,” *Technical Report UCID 19062* (1981).
- [40] D. A. Anderson, “Equidistribution Schemes. Poisson Generators and Adaptive Grids,” *Appl. Math. Comput.* 24 (3), 211–227 (1987).
- [41] K. L. Bogomolov, L. M. Degtyarev, and V. F. Tishkin, “Variation Method for Generating the High Aspect Uniform Adaptive Meshes,” *Mat. Model.* 13 (5) (2001).
- [42] G. H. Klopfer and D. D. McRae, “The Nonlinear Modified Equation Approach to Analyzing Finite Difference Schemes,” *AIAA Paper*, No. 81–1029 (1981).
- [43] J. P. Boris, “New Directions in Computational Fluid Dynamics,” *Ann.*
- [44] A. E. Andrews, “Progress and Challenges in the Application of Artificial Intelligence to Computational Fluid Dynamics,” *AIAA J.* 26 (1) 40–46 (1988).
- [45] G. M. Antonova, “Optimizing Imitational Approach for Choosing Functioning Algorithms for Data Transfer Systems,” *Avtomat. Telemekhan.*, No. 9, 167–174 (1996).
- [46] G. M. Antonova, “Application of $LP\tau$ Optimization in the Frames of Optimizing Imitational Approach for Choosing the Noise Resistant Correcting Codes,” *Avtomat. Telemekhan.*, No. 9, 162–168 (1999).
- [47] G. M. Antonova, “Parallel Algorithm for Researching Dynamic Stochastic Processes or Systems, Presented by Imitation Models,” in *Proc. Int. Conf. Parallel Computations and Control Problems (PACO'2001)* (Moscow, 2001), Vol. 3, pp. 30–41.

-
- [48] G. M. Antonova, "LP τ –Search with Averaging as a New Technique for Optimal Decisions Making," Appendix to "Informatsionnye Tekhnologii" Zh., No. 6, 1–24 (2001).
 - [49] G. M. Antonova, "The Way to Optimize the System of Phase Tuning in Net Satellite Radionavigation System by Using LP τ –Search with Averaging," in *Proc. 2nd Int. Conf. on Control Problems* (Institut problem upravleniya, Moscow, 2003), pp. 91–98.
 - [50] G. M. Antonova and A. Yu. Baikov, "LP τ –Search with Averaging Using for Parameters Definition in Powerful Vacuum Resonator UHF-Devices of O-Type," in *Proc. 5th Int. Conf. "Systems Identification and Control Problems"* (Institut problem upravleniya, Moscow, 2006), pp. 823–837.
 - [51] G. M. Antonova, "Application of Recognition Procedures for Estimating the Efficiency Regions for LP τ –Search with Averaging," *Pattern Recogn. Image Anal.* 16 (4) (2006).
 - [52] G. M. Antonova, *Uniform Probing Mesh Methods for Investigating and Optimizing Dynamic Stochastic Systems* (Fizmatlit, Moscow, 2007) [in Russian].
 - [53] G.M. Antonova, "Optimization-Simulation with Continuous Criteria Using," in *Proc. 18th World Congress of International Federation of Automatic Control (IFAC'11)* (August 27-September 03, 2011. Milano, Italy), pp.5543-5548.
 - [54] G. M. Antonova, "Identification Approach for Investigating Stochastic Systems Presented by Imitation Models," in *Proc. 2nd Int. Conf. "Systems Identification and Control Problems"* (Moscow, 2003), pp. 2125–2138.
 - [55] J. Tou and R. Gonzales, *Pattern Recognition Principles* (Addison_Wesley, 1974; Mir, Moscow, 1978).
 - [56] Z. M. Kanevskii, M. I. Dorman, B. V. Tokarev, and V.V. Kretinin, *Information Transition with Feedback* (Svyaz', Moscow, 1976) [in Russian].
 - [57] D. D. Klovskii, *Discrete Messages Transmission with Radio Channels* (Svyaz', Moscow, 1969) [in Russian].
 - [58] M. Nakagami, *The m Distribution as General Formula of Intensity Distribution of Rapid Fading. Statistical Methods in Radio Wave Propagation* (New York, 1960).
 - [59] *Special Functions: Handbook*, Ed. by M. Abramovits and I. Stigan (Nauka, Moscow, 1979) [in Russian].

- [60] L. N. Bol'shev and N. V. Smirnov, *Tables for Mathematical Statistics* (Nauka, Moscow, 1983) [in Russian].
- [61] V. P. Bogomolov, A. P. Vinogradov, V. A. Voronchikhin, Yu. I. Zhuravlev, N. N. Katerinokhina, S. B. Larin, V. V. Ryazanov, and O. V. Sen'ko, *Software LOREG: Recognition Algorithms Based on Voting for Sets of Logical Regularities* (VTs RAN, Moscow, 1998) [in Russian].

Chapter 4

PRACTICAL USAGE OF ALGORITHMIC PROBABILITY IN PATTERN RECOGNITION

*Alexey S. Potapov**

AIDEUS and National Research University of Information Technology,
Mechanics and Optics, Russia

ABSTRACT

Solomonoff universal induction based on Algorithmic Probability (ALP) can be considered as the general theoretical basis for machine learning and, in particular, pattern recognition. However, its practical application encounters very difficult problems. One of them is incomputability caused by usage of the Turing-complete solution space. The Minimum Description Length (MDL) and the Minimum Message Length (MML) principles can be considered as simplified derivations of ALP applied to Turing-incomplete solution spaces. The MDL and MML principles have been successfully used to overcome overlearning and to enhance different pattern recognition methods including construction of nonlinear discrimination functions, support vector machines, mixture models, and others.

However, restriction of the solution space is not the only simplification in the MDL/MML approaches. All possible models of data are used to calculate ALP, while the only one best model is selected on the base of the MDL/MML principle. In this chapter, the possibility to

* Corresponding author: Alexey S. Potapov. E-mail: potapov@aideus.com.

utilize Turing-incomplete version of ALP in the practical tasks of pattern recognition is considered. This gives theoretically and experimentally grounded approach to use “overcomplex” models (similar to compositions of classifiers or mixtures of experts) without the risk of overlearning, but with better recognition rates than that of minimum description length models.

It is impossible to sum over all models even in Turing-incomplete model spaces. Thus, it is necessary to select some number of models, which should be taken into account. These models can correspond to different extrema of the MDL criterion. For example, if models can have different number of parameters than the best model for each number of parameters can be taken into consideration. Models constructed on some subsets of a training set can be used for calculating ALP in the case of families of models with fixed number of parameters.

Some concrete applications of the ALP-based approach are considered. Its difference from finite mixtures and possible connection with boosting are discussed.

1. INTRODUCTION

Many different pattern recognition methods exist. This diversity is partially caused by the variety of recognition tasks. However, another reason consists in absence of really comprehensive theory of pattern recognition. Even the most mathematically sound methods and theories contain heuristic elements. The problem of overlearning (overfitting) is one of negative consequences of the insufficient theoretical foundations.

Pattern recognition tasks can be interpreted as particular tasks of inductive inference, which consists in search for regularities in observation data (in construction of models of data sources). Methods of inductive inference contain such general components as the model space, decision criterion, optimization or search algorithm. Methods of pattern recognition also include these components in explicit or implicit form.

Any negative effect in pattern recognition is the result of selecting an inappropriate model, because of bad criterion, narrow model space or weak search algorithm. The overlearning effect is especially connected with the decision criterion. One of the most widely accepted correct criteria is Bayes' criterion. However, its usage encounters the well-known problem of prior probabilities [1]. Theory of probability allows one to infer posterior probabilities from prior probabilities, but doesn't tell us, how to introduce the very initial probabilities. Different heuristic techniques or semi-theoretical

criteria are frequently introduced. For example, overlearning of artificial neural networks is typically prevented by restricting the training time.

The most common observation consists in the fact that more complex models should be penalized, because they have more possibilities to fit the data. Such particular criteria as An Information Criterion [2] or Bayesian Information Criterion [3] are well-known. Similar ideas are summarized in the Minimum Description Length (MDL) [4] and Minimum Message Length (MML) [5] principles, which state that the best model to describe the data is the model that provides the minimum value of the sum of the length of the model and the length of the data described with the help of this model. The description length of the encoded data is usually estimated as its negative log likelihood. In practice, the length of the model is calculated using some heuristic coding scheme. For example, if the model is parametric than its length will be proportional to the number of parameters and the number of bits necessary to encode each parameter. Not only does this approach allow for successful practical applications (e.g. [6]–[8]), but also it has strong basis in algorithmic information theory, in which the notion of probability is derived from the amount of information defined on the base of pure combinatorial considerations.

However, incorrect generalization is not necessarily accompanied by overlearning. It can be caused simply by absence of the appropriate model in the used model space. Algorithmic information theory also provides us the most wide model space. This is the Turing-complete model space that contains all algorithms as models of possible data sources. Usage of the Turing-complete model space with prior probabilities defined by lengths of algorithms results in the universal induction/prediction method [9, 10] based on Algorithmic Probability (ALP). Existing practical methods are limited in their generalization capabilities, because they rely on the restricted model spaces, which don't contain all possible regularities. Unfortunately, direct search in the Turing-complete model space is computationally implausible.

The representational MDL (RMDL) principle was recently introduced [11] as an extension of the MDL principle that makes possible to take into account dependence of the model optimality criterion from prior information given in data representation. The RMDL principle gives criteria for automatic optimization of data representations, and partially bridges the gap between the theoretically ideal induction methods based on algorithmic complexity and practical applications of the MDL principle relied upon heuristic coding schemes.

Completeness of the model space is the essential issue, but there is another difference between the MDL principle that serves for selecting one best model and ALP that implies the usage of all possible models simultaneously [10]. Thus, it would be interesting to consider possible practical versions of ALP or to extend the practical MDL methods with the use of multiple models.

In this chapter, some methodological issues of adoption of Bayes' and Minimum Description Length criteria will be overviewed, and then techniques for the practical usage of Algorithmic Probabilities will be investigated.

2. BAYES' CRITERION

One of the most widely used mathematical criteria in inductive inference is based on Bayes' rule:

$$P(H | D) = \frac{P(H)P(D | H)}{P(D)} \quad (1)$$

where $P(H | D)$ is the posterior probability of the model H with the given data D ; $P(H)$ and $P(D)$ are the prior probabilities, and $P(D | H)$ is the likelihood of the data D with the given model H .

Bayes' rule can be directly applied to the classification problem. Let D be one pattern, and H be one of the classes. The most probable class for the given pattern can be easily selected maximizing $P(H | D)$, if the probability density distribution $P(D | H)$ of patterns within each class and the unconditional probabilities $P(H)$ are known.

Learning in statistical pattern recognition consists in inducing probability distributions on the base of some training set $\{d_i, h_i\}$, where d_i is the i -th pattern, and h_i is its class label. The prior probabilities $P(H)$ can be estimated from frequencies of each class in the training set. The distribution $P(D | H)$ should be represented as an element of some family $P(D | H, \mathbf{w})$, where \mathbf{w} is an indicator (e.g. parameter vector) of specific distribution. Using Bayes' rule and supposing independence of patterns one can obtain:

$$P(\mathbf{w} | D) = \frac{P(\mathbf{w}) \prod_i P(d_i | h_i, \mathbf{w})}{P(D)} \quad (2)$$

The values of $P(d_i | h_i, w)$ can be explicitly calculated for the specific distribution defined with w . However, there is a problem with evaluation of the prior probabilities $P(w)$. In order to specify these probabilities correctly one needs many training sets, for each of which true probability should be known. It is impossible, because such true probabilities are unknown even for human experts, who construct training sets.

Many researchers prefer to ignore prior probabilities and to use the maximum likelihood (ML) approach. The same result will be obtained if one supposes that the prior probabilities are equal. This supposition is evidently incorrect, because the prior distributions in the case of infinite model spaces become non-normalized.

In practice, it leads to the overlearning problem. Consider mixture Gaussian models as an example. The likelihood of data will be maximized for the maximum number of components in the mixture leading to the degenerated distribution.

The same overfitting effect also appears in the task of regression. For example, an attempt to find a polynomial that fits the given points with minimum error (maximum likelihood) will result in the polynomial with maximum degree that follows all errors in the data and possesses no generalization and extrapolation capabilities. The oversegmentation effect of the same origin appears in various segmentation tasks [12]: models with more segments will be more precise.

As it is pointed out in [10, 1], the problem of prior probabilities is the fundamental one. It is connected with some paradoxes in inductive inference such as Goodman's "Greu emerald paradox" (greu emeralds are green before some future date and blue after it). The paradox consists in the fact that observational data show the same evidence for emeralds to be green or greu.

Many criteria with heuristically introduced penalty for model complexity exist. And still, new criteria are being invented for particular tasks of machine learning. At the beginning, let's consider practical solutions.

3. PRACTICAL MINIMUM DESCRIPTION LENGTH PRINCIPLE

The most plausible solution of the problem of prior probabilities comes from the information-theoretic approach. Bayes' rule can be rewritten as follows.

$$\begin{aligned}
P(H | D) &\propto P(D | H)P(H) \Rightarrow \\
-\log_2 P(H | D) &\propto -\log_2 P(D | H) - \log_2 P(H) \Rightarrow \\
I(H | D) &\propto I(D | H) + I(H),
\end{aligned}$$

where I is the amount of information. Thus, maximization of the posterior probability corresponds to minimization of the amount of information. Similarly,

$$P(\mathbf{w} | D) \propto P(\mathbf{w}) \prod_i P(d_i | h_i, \mathbf{w}) \Rightarrow I(\mathbf{w} | D) \propto I(\mathbf{w}) + \sum_i I(d_i | h_i, \mathbf{w}) \quad (3)$$

Classical Shannon information theory states that the amount of information in a message can be calculated using its probability. It is easy to estimate minus log likelihood of the data for any distribution $P(d | h, \mathbf{w})$, but $P(\mathbf{w})$ is usually unknown. However, the value of $I(\mathbf{w})$ can be estimated directly within some coding scheme. For example, if \mathbf{w} is a parameter vector, its elements can be coded with some number of bits (precision) yielding the amount of information $I(\mathbf{w})$. This idea gives rise to rather natural technique for specifying prior probabilities. It is unified within the Minimum Description Length principle that can verbally be formulated as [13]: the best model of the given data source is the one which minimizes the sum of

- the length, in bits, of the model description;
- the length, in bits, of data encoded with the use of the model.

Usage of the MDL principle in pattern recognition is now widespread. For example, it was applied to choose appropriate complexity of nonlinear discrimination functions [14], support vector models [15], and the number of components in mixture models [16]. Consider the task of pattern recognition in the case of two classes, and generalized discrimination functions defined as

$$\kappa(\mathbf{x} | \mathbf{w}) = \sum_{i=1}^n w_i y_i(\mathbf{x}) = \mathbf{w} Y(\mathbf{x}) \quad \varphi(\mathbf{x}) = \begin{cases} 1, \kappa(\mathbf{x} | \mathbf{w}) < 0, \\ 2, \kappa(\mathbf{x} | \mathbf{w}) > 0, \end{cases} \quad (4)$$

where the pattern d defined as the feature vector \mathbf{x} , the vector \mathbf{w} is the parameter vector of the discrimination function $\kappa(\mathbf{x} | \mathbf{w})$, $y_i(\mathbf{x})$ is i -th generalized feature that is deterministically computed for the pattern \mathbf{x} .

In the discrimination function approach the MDL criterion has the form $L(\{h_i\}_{i=1}^n, \mathbf{w} | \{\mathbf{x}_i\}_{i=1}^n)$ meaning that only class labels are considered as the given data to be encoded using prior information about feature values, and each model is specified by its parameter vector.

Within some simplification assumptions one can obtain

$$L(\{h_i\}_{i=1}^n, \mathbf{w} | \{\mathbf{x}_i\}_{i=1}^n) = \frac{M}{2} \log_2 n + \frac{n}{2} \log_2 \frac{\varepsilon^2(\mathbf{w})}{n},$$

$$\varepsilon^2(\mathbf{w}) = \sum_{i=1}^n [z_i - \mathbf{w}Y(\mathbf{x})]^2, \quad (5)$$

where M is the number of components in the parameter vector \mathbf{w} , and $z_i = -1$ if i -th vector belongs to the first class, and $z_i = 1$ if i -th vector belongs to the second class. The equation (5) can be used to select among the discrimination functions with different number of parameters.

As an example, polynomial discrimination functions with different number of parameters (degrees of polynomials) were found for the set of patterns shown in Figure 1. Patterns belong to two classes indicated using different labels (circles and crosses).

Characteristics of the discrimination functions with minimum round-mean-square error $\varepsilon^2(\mathbf{w})$ for different number of parameters M can be found in Table 1. It can be seen that the solution with the minimum description length has also the best recognition rate on the new patterns ($\%_{\text{test}}$), which is not directly corresponds to the recognition rate on the learning sample ($\%_{\text{learn}}$).

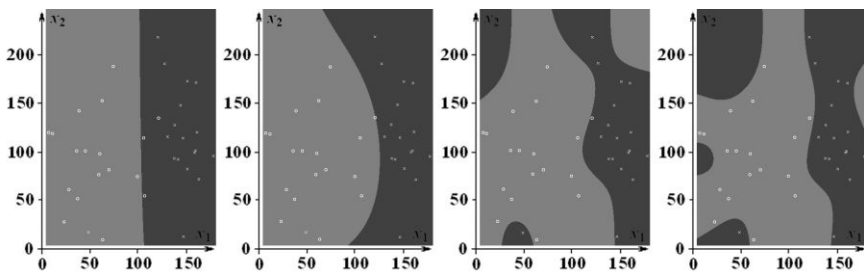


Figure 1. Discrimination functions with 4, 9, 16, and 25 parameters.

Table 1. Comparison of discrimination functions of different complexity

№	M	% _{learn}	% _{test}	L , bits
1	4	11.1	4,3	31.2
2	9	2.8	4.0	30.9
3	16	2.8	10.1	51.7
4	25	0.0	19.4	62.0

This is the traditional way to use the MDL principle in order to avoid overlearning that appears very fruitful in practice. At the same time, heuristic coding schemes specifying restricted subfamily of models are to be introduced for each particular task. Search within such subfamily can be easy, but only a priori restricted set of regularities can be captured.

4. ALGORITHMIC COMPLEXITY AND PROBABILITY

The MDL principle was formulated above without explicit theoretical foundations. To be more precise, the technique for estimating $I(w)$ on the base of the length of encoded values w within some coding scheme was not proven. The best theoretical foundation for defining information quantity without usage of probability can be found in algorithmic information theory.

Consider the following notion of (prefix) algorithmic complexity of a binary string β introduced by A.N. Kolmogorov [17]:

$$K_U(\beta) = \min_{\alpha} [l(\alpha) | U(\alpha) = \beta] \quad , \quad (6)$$

where U is a universal Turing machine (UTM), α is an arbitrary algorithm (program for UTM), and $l(\alpha)$ is its length. The expression $U(\alpha)=\beta$ means that the program α being executed on the UTM U produces the string β . In accordance with this notion, information quantity (defined as algorithmic or Kolmogorov complexity) contained in the data (string) equals to the length of the shortest program that can produce these data. In contrast to Shannon theory, this notion of information quantity relies not on probability, but on pure combinatorial assumptions.

The MDL principle can be derived from the Kolmogorov complexity if one divides the program $\alpha=\mu\delta$ into the algorithm itself (regular component of the model) μ and its input data (random component) δ :

$$K_U(\beta | \mu) = \min_{\delta} [l(\delta) | U(\mu\delta) = \beta], \quad K_U(\beta) = \min_{\mu} [l(\mu) + K_U(\beta | \mu)] \quad (7)$$

where $K(\beta | \mu)$ is a specific form of the conditional complexity, which is defined for the string β relative to the given model μ .

Consequently, one would like to choose the best model by minimizing the model complexity $l(\mu)$ and the model “precision” $K(\beta | \mu) = l(\delta)$ simultaneously, where the string δ can be considered as deviations of the data β from the model μ :

$$\mu^* = \arg \min_{\mu} [l(\mu) + K_U(\beta | \mu)] \quad (8)$$

Algorithmic information theory not only clarifies the problem of the model selection criterion, but also specifies the most universal model space. Indeed, any (computable) regularity presented in the given data can be found in this model space.

However, the equation (8) cannot be applied directly. The main reason is the problem of search in the Turing-complete model space. That’s why loose definitions of the MDL principle are used in practice.

Another problem consists in “excessive universality” of the criterion (8). It incorporates the universal prior distribution of model lengths $l(\mu)$ or prior probabilities $2^{-l(\mu)}$. This distribution is independent of the specific task to be solved implying that any available information should be directly included in the string β . For example, it is impossible to use the equation (8) in order to select the best class h for the single pattern d . An attempt to calculate $l(h) + K_U(d | h)$ instead of $l(h) + l(d | h)$ will be unsuccessful, because if one takes d and h as individual data strings, there will be no mutual information in them. Available information in the pattern recognition task is the training set $\{d_i, h_i\}$. One can try to find the best model for these data

$$\mu^* = \arg \min_{\mu} [l(\mu) + K_U(\{d_i, h_i\} | \mu)] \quad (8)$$

If there is a statistical connection between patterns d_i and their labels h_i , it can be captured by μ^* , because algorithmic models can also contain optimal codes for random variables. This statement of the induction task is possible (although additional prior information is difficult to include in it), but how to

recognize new patterns on the base of such μ^* ? The simplest yet powerful idea is to explicitly consider the pattern recognition problem as the mass problem.

5. BETWEEN THEORETICAL AND PRACTICAL MDL

As it was pointed out, the MDL principle helps to partially solve in practice such problems as overfitting, overlearning, oversegmentation, and so on. However, it can also be seen that coding schemes for description length estimation are introduced heuristically in the MDL-based methods. Ungrounded coding schemes are non-optimal and non-adaptive (independent of the given data). These schemes define algorithmically incomplete model spaces that cause corresponding methods of pattern recognition to be fundamentally restricted. Thus, there is a large gap between the theoretical MDL principle with the universal model space and prior probability distribution and its practical applications.

Tasks of inductive inference should be considered as mass problems in order to bridge this gap. Indeed, trained classifiers are usually applied independently for each pattern. In most cases the algorithmic complexity of concatenation of some data strings $\beta_1\beta_2...\beta_n$ is strictly smaller than the sum of their individual algorithmic complexities:

$$K_U(\beta_1\beta_2...\beta_n) \ll \sum_{i=1}^n K_U(\beta_i) \quad (9)$$

Moreover, the universal prior probability distribution appears to be dependent on the choice of Universal Turing Machine that can be considered as additional theoretical difficulty. Usually, this difficulty is assumed to be unessential, because a constant string v exists for any two UTMs U and V such that $(\forall \alpha) U(v\alpha) = V(\alpha)$. In other words, $(\forall \beta) K_U(\beta) \leq K_V(\beta) + C$, that is algorithmic complexities of any data string on two different UTMs differ only by a constant that doesn't depend on the given data. Consequently, influence of difference between UTMs will decrease with increase of data volume, and equivalent models will be selected.

However, the constant C may be very large in practice. Moreover, difference in algorithmic complexities will be unbounded in mass problems of pattern recognition, because only the following inequality will hold

$$\sum_{i=1}^n K_U(\beta_i) \leq \sum_{i=1}^n K_V(\beta_i) + nC \quad (10)$$

It can now be seen, why heuristic coding schemes are used in practical applications of the MDL principle in the tasks of pattern recognition instead of universal model space defined by some UTM. Not only does search in algorithmically complete space lead to computational problems, but selection of a specific coding scheme exerts great influence on the model quality criterion and consequently on efficiency of the corresponding method.

These difficulties are the most crucial for pattern recognition tasks as mass problems, therefore difference between the left and right parts of equation (9) and equation (10) should be minimized. Sum of algorithmic complexities of data strings (sum of lengths of their independent descriptions) is much larger than algorithmic complexity of their concatenation (length of their joint description), because these sets contain mutual information. This mutual information should be removed from descriptions of individual data strings, and should be considered as prior information in corresponding methods. This implies usage of conditional algorithmic complexity. Indeed,

$$K_U(\beta_1\beta_2...\beta_n) \approx \min_S \left(\sum_{i=1}^n K_U(\beta_i | S) + l(S) \right), \quad (11)$$

where the conditional algorithmic complexity can be calculated as $K_U(\beta_i | S) = \min_{\mu} (l(\mu) | U(S\mu) = \beta_i)$, S is some string, and $l(S)$ is its length [18, 11].

It can be shown that $(\forall U, V, S)(\exists S')(\forall \beta) K_U(\beta | S') = K_V(\beta | S)$. Let $S' = vS$, where v is interpreter of V on U , then $(\forall \alpha) U(S'\alpha) = U(vS\alpha) = V(S\alpha)$. Consequently,

$$\sum_{i=1}^n K_U(\beta_i | S') = \sum_{i=1}^n K_V(\beta_i | S) \quad (12)$$

Since S can be interpreted as an algorithm (some program for UTM), which produces any given data string from its description, the algorithm S

precisely fits the verbal notion of information representation formulated by David Marr [19]. Therefore, the following more strict definition can be given.

Definition. The program S for the UTM U can be called *representation* of the collection of data strings (patterns, images, etc.) $B = \{\beta_1, \dots, \beta_n\}$, if $(\forall \beta \in B)(\exists \mu, \delta \in \{0,1\}^*) U(S\mu\delta) = \beta$. The string $\mu\delta$ can be called *description* of β within the representation S . This description consists of the regular μ and the random δ components.

If the data string is described within some representation, then the mentioned above difficulties will be eliminated. In particular, the choice of UTM will not influence on the model selection for the specific data string because of equation (12). We will omit indication of the specific UTM and write $K_S(\beta)$ instead of $K_U(\beta | S)$, because S can be treated as a virtual machine emulated on the machine U . It should be noted that some representation S usually specifies algorithmically incomplete model space, and the complexity $K_S(\beta)$ turns out to be computable in practice (in contrast to the complexity $K_U(\beta)$).

The formal notion of representation can be used to extend the MDL principle on mass problems giving the representational MDL principle that consists of two parts [18].

- 1 The best model μ of data β *within the given representation S* is the model, for which the sum of the following components is minimized:
 - the model length $l(\mu)$;
 - the length of data described with the use of model $K_S(\beta | \mu)$.

Selection criterion and the best model can be calculated as

$$L_S(\beta, \mu) = K_S(\beta | \mu) + l(\mu) \quad \text{and} \quad \mu^* = \arg \min_{\mu} L_S(\beta, \mu) \quad (13)$$

- 2 The best representation S for the collection of data strings $B = \{\beta_1, \dots, \beta_n\}$ is the representation, for which the sum of the following components is minimized
 - the length of representation $l(S)$;
 - the sum of lengths of data strings described within the representation $\sum_{i=1}^n K(\beta_i | S)$.

Selection criterion and the best representation can be calculated as

$$L(B, S) = l(S) + \sum_{i=1}^n K_s(\beta_i) \quad S^* = \arg \min_S L(B, S) \quad (14)$$

The RMDL principle specifies dependence of the model quality criterion from the used representation (description language), and also gives criteria for optimization of the representation itself. Thus, theoretical grounds for representation optimization depending on the problem domain are obtained instead of heuristic selection of coding schemes occurring in practical implementation of the MDL criterion.

Of course, the RMDL principle doesn't give complete solution of the problem of automatic representation optimization, because the best S should be selected from the whole space of algorithms. It gives only criteria for their comparison that can be practically used only with efficient representation search (or generation) procedures. Nevertheless, this principle can be used for objective comparison of hand-crafted representations (question of optimality and bounds of applicability of heuristic coding schemes was not even stated), and also for automatic optimization of representations within their simple families. The RMDL principle is more interesting in application to image analysis problems, because the idea of constructing general models describing pattern sets is very natural in pattern recognition. Nevertheless, the RMDL principle not only shows that the existing pattern recognition methods use particular representations specifying algorithmically incomplete model spaces, but it also reveals possibilities of weakening these shortcomings.

The RMDL principle helps to build "synthetic" pattern recognition systems [20], in which the choice of the best particular classifier is carried out on the base of the description length criterion. Let's consider extension of family of representations in the Gaussian mixture method for the task of clustering as an example. The Gaussian mixture is represented in the form

$$p(\mathbf{x} | \mathbf{w}) = \sum_{i=1}^d P_i p(\mathbf{x} | C_i, \mathbf{y}_i) \quad (15)$$

where \mathbf{x} is the pattern (feature vector), P_i is the weight of i -th component of the mixture, $p(\mathbf{x} | C_i, \mathbf{y}_i)$ is the normal distribution with the covariance matrix C_i and the mean vector \mathbf{y}_i , \mathbf{w} is the combined vector of M parameters of the Gaussian mixture.

In this case, the pattern recognition task is reduced to estimation of mixture parameters \mathbf{w} on the base of the training set $\mathbf{B} = \{\mathbf{x}_i\}_{i=1}^n$. As it was noted, the MDL principle helps to solve the problem of selection of the number d of mixture components, but it doesn't show the possibility to extend this representation. The RMDL criterion will have the following form for this representation

$$L(\mathbf{B}, S, \mathbf{w}) = l(S) + l(\mathbf{w}) - \sum_{i=1}^n \log_2 p(\mathbf{x}_i | \mathbf{w}) \quad , \quad l(\mathbf{w}) = \frac{M}{2} \log_2 n \quad (16)$$

The complexity $l(S)$ depends on the family of representations under consideration. on Selection between different representations S (different forms of probability density function p) can be carried out on the base of this criterion.

Let's consider simpler representations as alternatives, which can appear to be more efficient. Gaussian mixtures will be specified by diagonal covariance matrix $\mathbf{C}_{i,i} = \sigma_i^2$ in one representation, and the unity matrix multiplied by dispersion $\mathbf{C} = \sigma^2 \mathbf{I}$ in another representation. Seemingly, these two representations define models, which are the subset of models defined by representation on the base of full Gaussian mixtures, thus, their inclusion should be useless. However, identical distributions within different representations correspond to different models in inductive inference, because their complexity (in particular, number of parameters) is different.

Consider the example in Figure 2.

Here, 5-dimensional space of features is used (subset for two features is shown in the figure), and patterns are distributed in three clusters (these clusters are separable in the given 5-dimensional space). Results of clustering with the use of three different types of representations and different numbers of components in mixtures are shown.

Corresponding description lengths in bits are given in Table 2. As it can be seen from the table, minimum description length within different representations is achieved for different number of components in mixtures. The reason is that Gaussian distribution is defined by 20 parameters in 5-dimensional space, and there is not enough information in the used training set even to estimate each parameter of mixture with three components. At the same time, the simplest representation appears also to be less efficient despite models have smaller number of parameters within it.

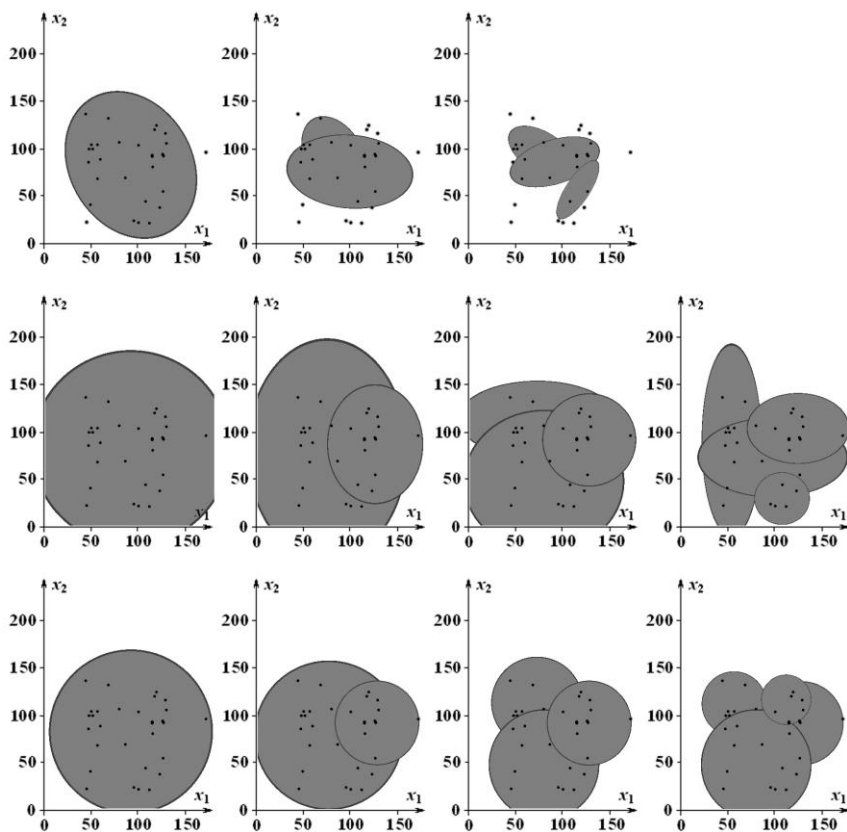


Figure 2. Mixtures with different restrictions on the covariance matrix and different numbers of components.

Table 2. Description lengths (in bits) of the training set ($n=30$) within different mixture models

Type	$M=1$	$M=2$	$M=3$	$M=4$
C	1026	1042	1069	—
$C_{i,i} = \sigma_i^2$	1049	1037	1004	1028
$\mathbf{C} = \sigma^2 \mathbf{I}$	1048	1046	1013	1009

As it can be seen, in order to select correct number of clusters one needs to take different representations into account and to make choice between them. Even small difference between representations can be important. Of course,

automatic selection (on the base of the RMDL criterion) among more diverse representations will significantly increase capabilities of pattern recognition and clustering methods.

Although the RMDL principle allows for using different classifiers simultaneously, it doesn't account for compositions or mixtures of classifiers, which have become popular recently (e.g. [21]). Is it possible to explain within theoretic-information approach, why complex compositions of classification algorithms don't have propensity for overlearning?

6. ALGORITHMIC PROBABILITY

Algorithmic complexity gives the theoretical basis for solving the problem of learning and pattern recognition. Its direct application in practice is impossible, but MDL-based approaches exist, which allow for its useful practical approximations. However, it can be noted that not only does the practical MDL principle approximate Kolmogorov complexity, but also Kolmogorov complexity approximates algorithmic probability [10].

R. Solomonoff proposed to derive probability from algorithmic complexity and to use it in prediction. Indeed, if optimal codes can be derived from probabilities, this task can be inverted in order to find probabilities from optimal codes.

The prior probability $P(\alpha)$ of a program α is connected with its length $l(\alpha)$ as following:

$$P(\alpha) = 2^{-l(\alpha)} \quad (17)$$

Arbitrary string β can be generated by a number of programs α_i , so its Algorithmic Probability (ALP) can be calculated using the equation

$$P_{ALP}(\beta) = \sum_{\alpha: U(\alpha)=\beta} 2^{-l(\alpha)} \quad (18)$$

It can be seen that Kolmogorov complexity approximates ALP since it uses only one term of (18):

$$P_K(\beta) = \max_{\alpha: U(\alpha)=\beta} 2^{-l(\alpha)} \leq \sum_{\alpha: U(\alpha)=\beta} 2^{-l(\alpha)} = P_{ALP}(\beta)$$

Kolmogorov complexity uses only one best model while ALP relies on all appropriate models making convergence of learning and prediction faster. Because it is difficult to develop good practical approximations of Kolmogorov complexity, one can ask, is it really necessary to consider even more computationally inefficient ALP? However, there are reasons to do this.

One of them consists in the fact of relative success of compositions of classifiers. Indeed, such compositions are very complex, and it is quite surprising that they are not subjected to strong overlearning as it should follow from the MDL principle. Even usage of different classifiers with selection of the best one on the base of the RMDL principle doesn't correspond to composition of classifiers. However, ALP accounts for different models simultaneously, so this difference between ALP and MDL can be possibly "responsible" for efficiency of compositions of classifiers.

Is it possible to use some approximations of ALP in practice in the same way as the practical MDL principle is used? Let's consider the most direct application of ALP to the task of prediction.

Here, the prefix algorithmic probability is used:

$$P_{ALP}(\beta) = \sum_{\alpha: U(\alpha)=\beta\dots} 2^{-l(\alpha)}$$

meaning that each model α produces the given string β as the prefix of some string with arbitrary continuation.

Apparently, probability of appearance of some string β' after the string β is of interest. This conditional probability is defined as the probability of the string $\beta\beta'$ divided by the probability of the string β :

$$P_{ALP}(\beta' | \beta) = P_{ALP}(\beta\beta') / P_{ALP}(\beta) . \quad (19)$$

The prefix probability is more appropriate here, because the algorithms that print $\beta\beta'$ are also taken into account, when the probability $P_{ALP}(\beta)$ is calculated: $(\forall \alpha) U(\alpha) = \beta\beta' \dots \Rightarrow U(\alpha) = \beta \dots$. Thus, the probability $P_{ALP}(\beta)$ cannot be smaller than the probability $P_{ALP}(\beta\beta')$.

It should be pointed out that conditional complexity or probability can be calculated relative to the given model, its partial output (e.g. β) or even its input. All these quantities have different meaning, but all of them can be referred to as conditional probability/complexity. It is usually clear from the context, what type of prior information is given. In particular, the conditional

algorithmic complexity in (7) is calculated relative to the given model, while the conditional algorithmic probability in (19) is calculated relative to the given prefix string.

7. TOWARDS PRACTICAL ALGORITHMIC PROBABILITY

Consider the task of pattern recognition, in which the training set $\{d_i, h_i\}_{i=1}^n$ is given. The task is to recognize new pattern d_{n+1} , i.e. to find the most appropriate class label h_{n+1} for it. If no additional prior information is given, the most ideal way to solve this task is to use the conditional algorithmic probability

$$P_{ALP}(h_{n+1} | d_1, h_1, \dots, d_n, h_n, d_{n+1}) = \frac{P_{ALP}(d_1, h_1, \dots, d_n, h_n, d_{n+1}, h_{n+1})}{P_{ALP}(d_1, h_1, \dots, d_n, h_n, d_{n+1})}.$$

The denominator is independent of h_{n+1} , so it can be ignored unless the absolute values of probabilities are desired.

The most common assumption in pattern recognition consists in irrelevance of the order of patterns in the training set. This assumption can be violated in practice, but it is usually accepted. In the other case, one needs to consider much more general task of sequential prediction. This assumption doesn't imply total independence of patterns, and it is impossible to use the equality:

$$P_{ALP}(d_1, h_1, \dots, d_{n+1}, h_{n+1}) = \prod_{i=1}^{n+1} P_{ALP}(d_i, h_i).$$

Instead, it implies that we should consider such programs that can produce patterns by their descriptions presented in arbitrary order, i.e. $U(S\alpha_1 \dots \alpha_{n+1}) = (d_1, h_1) \dots (d_{n+1}, h_{n+1})$, and transposition of α_i leads to corresponding transposition of (d_i, h_i) . Consequently,

$$P_{ALP}(d_1, h_1, \dots, d_{n+1}, h_{n+1}) = \sum_{S, \alpha_1 \dots \alpha_{n+1}; U(S\alpha_i) = (d_i, h_i)} 2^{-l(S\alpha_1 \dots \alpha_{n+1})} = \sum_{S, \alpha_1 \dots \alpha_{n+1}; U(S\alpha_i) = (d_i, h_i)} \left[2^{-l(S)} \prod 2^{-l(\alpha_i)} \right] \quad (20)$$

This result can be interpreted as conditional independence of patterns relative to the given S .

The RMDL principle indicates to choose the shortest solution for $S\alpha_1 \dots \alpha_{n+1}$ yielding (if one ignores additional d_{n+1} in the denominator)

$$P_{ALP}(h_{n+1} | d_1, h_1, \dots, d_n, h_n, d_{n+1}) \approx \frac{2^{-l(S^*)} \prod_{i=1}^{n+1} 2^{-l(\alpha_i^*)}}{2^{-l(S^*)} \prod_{i=1}^n 2^{-l(\alpha_i^*)}} = 2^{-l(\alpha_{n+1}^*)} \quad (21)$$

The task of learning the best S^* is almost separated from the task of recognizing d_{n+1} . However, the best S^* should compress all (d_i, h_i) including (d_{n+1}, h_{n+1}) . Given some S and pattern d to be recognized one should find such shortest α that produces this d and some h : $U(S\alpha) = (d, h)$, and this h will be the most probable class label. The program S can be considered as the representation assigning probabilities to patterns in classes.

In this context, each S specifies a classifier. It can be seen from the equation (20) that ALP (in contrast to MDL) can be used to introduce compositions of classifiers. However, this equation is still too general to be used in practice. In particular, it is difficult to consider generative models that produce both d_i and h_i . One of two simplifications is usually applied. The first approach consists in considering patterns d_i as prior information and generating only h_i . It is utilized in the method of discrimination functions. This might seem natural, because we are indeed interested in predicting h using d . However, this approach focuses only on surfaces separating classes and ignores distributions of patterns within classes.

Another approach consists in describing pattern distributions within each class independently. This approach also seems rather natural within statistical framework. Since Bayes' rule yields $P(H|D) \propto P(D|H)P(H)$, one would like to estimate the density distribution $P(D|H)$ for each class. The fact that this is also simplification becomes obvious in the context of algorithmic information/probability. For example, constructing such two algorithms S_1 and S_2 that $U(S_1\alpha_i^{(1)}) = d_i^{(1)}$ and $U(S_2\alpha_i^{(2)}) = d_i^{(2)}$, where $d_i^{(1)}$ and $d_i^{(2)}$ are patterns belonging to the first and second classes correspondingly, is not the same as constructing one common algorithm S : $U(S\alpha_i) = (d_i, h_i)$. The algorithm S can be simpler than the "sum" of the algorithms S_1 and S_2 , because it can account for mutual regularities in distributions of $\{d_i^{(1)}\}$ and $\{d_i^{(2)}\}$, and

these regularities will require less amount of data to be learned. That's why separate estimation of distributions of patterns for each class can result in non-optimal recognition results.

The class labels h_i are encoded using the patterns d_i in the first approach, and the patterns d_i are encoded using the class labels h_i in the second approach. However, the main problem is that even with these simplifications the algorithms S cannot be directly enumerated. Thus, here we will simply consider some examples of possible applications of ALP to the specific pattern recognition methods in the way similar to the mentioned practical applications of the MDL principle.

Consider the following coding scheme that can account for coding both patterns and class labels. Let the pattern d is encoded on the base of its probability determined by the mixture $P(h_1)P_1(d)+P(h_2)P_2(d)$. Then its class label h is reconstructed by comparison of $P(h_1)P_1(d)$ and $P(h_2)P_2(d)$, and the flag indicating its correctness is encoded on the base of corresponding probability. The total description length will be

$$L_{tot} = L(\{h_i, d_i\} | S) = L_{dist} + L_{err},$$

$$L_{dist} = - \sum_{i=1}^{n_1+n_2} \log_2 (P(h_1)P_1(d_i) + P(h_2)P_2(d_i)); \quad L_{err} = \log_2 C_{n+1}^{n_{err}}, \quad (22)$$

where n_1 and n_2 are the number of pattern in each class including the pattern to be recognized ($n_1+n_2=n+1$), which class label is a part of the model to be evaluated; L_{dist} is the description length of the patterns d_i distributed in the feature space, and L_{err} is the description length of (incorrectly reconstructed) class labels h_i . The value of $L_{tot}=l(\alpha_1 \dots \alpha_{n+1})$ is estimated within certain general model S and doesn't include $l(S)$, which depends on the considered family of classifiers.

Let's compare this coding scheme with the scheme, in which class labels are encoded at first and then patterns are encoded on the base of probabilities $P_{1,2}(d)$ depending on the class label of each pattern. The total description length of the training set will be

$$L'_{tot} = -n_1 \log_2 P(h_1) - n_2 \log_2 P(h_2) - \sum_{i=1}^{n_1} \log_2 P_1(d_i^{(1)}) - \sum_{i=1}^{n_2} \log_2 P_2(d_i^{(2)}) =$$

$$= - \sum_{i=1}^{n_1} P_1(d_i^{(1)}) \log_2 P(h_1) - \sum_{i=1}^{n_2} P_2(d_i^{(2)}) \log_2 P(h_2).$$

Obviously,

$$-\log_2(P(h_1)P_1(d_i) + P(h_2)P_2(h_i)) \leq -\log_2(P(h_k)P_k(d_i)), k = 1, 2$$

Therefore,

$$-\sum_{i=1}^{n_1+n_2} \log_2(P(h_1)P_1(d_i) + P(h_2)P_2(d_i)) \leq -\sum_{i=1}^{n_1} P_1(h_i^{(1)}) \log_2 P(h_1) - \sum_{i=1}^{n_2} P_2(d_i^{(2)}) \log_2 P(h_2)$$

This means that in absence of classification errors the first coding scheme will be more efficient. Only in the case, when it is impossible to predict h_i well, they will be comparable.

Apparently, the precision of the general model (representation) S is determined as a compromise between its precision in describing distribution of patterns and precision of assigning class labels to them. One can decrease the first component of precision in order to increase the second component and visa versa. If we start from the model that describes the pattern distribution most precisely, then we should try to increase its recognition precision by the cost of the distribution precision. It can be done by increasing weights of incorrectly recognized patterns and recalculating pattern distribution that will be less precise with respect to the whole training set. The total description length will probably decrease at first, but achieving zero value of L_{err} may lead to greater increase of L_{dist} .

It is difficult to achieve good recognition results starting with a mixture distribution that was constructed without taking class labels of the training patterns into account (i.e. within the unsupervised learning framework). Actually, initial unsupervised minimization of L_{dist} can be very useful as it is shown in the deep learning approach, but it requires utilization of another family of representations, namely distributed representations, which are not considered in this chapter. More practical approach in our case is to start with a mixture $P(h_1)P_1(d) + P(h_2)P_2(d)$, in which the distributions $P_1(d)$ and $P_2(d)$ are constructed using patterns, which belong to the corresponding classes only. It doesn't guarantee that L_{err} is minimal, so increase of weights of incorrectly classified patterns can still decrease the total description length. On the other hand, decrease of weights of some patterns can also lead to decrease of the description length. The former case corresponds to boosting, while the latter case corresponds to rejection of outliers. Conventional boosting is sensitive to outliers. However, the description length criterion can account for both

boosting and outliers rejection in the unified fashion that makes this approach promising. Additionally, classifiers S constructed with different weights of training patterns can be combined in the form of mixture (20).

Consider the simplest example of Gaussian distributions estimation for the two classes shown in Figure 3. Training sample contained 100 patterns.

The description length criterion was the same as equation (16), but with additional L_{err} component. At first, Gaussian distributions were estimated in such the way that the distribution precision (minimize L_{dist}) was maximized. Then, weights of incorrectly classified patterns were increased, and new distributions were estimated with smaller distribution precision, but higher classification precision. Changes of the total description length with increase of L_{dist} (caused by increase of weights of incorrectly classified patterns) are shown in Figure 4.

It can be seen that L_{tot} initially decreases with increase of weights of incorrectly classified patterns, because L_{dist} grows slower than L_{err} decreases. However, L_{tot} begins to grow after some point.

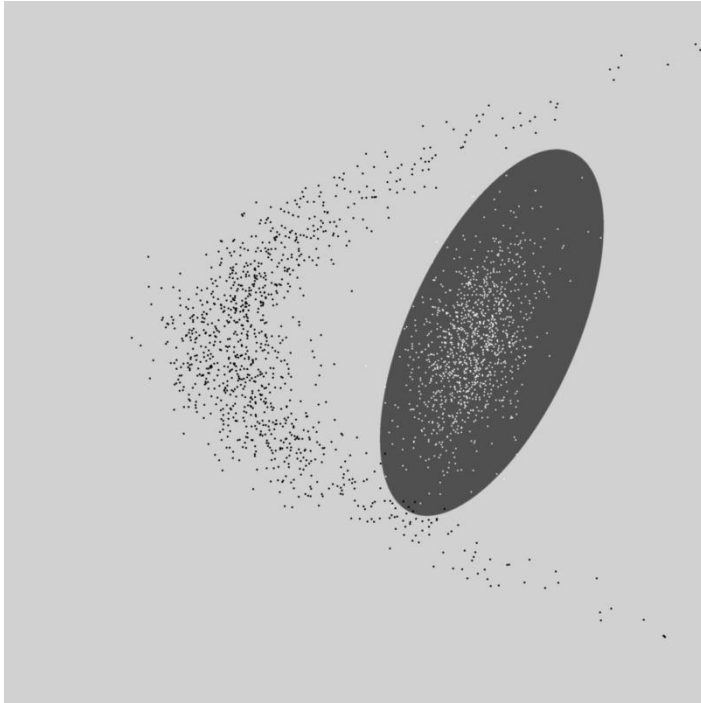


Figure 3. Example of the test sample.

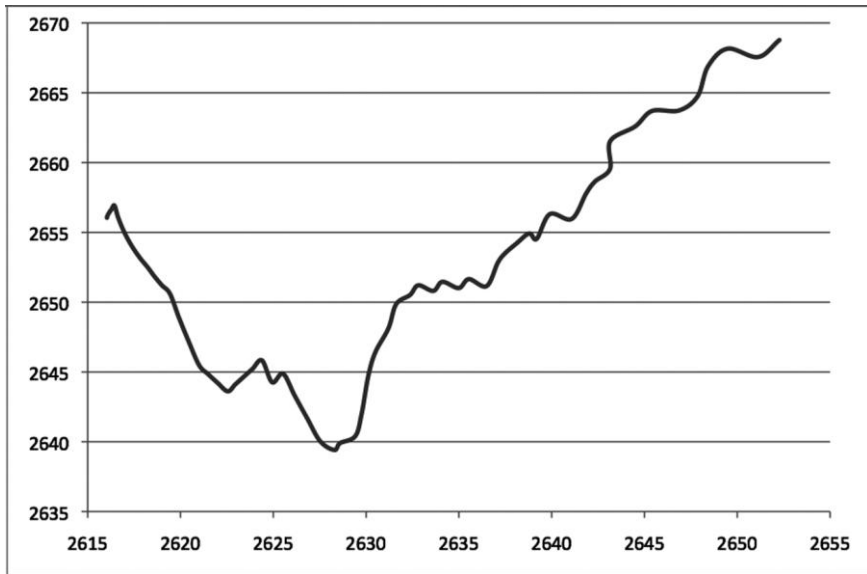

 Figure 4. Dependence of L_{tot} from L_{dist} .

Table 3. Recognition rate of different classifiers

	Smallest L_{dist}	Smallest L_{err}	Smallest L_{tot}	ALP-mixture
Errors on the test sample	3.0%	2.1%	1.45%	1.4%

Using the MDL principle one should expect that the model with the smallest L_{tot} will most probably yield the best recognition results (here, classifiers with the same complexity S are considered). However, within the ALP framework one shouldn't choose only one best model, but can take all these models with weights $2^{-L_{tot}}$.

It should also be pointed out that L_{tot} is calculated using all training patterns and the pattern to be recognized.

In contrast, the distributions $P_1(d)$ and $P_2(d)$ are traditionally estimated only on the base of the training set, and then they are applied to the new pattern without being updated. The latter approach requires much less computational resources, but it is less precise.

Resulting recognition errors of the classifiers with $P_1(d)$ and $P_2(d)$ corresponding to the distributions with the minimal L_{dist} , L_{err} and L_{tot} , and the "ALP-mixture" of classifiers are shown in Table 3.

Here, ALP-mixtures give minor average improvement of the recognition rate in addition to the selection of the model with the best L_{tot} . And even ALP-mixture doesn't allow for the best possible recognition, because very restricted representations (families of classifiers) are used. For example, usage of the two-component Gaussian mixtures for describing distributions for each class results in 0.5% recognition error rate on the same test sample.

It should be pointed out that ALP-mixtures of Gaussians are not the same as conventional Gaussian mixtures. While using Gaussian mixtures in the form (15), one tries to fit all its components to the given training patterns simultaneously. That's why Gaussian mixtures with large number of components will tend to overlearning. At the same time, in the case of ALP-mixtures its components are fit to the given data individually – they don't constitute one common model, but stand for different alternative models. In particular, this implies that ALP-mixture of mixtures of Gaussians is possible, and it will differ from both ALP-mixtures of Gaussians and conventional mixtures of Gaussians.

One can expect that usage of ALP-mixtures will have stronger effect, when there are models, which correspond to different types of regularities and have similar description lengths. Let's consider the same classes as presented in Figure 3, but the convex (right) class will be shifted to the left (classes will be closer, and their overlapping will be higher). Let the distribution of pattern in each class be described using Gaussian mixtures with different number of components. A pair of mixtures specifies a classifier. The best classifier for fixed numbers of components in two mixtures can be constructed using the technique described above:

- 1 The mixture of Gaussians is estimated for each class (e.g. using expectation-maximization method).
- 2 Weights of incorrectly classified patterns are increased.
- 3 Parameters of the mixture are re-evaluated.
- 4 Steps 2–3 are executed until the value of L_{tot} starts to constantly increase.
- 5 The mixture with the lowest L_{tot} is chosen.

Additionally, weights of some patterns can be decreased if this results in decrease of the total description length in order to exclude outliers.

One needs also to account for the value of $l(S)$. Since the basic algorithms for different classifiers are the same here, difference in their complexities will depend only on the number M of their parameters $l(w)=0.5M\log_2(n+1)$; see the

equation (16). Comparison results for pairs of mixtures with different numbers of components are shown in Table 4. The error rate and the value of $L_{tot}+l(w)$ are given.

It is obvious from comparison of the first column of the table with other columns that the description of the concave class with single Gaussian is very inefficient. Mixtures of three or four Gaussians are much more appropriate. On the contrary, the convex class is better described using only one Gaussian (the first row of the table).

This is quite natural, but an important point is that the description length corresponds to the error rate on test sample very well. The only prominent discrepancy can be observed in the case, when the concave class is described using single Gaussian, and the convex class is described using mixtures of several Gaussians (the first column in the table).

Several solutions with different number of components in mixtures have close values of description lengths. The ALP-mixture of such solutions specifies a discrimination surface, which differs from surfaces specified by individual solutions.

In this case, the error rate of the ALP-mixture equals to 6.3%. Improvement of the recognition rate is not very impressive. Additionally, it is unstable – the error rate of the ALP-mixture can appear to be the same or even slightly worse than that of the solution with the minimum description length depending on the random seed.

Of course, more advanced families of classifiers and real-world examples should be considered in order to confidently conclude if ALP-mixtures can yield significant increase of recognition rate.

Table 4. Error rates and description lengths (in bits) for pairs of mixtures with different number of components

# Components	1	2	3	4
1	9.0% 2627	6.8% 2517	6.7% 2509	6.7% 2510
2	8.7% 2635	6.9% 2518	7.2% 2518	6.8% 2515
3	8.5% 2643	6.9% 2528	7.4% 2521	6.9% 2517
4	10.6% 2630	7.4% 2537	7.2% 2533	8.4% 2536

CONCLUSION

The approach to pattern recognition based on algorithmic complexity and probability was considered in this chapter. Such practical approximations of algorithmic complexity as the Minimum Description Length and Minimum Message Length principles are well known for their capability to solve the overlearning problem. However, not only do the MML/MDL principles approximate algorithmic complexity, but also algorithmic complexity approximates algorithmic probability. Solomonoff's universal prediction method based on ALP calculated using all algorithmic models, which produce the given data with different continuations, can be applied to state the task of pattern recognition correctly. The most general solution of the task of pattern recognition is reduced to estimation of the conditional algorithmic probabilities of different class labels relative to the given training set and the pattern to be recognized. This conditional algorithmic probability is proportional to the algorithmic probability of the data string composed from the training patterns and their class labels extended with the pattern to be recognized and its hypothesized class label. Assuming irrelevance of order of patterns in this data string, the task of estimating its algorithmic probability can be greatly simplified. However, it still requires search in the Turing-complete model space. Thus, for the sake of practical applicability some restricted families of classifiers and corresponding heuristic coding schemes are to be used as it is usually done in applications of the MDL principle.

Even careful application of the MDL principle to the task of pattern recognition stated in the ALP framework helps to improve recognition rates of certain families of classifiers. In particular, it allows one to account for both boosting and rejection of outliers in the unified fashion. This effect is shown on the example of normal mixtures. Additionally, ALP explains how compositions of classifiers, which seem to be complex, can avoid overlearning. Such compositions referred in this chapter to as ALP-mixtures should consist of independently constructed classifiers taken with weights proportional to 2^{-L_i} , where L_i is the description length of the data string encoded using i -th classifier.

However, practical application of ALP-mixtures results in rather modest increase of recognition rate by the cost of increase of computational efforts in comparison with the use of the single model with the best description length. Indeed, discrimination surfaces produced by ALP-mixtures considerably differ from surfaces produced by individual models only when several alternative models have similar values of the description length. Such situations usually

appear, if the given data contain regularities absent in the model space (or the complexity of the underlying regularity exceeds the amount of available information); but ALP-mixtures don't recover missing regularities and don't perform necessary generalizations. Nevertheless, more complex examples should be investigated and richer model spaces should be considered in order to confirm capabilities and limitations of practical approximations of ALP.

REFERENCES

- [1] Li, M., Vitanyi, P. (1992). Philosophical issues in Kolmogorov complexity. *Proc. ICALP'92, invited lecture*, pp. 1–15.
- [2] Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *2nd Int. Symposium on Information Theory*, pp. 267–281.
- [3] Schwarz, G. (1978). Estimating dimension of a model. *Ann. Stat.*, vol. 6, pp. 461–464.
- [4] Rissanen, J. J. (1978). Modeling by the shortest data description. *Automatica-J.IFAC*, vol. 14, pp. 465–471.
- [5] Wallace, C. S., Boulton, D. M. (1968). An information measure for classification. *Comput. J.*, vol. 11, pp. 185–195.
- [6] Lee, T. C. M. (2002). Tree-based wavelet regression for correlated data using the minimum description length principle. *Australian and New Zealand Journal of Statistics*, vol. 44, no. 1, pp. 23–39.
- [7] Thomas, I. et al. (1998). A minimum message length evaluation metric for lexical access in speech understanding. In: H. Y. Lee and H. Motoda (eds.), *Proc. 5th Pacific Rim Int. Conf. on Artificial Intelligence (PRICAI'98)*, pp. 49–54.
- [8] Tabus, I., Astola, J. (2001). On the use of MDL principle in gene expression prediction. *J. Appl. Signal Proc.*, no. 4, pp. 297–303.
- [9] Solomonoff, R. (1964). A formal theory of inductive inference, part 1 and part 2. *Information and Control*, vol. 7, pp. 1–22, 224–254.
- [10] Solomonoff, R. (1997). Does algorithmic probability solve the problem of induction? Oxbridge Research, P.O.B. 391887, Cambridge, Mass. 02139.
- [11] Potapov, A. S. (2008). Comparative analysis of structural representations of images based on the principle of representational minimum description length. *Journal of Optical Technology*, vol. 75, no. 11, pp. 715–720.

-
- [12] Lee, T. (2000). A minimum description length based image segmentation procedure, and its comparison with a cross-validation based segmentation procedure. *J. of American Statistical Assoc.*, vol. 95, pp. 259–270.
 - [13] Vitanyi, P. M. B., Li, M. (2000). Minimum description length induction, Bayesianism, and Kolmogorov complexity. *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 446–464.
 - [14] Sato, M., Kudo, M., Toyama, J., Shimbo, M. (1997). Construction of a nonlinear discrimination function based on the MDL criterion. *1st Int. Workshop on Stat. Techniques in Pattern Recognition*, pp. 141–146.
 - [15] Von Luxburg, U., Bousquet, O., Schölkopf, B. (2004). A compression approach to support vector model selection. *Machine Learning Research* 5, pp. 293–323.
 - [16] Tenmoto, H., Kudo, M., Shimbo, M. (1998) MDL-based selection of the number of components in mixture models for pattern classification. *Advances in Pattern Recognition*, num. 1451 in *Lecture Notes in Computer Science*: Springer, pp. 831–836.
 - [17] Kolmogorov, A. (1968). Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, vol. 14, no. 5, pp. 662–664.
 - [18] Potapov, A. S. (2008). Investigation of image representation on the base of representational minimum description length principle. *Trans. of institutes of higher education, instrument-making*, vol. 51, no. 7, pp. 3–7 (in Russian).
 - [19] Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, CA.
 - [20] Potapov, A. S. (2008). Synthetic pattern recognition methods based on the representational minimum description length principle. *Proc. OSAV'2008, the 2nd Int. Topical Meeting on Optical Sensing and Artificial Vision*, pp. 354–362.
 - [21] Bauer, E., Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, vol. 36, pp. 105–139.

Chapter 5

PATTERN RECOGNITION USING QUATERNION COLOR MOMENTS

E. G. Karakasis^{1,}, G. A. Papakostas^{2,†} and D. E. Koulouriotis^{1,‡}*

¹Department of Production Engineering and Management,
Democritus University of Thrace, Greece

²Department of Industrial Informatics, TEI of Kavala,
Human-Machines Interaction (HMI) Laboratory, Greece

Abstract

Image moments have been established in the area of pattern recognition and classification, since they can represent image content very effectively. One of the first moment family, and probably one of the most used, is the geometric one. However, a large variety of moments have been introduced so far. Orthogonal continuous moments like Zernike, Fourier-Mellin and pseudo Zernike or orthogonal discrete moments like Tchebichef, Krawtchouk and dual Hahn are only some of the most widespread families. Despite this variety in moment families, their vast majority has been applied in pattern recognition or classification problems where only gray images are considered. Only lately scientists try to address the issue of calculating moments for color images. The traditional way to apply moments to such images is either to use a color reduction method, with the known consequences of losing information, or to convert the color

*E-mail address: ekarakas@pme.duth.gr

†E-mail address: gapak@teikav.edu.gr, gapakos@ee.duth.gr

‡E-mail address: jimk@pme.duth.gr

model from RGB to HSV in order to use only the H channel. Another perspective is to represent each color pixel as a 3-element vector. The resulted image can be used in order to compute color moments. This method results in to calculate 3-element vectorized moments, where each element is in relation with the corresponding channel but not with the other elements. Quaternion moments, which have been lately attracting the interest of scientific community, address this problem elegantly. By representing each color pixel as a quaternion and using simple quaternion algebra, the resulted quaternion moments are directly connected to the image color space. In this chapter the basic theory as well as a comparison of the aforementioned two types of color moments is presented. The experimental analysis includes, except of image reconstruction and computation time cases, indicative classification examples in noise free and noisy conditions.

Keywords: Quaternion color moments, image reconstruction, moment invariants, image moments

1. Introduction

Moment functions, as great features and image content descriptors, have been widely used in the area of image analysis [1, 2, 3]. Their main characteristic is that an image, defined in a Cartesian coordinate system, can be projected into a polynomial space. Moments get their name as well as inherit their properties according to the used kernel (basis) function. As kernels can be used orthogonal or not, real or complex-valued functions, defined in a continuous or discrete domain. An important characteristic of moments is that with proper handling they can become invariant to spatial transformations (translation, rotation, scaling, affine, etc.) [4, 5, 6]. Another characteristic of image moments is their ability, according to the used kernel, to reconstruct the original image. Moments with discrete orthogonal kernels present the best reconstruction results.

The first and most simple family is the geometric moments (GMs). Their simplicity makes the production of the corresponding moment invariants easy. However, GMs suffer from information redundancy. In 1980, Teague [7] introduced the Zernike moments (ZMs). ZMs make use of continuous orthogonal polynomial as kernel, which is defined in a polar coordinate system. Due to orthogonality property of the polynomial, the new moments are characterized by minimum information redundancy and great discriminative power. Other similar continuous orthogonal moment families are the pseudo Zernike (PZMs) and

orthogonal Fourier-Mellin (OFMMs) [8]. However, these families have - due to their continuous nature - two main drawbacks: 1) the image coordinates have to be transformed from Cartesian to polar coordinate system and 2) they are characterized by approximation errors (integrals are substituted by summations).

The approximation errors may be eliminated by using exact forms of continuous moment families [9], nevertheless, there is another perspective that was introduced by Mukundan [10]: the usage of discrete orthogonal polynomials. Moments that use such polynomials as kernel are free of approximations errors. Mukundan was initially used the Tchebichef polynomials to define discrete Cartesian moments (TMs). Other similar families are the Krawtchouk (KMs) [11] and dual Hahn moments (HMs) [12]. Discrete moments present better discriminative power compared to the continuous ones. However, discrete moments, initially defined in Cartesian coordinate system, cannot form rotation invariants as easy as in the case of the aforementioned continuous ones, which are natively defined in polar coordinates. Therefore, Mukundan introduced in [13] the radial Tchebichef moments which can easily produce rotation moment invariants.

The moment functions have been initially applied to gray images, nevertheless, color images can also be addressed by such functions, though in a slightly different perspective. In this chapter, two different methods for moments computation are going to be presented and compared. In the first one, let us call it “ordinary”, each color channel is treated as a separate gray image. The second one uses quaternions in order to represent each color pixel.

Quaternion is a four-dimensional number introduced by Hamilton in 1843 [14, 15]. It consists of three imaginary parts (i , j and k), thus it is characterized as hyper-complex. If $a, b, c, d \in \mathbb{R}$ then a quaternion q can be defined as $q = a + bi + cj + dk$. Quaternions and complex numbers share a lot of properties, nevertheless, their main difference is that the multiplication of quaternions is not commutative ($q_1 q_2 \neq q_2 q_1$).

Quaternion theory has been successfully applied in image analysis. Sangwine [16] is the one who has initially applied the use of quaternions in color images, while Ell [17] had earlier introduced the quaternion Fourier Transform. In the field of image moments, Chen et al. [18] extended the definition of conventional Zernike moments using the theory of quaternions in order to deal with color images.

2. Quaternion Basics

This section aims to serve as a simple introduction to basic quaternion theory. Quaternions were initially introduced by Hamilton in 1843 [14, 15] as a generalization of the complex numbers and, as has already been mentioned in section 1, the main difference between quaternions and complex numbers is that the quaternion multiplication is not commutative ($q_1 q_2 \neq q_2 q_1$). In contrast with the complex numbers, quaternions have three imaginary parts i , j and k . Multiplying them two at a time, the following rules can be deduced

$$i^2 = j^2 = k^2 = -1, \quad ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j \quad (1)$$

The quaternion q can be defined as follows

$$q = q_r + q_i i + q_j j + q_k k \quad (2)$$

$$q_r, q_i, q_j, q_k \in \mathbb{R}$$

or similarly

$$q = Re(q) + Im(q) \quad (3)$$

where

$$Re(q) = q_r \quad \text{and} \quad Im(q) = q_i i + q_j j + q_k k$$

The quaternion conjugate is given by

$$\bar{q} = q_r - q_i i - q_j j - q_k k \quad (4)$$

and the modulus (or norm) is defined as

$$|q| = \sqrt{q_r^2 + q_i^2 + q_j^2 + q_k^2} \quad (5)$$

The addition and the multiplication of quaternions can be defined as

$$q + p = (q_r + p_r) + (q_i + p_i)i + (q_j + p_j)j + (q_k + p_k)k \quad (6)$$

and

$$\begin{aligned} qp &= (q_r p_r - q_i p_i - q_j p_j - q_k p_k) \\ &+ (q_i p_r + q_r p_i - q_k p_j + q_j p_k)i \\ &+ (q_j p_r + q_k p_i + q_r p_j - q_i p_k)j \\ &+ (q_k p_r - q_j p_i + q_i p_j + q_r p_k)k \end{aligned} \quad (7)$$

respectively. Finally, the inverse of the quaternion q can be computed by

$$q^{-1} = \frac{\bar{q}}{|q|^2} \quad (8)$$

The quaternions can also be expressed in polar forms as follows

$$q = |q|e^{\mu\phi}, \quad |e^{\mu\phi}| = 1 \quad (9)$$

where

$$e^{\mu\phi} = \cos(\phi) + \mu \cdot \sin(\phi) \quad (10)$$

is the Euler's formula and

$$\mu = \frac{Im(q)}{|Im(q)|}, \quad |\mu| = 1$$

$$\phi = \arctan\left(\frac{|Im(q)|}{Re(q)}\right)$$

It should be mentioned that the quaternion μ is called unit pure quaternion. The words *unit* and *pure* correspond to the relations $|\mu| = 1$ and $Re(\mu) = 0$, respectively.

3. Moment Categories

In this section a general categorization of moment families is going to be presented. The categories are based mainly on the characteristics of the used kernel function. In the introduction is referred that as kernel functions can be used orthogonal or not, real or complex-valued functions, defined in a continuous or discrete domain. Therefore, the moment families are categorized considering the property of continuity, orthogonality and the used coordinate system.

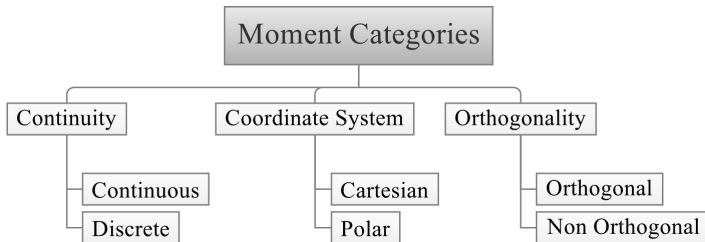


Figure 1. Moment families categorization.

In Fig. (1) it can be seen that based on the continuity the families can be characterized as continuous or discrete, while on the basis of the used coordinate system the families can be classified as Cartesian or polar. Finally, considering the property of orthogonality, the moment families can be categorized as orthogonal or non orthogonal. In Fig. (2) some of the most widespread moment families are presented based on the particular categorization.

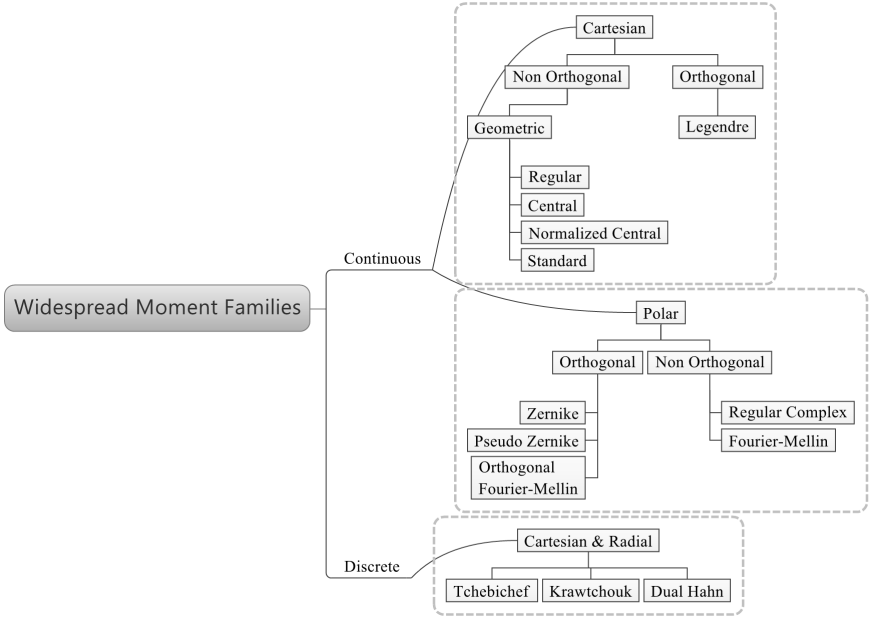


Figure 2. Widespread Moment Families.

As it can be observed from Fig. (2), the Tchebichef, Krawtchouk and Dual Hahn discrete moment families can be expressed in both Cartesian and radial form. Radial and polar moments are defined in polar coordinate system. However, instead of polar moments which are natively defined in polar coordinates, radial moments are Cartesian ones which have been transformed into polar. Radial Tchebichef are the first introduced discrete radial moments [13].

4. Producing Radial Color Moments

As has already been mentioned, Mukundan proposed the radial Tchebichef moments [13] in order to combine the advantages of both polar (easy way of producing rotation invariants) and discrete Cartesian moments (accuracy, no approximation errors). For the same reason the radial Tchebichef moments are also selected for the purposes of this chapter.

4.1. Tchebichef Polynomials

In his work [10], Mukundan proposed the usage of discrete orthogonal polynomials as kernel functions and more specifically, the usage of Tchebichef polynomials. According to Mukundan, such polynomials reduce the computational cost and the required image coordinates transformation of the continuous orthogonal moments. The discrete Tchebichef polynomials, $t_n(x)$, $x \in \{0, \mathbb{Z}^*\}$, (\mathbb{Z}^* is the set of positive integers) satisfy the following orthogonality condition

$$\sum_{x=0}^{N-1} t_n(x) t_m(x) = \rho(n, N) \delta_{nm} \quad (11)$$

where

$$\rho(n, N) = (2n)! \binom{N+n}{2n+1} \quad , \quad n \in [0, N-1] \quad (12)$$

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}$$

The discrete Tchebichef polynomials [10] can be defined as

$$t_n(x) = (1-N)_n \cdot {}_3F_2(-n, -x, 1+n; 1, 1-N; 1) \quad (13)$$

$$n, x \in [0, N-1]$$

where ${}_3F_2(\cdot)$ is the generalized hypergeometric function given by

$${}_3F_2(a_1, a_2, a_3; b_1, b_2; z) = \sum_{k=0}^{\infty} \frac{(a_1)_k \cdot (a_2)_k \cdot (a_3)_k}{(b_1)_k \cdot (b_2)_k} \cdot \frac{z^k}{k!} \quad (14)$$

N is the image dimension and $(a)_k$ is the Pochhammer symbol

$$(a)_k = a \cdot (a+1) \cdot \dots \cdot (a+k-1) \quad (15)$$

The Tchebichef polynomials obey in the following recurrence formula

$$(n+1)t_{n+1}(x) = (2n+1)(2x-N+1)t_n(x) - n(N^2-n^2)t_{n-1}(x) \quad (16)$$

According to [19] the Tchebichef polynomials can also be written as

$$t_n(x) = \sum_{k=0}^n B_{nk} \sum_{i=0}^k S_{ki} x^i \quad (17)$$

Where

$$S_{ki} = S_{(k-1)(i-1)} - (k-1)S_{(k-1)i} \quad (18)$$

$$k, i \geq 1$$

$$S_{k0} = S_{0i} = 0 \text{ for } k, i \geq 1 \text{ and } S_{00} = 1$$

are the Stirling numbers of first kind and the factor B_{nk} is given by

$$B_{nk} = \frac{(n+k)!}{(n-k)!(k!)^2} < n-N >_{n-k} \quad (19)$$

$$< a >_k = a(a-1) \cdots (a-k+1)$$

$$< a >_0 = 1$$

By transforming eq. (17) into the form $t_n(x) = \sum_{i=0}^n a_{ni} x^i$

$$\begin{aligned} t_n(x) &= \sum_{k=0}^n B_{nk} \sum_{i=0}^k S_{ki} x^i \\ &= B_{n0} S_{00} x^0 \\ &+ B_{n1} S_{10} x^0 + B_{n1} S_{11} x^1 \\ &+ B_{n2} S_{20} x^0 + B_{n2} S_{21} x^1 + B_{n2} S_{22} x^2 \\ &+ \dots \\ &+ B_{nn} S_{n0} x^0 + B_{nn} S_{n1} x^1 + \dots + B_{nn} S_{nn} x^n \\ &= \sum_{i=0}^n \left(\sum_{k=i}^n B_{nk} S_{ki} \right) x^i = \sum_{i=0}^n a_{ni} x^i \end{aligned} \quad (20)$$

we find that the Tchebichef coefficients a_{ni} are given by

$$a_{ni} = \sum_{k=i}^n B_{nk} S_{ki} \quad (21)$$

4.2. Standard Radial Moments

The radial moments are defined in polar coordinates, and according to Mukundan, we can define such moments in a similar way like polar ones (e.g. Zernike). Therefore, an one-dimensional polynomial of order n , $P_n(r)$, where

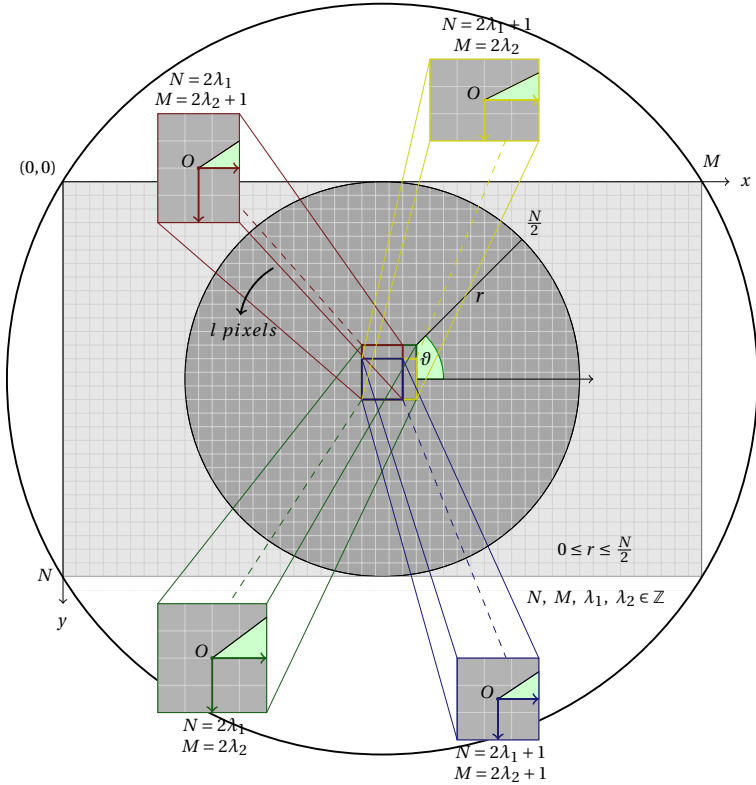


Figure 3. Image in Cartesian and polar coordinates.

$r \in [0, \lfloor \frac{N}{2} \rfloor]$ denotes the radius of the image's internal circular area (see Fig. 3), must combined with the circular function $e^{-im\theta_k}$, where i is imaginary unit, m denotes the repetition, θ_k is an angle given by $\theta_k = 2\pi k/l$, $0^\circ \leq \theta_k \leq 2\pi$ and the factor l denotes the maximum number of pixels along the circumference (see Fig. 3). Any $N \times N$ gray image $f(x, y)$, defined in Cartesian coordinates, can be transformed in polar ones, $f(r, \theta_k)$, by using the variables r and θ_k . The relation between the Cartesian and polar coordinates is given by the following

equations.

$$x = \left\lfloor \frac{rN}{2(\lfloor \frac{N}{2} \rfloor - 1)} \cdot \cos(\theta_k) + \frac{N}{2} \right\rfloor \quad (22)$$

$$y = \left\lfloor \frac{rN}{2(\lfloor \frac{N}{2} \rfloor - 1)} \cdot \sin(\theta_k) + \frac{N}{2} \right\rfloor \quad (23)$$

We can extend the Mukundan's radial Tchebichef moments by using as kernel polynomial any discrete orthogonal polynomial. The extended version of discrete radial moments, R_{nm} , of order n and repetition m , can be defined as follows.

$$R_{nm} = \frac{1}{l \cdot W_n} \cdot \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} P_n(r) e^{-im\theta_k} f(r, \theta_k) \quad (24)$$

where W_n is a scaling factor which depends on the corresponding family.

4.3. Radial Vectorized Color Moments

The vectorized color moments can be easily computed by representing each color pixel of an image as a 3-element vector and by using a slightly different version of eq. (24). If $f_c(r, \theta_k)$ is a color image in polar coordinates and $f_c^r(r, \theta_k)$, $f_c^g(r, \theta_k)$ as well as $f_c^b(r, \theta_k)$ are the red, green and blue color channels, respectively, then a vectorized color image is defined as $f_v(r, \theta_k) = [f_c^r(r, \theta_k), f_c^g(r, \theta_k), f_c^b(r, \theta_k)]$. The radial vectorized color moments can be computed by replacing the gray image $f(r, \theta_k)$ in eq. (24) with the vectorized image $f_v(r, \theta_k)$.

$$V_{nm} = \frac{1}{l \cdot W_n} \cdot \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} P_n(r) e^{-im\theta_k} f_v(r, \theta_k) \quad (25)$$

$$= \frac{1}{l \cdot W_n} \cdot \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} P_n(r) e^{-im\theta_k} [f_c^r(r, \theta_k), f_c^g(r, \theta_k), f_c^b(r, \theta_k)] \quad (26)$$

$$= [V_{nm}^r, V_{nm}^g, V_{nm}^b] \quad (27)$$

Examining eq. (27), it can be deduced that each element of the radial vectorized color moments of order (n, m) is related only by the corresponding color channel. Furthermore, due to the complex nature of these moments, each element

V_{nm}^k , $k = r, g, b$, consists of two parts: a real and a complex one. Therefore, each vectorized color moment V_{nm} consists of six numbers.

4.4. Radial Quaternion Color Moments

The radial quaternion color moments can be defined in the same spirit as in eq. (24). However, initially a color image must be represented in quaternion form. Supposing that $f_c(r, \theta_k)$ is a color image in polar coordinates, then the quaternion image $f_q(r, \theta_k)$ can be defined as follows.

$$f_q(r, \theta_k) = 0 + f_c^r(r, \theta_k)\mathbf{i} + f_c^g(r, \theta_k)\mathbf{j} + f_c^b(r, \theta_k)\mathbf{k} \quad (28)$$

where \mathbf{i} , \mathbf{j} and \mathbf{k} are imaginary parts and f_c^r , f_c^g as well as f_c^b are the red, green and blue color channels, respectively. By replacing the imaginary unit i in eq. (24) with the unit pure quaternion $\mu = \frac{\mathbf{i}+\mathbf{j}+\mathbf{k}}{\sqrt{3}}$ and the gray image $f(r, \theta_k)$ with the quaternion image $f_q(r, \theta_k)$, the radial quaternion color moments can be defined as

$$Q_{nm} = \frac{1}{l \cdot W_n} \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} P_n(r) e^{-\mu m \theta_k} f_q(r, \theta_k) \quad (29)$$

Since the multiplication of quaternion is not commutative, there is another similar form of eq. (29)

$$Q_{nm} = \frac{1}{l \cdot W_n} \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} f_q(r, \theta_k) P_n(r) e^{-\mu m \theta_k} \quad (30)$$

For the sake of simplicity only eq. (30) is going to be presented. Quaternion color moments Q_{nm} are consisting of four numbers: one real and three complex components.

4.5. Image Reconstruction

Supposing that a set of radial (standard, vectorized or quaternion) moments up to order $n_{max} \times m_{max}$ is known, then the following approximate forms can be used in order to reconstruct:

1. the gray image

$$f(r, \theta) \approx \sum_{n=0}^{n_{max}} \sum_{m=0}^{m_{max}} R_{nm} P_n(r) e^{im\theta} \quad (31)$$

2. the vectorized color image

$$f_v(r, \theta) \approx \sum_{n=0}^{n_{max}} \sum_{m=0}^{m_{max}} V_{nm} P_n(r) e^{im\theta} \quad (32)$$

3. the quaternion color image

$$f_q(r, \theta) \approx \sum_{n=0}^{n_{max}} \sum_{m=0}^{m_{max}} Q_{nm} P_n(r) e^{\mu m\theta} \quad (33)$$

where R_{nm} , V_{nm} and Q_{nm} are the radial standard, vectorized and quaternion moments of order (n, m) , respectively.

5. Radial Moment Invariants

At this point it should be clarified that the presented radial (standard, vectorized or quaternion) moments and their corresponding invariants can be used only for moments of discrete kernel polynomial (e.g. Tchebichef, Krawtchouk, dual Hahn etc.). Furthermore, the following formulas can be used for all types of moments (standard, vectorized and quaternion).

5.1. Translation

A characteristic of radial or polar moments is that the origin of a color image is placed in the centroid. The coordinates of centroid can be found by

$$x_c = \frac{G_{10}}{G_{00}} \quad , \quad y_c = \frac{G_{01}}{G_{00}}$$

where G_{nm} are the regular geometric moments of the corresponding gray image. Thus, translation invariance can be considered as native property of radial/polar moments.

5.2. Scaling

The authors have shown in [20], based on Xiao's work [21], that the following form

$$J_{nm}^s = \frac{1}{W_n} \sum_{t=0}^n W_t \left(\sum_{z=0}^n (M_{00})^{-(z+1)} a_{nz} D_{zt} \right) M_{tm} \quad (34)$$

is invariant to scale image transformations. M_{nm} is the standard, vectorized or quaternion moment of order (n, m) , W_t is a weighted factor which depends on the moment family, a_{nz} represents the coefficients of the kernel polynomial and D_{zt} is the (z, t) element of the matrix D . The matrix D is the Moore–Penrose pseudo-inverse ($D = A^+$) of matrix A which is formed as follows

$$A = \begin{bmatrix} a_{00} & 0 & \cdots & 0 \\ a_{10} & a_{11} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

5.3. Rotation

In the same work [20], authors showed that the following equation is invariant to rotation only for radial moments of discrete kernel polynomial.

$$J_{nm}^r = \frac{M_{nm}}{M_{0m}} \quad (35)$$

Where M_{nm} is the standard, vectorized or quaternion moment of order (n, m) .

5.4. Rotation and Scaling

It can easily be deduced that the combination of the above eqs. (34) and (35) can lead to radial moments invariant to rotation and scaling.

$$J_{nm}^{sr} = \frac{\frac{1}{W_n} \sum_{t=0}^n W_t \left(\sum_{z=0}^n (M_{00})^{-z} a_{nz} D_{zt} \right) M_{tm}}{M_{0m}} \quad (36)$$

Where M_{nm} is the standard, vectorized or quaternion moment of order (n, m) .

6. Computational Aspects & Algorithms

The computation of quaternion or vectorized color moments with eqs. (30) or (26) can be accelerated by using the Euler's formula. Thus, the factors $e^{-\mu m \theta_k}$ and $e^{-i m \theta_k}$ in eqs. (30) and (26), respectively, can be rewritten as

$$e^{-\xi m \theta_k} = \cos(m \theta_k) - \xi \cdot \sin(m \theta_k) \quad (37)$$

where $\xi = \mu, i$. Using eq. (37), then eqs. (30) and (26) can be separated as follows

$$M_{nm}^C = \frac{1}{l \cdot W_n} \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} f_q(r, \theta_k) P_n(r) \cos(m \theta_k) \quad (38)$$

$$M_{nm}^S = \frac{1}{l \cdot W_n} \sum_{r=0}^{\lfloor \frac{N}{2} \rfloor - 1} \sum_{k=0}^{l-1} f_q(r, \theta_k) P_n(r) \sin(m \theta_k) \quad (39)$$

$$M_{nm} = M_{nm}^C - M_{nm}^S \cdot \xi \quad (40)$$

where M_{nm} is the standard, vectorized or quaternion moment of order (n, m) . In eqs. (38) and (39) only real numbers are multiplied. The only quaternion or vector multiplication takes place in eq. (40). Therefore, moments computation using the latter equation is faster than that of eqs. (30) and (26). In Fig. (4) the algorithm for computing the radial quaternion or vectorized color moments is presented. Furthermore, quaternion and vectorized color moment invariants can be calculated by using the algorithm illustrated in Fig. (5).

7. Experimental Analysis

In this section the reconstruction error, the computation time and the classification performance of the quaternion and vectorized color moments are examined.

7.1. Reconstruction

The following known images (size 128×128 pixels): Lena, f16, baboon and house from USC-SIPI Image Database [22], are used as indicative examples in order to compare the reconstruction error of radial quaternion and vectorized

DISCRETE RADIAL QUATERNION OR VECTORIZED COLOR MOMENTS

Input: P_n : n -th Degree Discrete Polynomial
Input: I : $N \times N$ Quaternion ($f_q(x, y)$) or Vectorized ($f_v(x, y)$) Image
Input: N : Image Dimension (Width & Height)
Input: n : Moment Order
Input: m : Moment Repetition
Input: l : the maximum number of pixels along the circumference of the circle
Input: W_n : Moment's Family Weighted Function

```

1: Procedure RADIAL MOMENTS( $P_n, I, N, n, m, l, W_n$ )
2:
3:    $MR \leftarrow \lfloor \frac{N}{2} \rfloor$  ▷ max radius
4:
5:    $\xi \leftarrow \mu$  or  $i$  ▷  $\mu$ : unit pure quaternion,  $i$ : complex unit
6:
7:    $M_{nm}^C \leftarrow 0$  ▷ moment's initialization
8:
9:    $M_{nm}^S \leftarrow 0$  ▷ moment's initialization
10:
11:   for  $r \leftarrow (0 : MR - 1)$  do
12:     for  $k \leftarrow (0 : l - 1)$  do
13:        $\theta_k \leftarrow \frac{2\pi k}{l}$ 
14:
15:        $x \leftarrow \lfloor r \left( \frac{N}{2(MR-1)} \right) \cos(\theta_k) + \frac{N}{2} \rfloor$ 
16:
17:        $y \leftarrow \lfloor r \left( \frac{N}{2(MR-1)} \right) \sin(\theta_k) + \frac{N}{2} \rfloor$ 
18:
19:        $M_{nm}^C \leftarrow M_{nm}^C + I(x, y) P_n(r) \cos(m\theta_k)$ 
20:
21:        $M_{nm}^S \leftarrow M_{nm}^S + I(x, y) P_n(r) \sin(m\theta_k)$ 
22:     end for
23:   end for
24:
25:    $M_{nm} \leftarrow M_{nm}^C - M_{nm}^S \cdot \xi$ 
26:
27:    $M_{nm} \leftarrow \frac{M_{nm}}{l \cdot W_n}$ 
28:
29:   return  $M_{nm}$ 
30: end Procedure

```

Figure 4. Discrete radial quaternion moments computation algorithm.

color moments. These moments are computed, for each image, from zero order up to order (40, 40) with step 5. The Mean Absolute Percentage Error (MAPE), given by eq. (41), is used in order to measure the error between the reconstructed, resulted by eqs. (33) and (32), and the original images. The results are illustrated in Fig. (6).

$$MAPE = \frac{100}{N^2} \cdot \sum_{x=1}^N \sum_{y=1}^N \frac{|I(x, y) - R(x, y)|}{I(x, y)} \quad (41)$$

 RADIAL DISCRETE QUATERNION OR VECTORIZED MOMENT INVARIANTS

Input: M : Quaternion or Vectorized Color Moments Up To Order (n, m)
Input: n : Moment Order
Input: m : Moment Repetition
Input: A : Matrix of Polynomial Coefficients
Input: D : Pseudo-Inverse of Matrix A
Input: W_n : Moment's Family Weighted Function

```

1: Procedure RADIAL-QUAT-MOM-INV( $M, n, m, A, D, W_n$ )
2:
3:    $\xi \leftarrow \mu$  or  $i$                                  $\triangleright \mu$ : unit pure quaternion,  $i$ : complex unit
4:
5:    $J_{nm}^{sr} \leftarrow 0$                                  $\triangleright$  moment's initialization
6:
7:   for  $t \leftarrow (0 : n)$  do
8:     for  $k \leftarrow (0 : n)$  do
9:        $J_{nm}^{sr} \leftarrow J_{nm}^{sr} + \frac{W_t}{W_n} A_{nz} D_{zt} \frac{(M_{00})^{-z} M_{tm}}{M_{0m}}$ 
10:    end for
11:  end for
12:
13:  return  $J_{nm}^{sr}$ 
14: end Procedure
  
```

Figure 5. Algorithm for discrete radial quaternion moment invariants computation.

Where $N = 128$ (pixels), I is the original image and R is the corresponding reconstructed image.

From Fig. (6) it can be clearly concluded that both quaternion and vectorized color moments result almost the same quality of reconstructed image, with the vectorized moments to have slightly better performance.

7.2. Computation Time

The same four images as in the subsection 7.1, are used in order to compare the computation time of quaternion and vectorized color moments. Initially, these moments are computed up to order (40, 40) with step 5 for each image. Afterwards, the Percentage Difference of the Computation Time (PD_{tr}) between the vectorized and quaternion color moments is estimated by using eq. (42).

$$PD_{tr} = 100 \cdot \frac{V_{tr} - Q_{tr}}{V_{tr}} \quad (42)$$

Where V_{tr} is the computation time of vectorized color moments up to order r , for image t , ($t = 1, 2, 3, 4$), while Q_{tr} is the corresponding

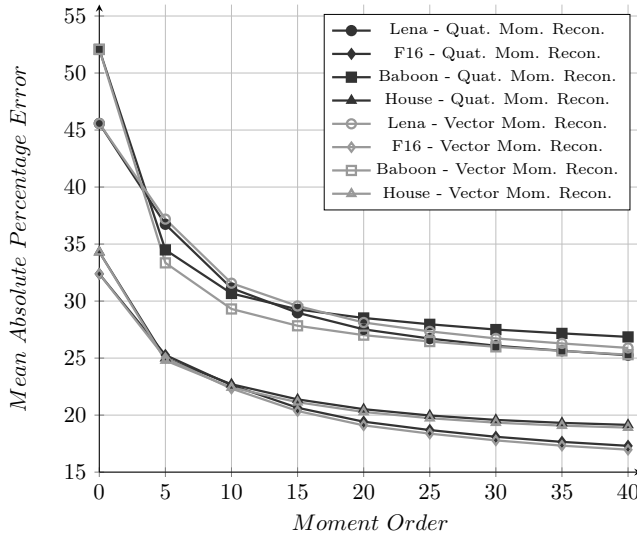


Figure 6. Reconstruction error of Lena, F16, Baboon and House images.

time for the case of quaternion color moments. Subsequently, the Mean PD_{tr} ($MPD_r = \frac{PD_{1r} + PD_{2r} + PD_{3r} + PD_{4r}}{4}$) is calculated for every order r and finally, the overall mean percentage difference of the computation time ($OMPD$) between the quaternion and vectorized color moments is calculated, $OMPD = \text{mean}(MPD_r)$. The results are illustrated in Fig. (7).
 $\forall r \in \{0:5:40\}$

By examining the Fig. (7), it can be deduced that the quaternion color moments are faster than their vectorized counterparts for every examined order (on average $OMPD = 15.52\%$ faster than the vectorized moments).

7.3. Classification

In this subsection a classification example is presented in order to evaluate the performance of both vector and quaternion moment invariants.

7.3.1. Experimental Protocol

Used Datasets The ALOI [23] image dataset has been selected as the basis on which the used dataset has been produced. The ALOI dataset includes im-

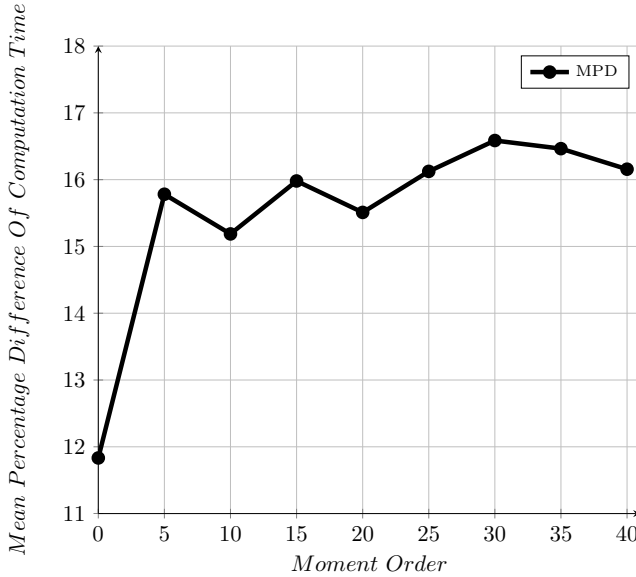


Figure 7. Mean percentage difference of the computation time between the quaternion and vector moment invariants.

ages that have been taken under four different options: 1) illumination direction, 2) illumination color, 3) object viewpoint and 4) wide-baseline stereo images. Here, the second option (illumination color) is selected to serve the aim of the corresponding classification scenario. The created dataset consists of twenty classes/objects (ALOI objects name: 1, 10, 11, ..., 19, 100, 101, ..., 108). For each class three different illumination conditions (illumination condition code: $i110$, $i170$ and $i250$) have been selected. These three images are rotated (rotation angles: 0° , 30° , 45° , 90°) and scaled (scaling factors: 0.70, 0.85, 1, 1.15, 1.30) for each object creating 60 $\{(3 \text{ illum. cond.}) \times (4 \text{ rot. angles}) \times (5 \text{ scal. factors})\}$ instances per class. Finally, Gaussian noise of zero mean and 0.01, 0.02, 0.03, 0.04, 0.05, 0.075 as well as 0.10 variance is added in the created dataset.

The three different illumination conditions have been used in order to illustrate the moments color sensitivity. Supposing that the selected pattern belongs to the first illumination condition, it is expected that the more the color sensitivity of a moment type, the worse the classification rate.

From now on, the above mentioned dataset will be called “*main*” dataset.

Except of the *main*, we define three more image sets, which are in fact subsets of the *main* one. These subsets, named *Si110*, *Si170* and *Si250*, include images with illumination conditions of code *i110*, *i170* and *i250*, respectively. Information regarding each dataset are presented in Table (1).

Table 1. Datasets Information

Dataset Name	Number of Classes	Number of Instances	Total Images
<i>Main</i>	20	60	1200
<i>Si110</i>	20	20	400
<i>Si170</i>	20	20	400
<i>Si250</i>	20	20	400

The above datasets are separated into train and test datasets. The 25% of images with illumination condition of code *i110* are selected from every class of the *main* dataset in order to form *main* train dataset (*TRD1*) {train dataset size: $0.25 \times (1 \text{ illum. cond.}) \times (4 \text{ rot. ang.}) \times (5 \text{ sc. fact.}) \times (20 \text{ classes}) = 100$ images}. The rest 1100 images {test dataset size: $[0.75 \times (1 \text{ illum. cond.}) + (2 \text{ illum. cond.})] \times (4 \text{ rot. ang.}) \times (5 \text{ sc. fact.}) \times (20 \text{ classes}) = 1100$ images} constitute the corresponding *main* test dataset (*TSD1*).

The train parts of *Si110*, *Si170* and *Si250* datasets, named *TRD2*, *TRD3* and *TRD4*, respectively, include 5 instances per class {train datasets size: $(5 \text{ instances}) \times (20 \text{ classes}) = 100$ images}. The corresponding test parts, named *TSD2*, *TSD3* and *TSD4*, respectively, consist of 300 images {test datasets size: $(15 \text{ instances}) \times (20 \text{ classes}) = 300$ images}. Information regarding each train and test set are presented in Table (2).

Table 2. Train and test sets information

Train Dataset	Number of Classes	Number of Instances	Total Images
<i>TRD1, TRD2, TRD3, TRD4</i>	20	5	100
Test Dataset	Number of Classes	Number of Instances	Total Images
<i>TSD1</i>	20	55	1100
<i>TSD2, TSD3, TSD4</i>	20	15	300

Feature Selection The radial Tchebichef vectorized and quaternion color moment invariants are computed up to order $(n, m)=(5, 5)$ {that is 36 moment invariants} for each train and test dataset. Thus, for the case of vectorized color moment invariants we have a feature space of $(6 \text{ values per moment}) \times (36 \text{ moments}) = 216$ elements for each instance, while for the case of quaternion color moment invariants, the feature space consists of $(4 \text{ values per moment}) \times (36 \text{ moments}) = 144$ elements.

In order to extract the desired feature vector we adopt the following process. Initially, the vectorized and quaternion color moment invariants, that have resulted from the train and test datasets, are normalized so as each feature to take values between the range $[0, 1]$. Afterwards, The FSDD [24] method is used to both type of invariants in order to locate the best 100 features. These features constitute our feature vector.

Classifier The KNN classifier with $N = 1$ and the Euclidean distance, given by eq. (43), are selected in order to evaluate the moment invariants.

$$E = \frac{\sqrt{\sum_{k=1}^L [P(k) - I(k)]^2}}{L} \quad (43)$$

Where E is the Euclidean distance between the pattern P and the instance I . L is the length of feature vectors (here $L = 100$).

The classification rate is measured by the following equation.

$$\text{Classification Rate} = \frac{\text{Number of correctly classified samples}}{\text{Total samples in the test dataset}} \times 100 \quad (44)$$

7.3.2. Results

The resulted rates of the classification process, using the above mentioned datasets, feature vector and classifier, are illustrated in Table (3). By examining the Table (3) can be deduced that for the test datasets $TSD2$, $TSD3$ and $TSD4$, both the quaternion and vectorized color moment invariants present similar results - the quaternion color moment invariants present slightly better behavior - not only for the case of noise free, but also in noisy conditions. However, an interesting difference is located in the first dataset, $TSD1$, for the case of noise free conditions, where we see that the classification rate of the quaternion

Table 3. Classification rates for the radial Tchebichef quaternion and vectorized color moment invariants

Moment Invariants	Dataset	Noise Free	Gaussian Noise Variance							Mean
			0.01	0.02	0.03	0.04	0.05	0.075	0.10	
Radial Quaternion	<i>TSD1</i>	88.45	76.64	75	70.73	68.36	69.27	65.73	63.64	72.23
	<i>TSD2</i>	100.00	99.33	99.33	94.33	93.67	93.67	91.00	85.67	94.63
	<i>TSD3</i>	99.67	100.00	99.00	96.67	97.00	97.00	93.67	91.00	96.75
	<i>TSD4</i>	100.00	99.67	99.00	98.67	98.67	97.00	96.00	92.33	97.67
Radial Vector	<i>TSD1</i>	98.91	73.73	73.91	70.36	66.27	67.45	66.73	65.00	72.80
	<i>TSD2</i>	100.00	98.67	95.67	97.67	93.67	93.67	90.67	87.00	94.63
	<i>TSD3</i>	99.67	98.33	97.67	93.67	95.67	94.33	91.67	90.00	95.13
	<i>TSD4</i>	100.00	99.67	97.67	98.33	95.67	96.33	95.33	90.67	96.71

color moment invariants is 88, 45%, while the corresponding rate of vectorized ones is 98, 91%. As mentioned, the dataset *TSD1* has been created in order to examine the moments color sensitivity. This dataset includes for each class the same 20 instances for three slightly different illumination/color conditions (total instances per class: $20 \times 3 = 60$). Thus, it is expected that the better the classification rate, the worse the color sensitivity. Considering the characteristic of *TSD1*, it can be deduced that the radial Tchebichef quaternion color moment invariants are more sensitive to color changes than their vectorized counterparts for noise free conditions. For the case of noisy conditions the two moment types present similar results.

7.4. Synopsis

The above experimental analysis showed that the quaternion color moments are calculated faster than their vectorized counterparts. Furthermore, both the two types of moments present similar quality of the reconstructed image as well as similar classification performance. Finally, the quaternion color moment invariants present better color sensitivity than the vectorized ones for the case of noise free conditions, while for the case of noisy conditions the two moment types present similar results. These conclusions are gathered and illustrated briefly in Table (4).

Table 4. Synopsis table

Radial Moments Type	Comput. Time	Reconst. Quality*	Classific. Perform.*	Color Sensitivity (noise free cond.)	Color Sensitivity* (noisy cond.)
Quaternion	✓		✓	✓	
Vector		✓			✓

*: Similar, slightly different results.

Conclusion

In this chapter a study of radial color moments and moment invariants was attempted. Two methods of representing color image information are examined. In the first one each color pixel is represented by a 3-element vector, while in the second, the color information is represented by a quaternion number. These two color representation schemes result in the corresponding radial vectorized and quaternion color moments and moment invariants. In the experimental analysis the Tchebichef polynomials are used as kernel function for both moment types. The results show that both moment types are very effective, nevertheless, the quaternion color moments achieve better computation time as well as classification rates. Finally, quaternion color moment invariants seem to be more sensitive to color changes than their vectorized counterparts for noise free conditions.

References

- [1] G. A. Papakostas, Y. S. Boutalis, D. A. Karras, and B. G. Mertzios, "A new class of zernike moments for computer vision applications," *Information Sciences*, vol. 177, no. 13, pp. 2802–2819, 2007.
- [2] G. A. Papakostas, D. E. Koulouriotis, and E. G. Karakasis, "Computing orthogonal moments in biomedical imaging," in *2009 16th International Conference on Systems, Signals and Image Processing, IWSSIP 2009*, 2009.

-
- [3] G. A. Papakostas, E. G. Karakasis, and D. E. Koulouriotis, "Accurate and speedy computation of image legendre moments for computer vision applications," *Image and Vision Computing*, vol. 28, no. 3, pp. 414–423, 2010.
 - [4] G. A. Papakostas, E. G. Karakasis, and D. E. Koulouriotis, "Novel moment invariants for improved classification performance in computer vision applications," *Pattern Recognition*, vol. 43, no. 1, pp. 58–68, 2010.
 - [5] J. Flusser and T. Suk, "Pattern recognition by affine moment invariants," *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, 1993.
 - [6] K. M. Hosny, "Refined translation and scale legendre moment invariants," *Pattern Recognition Letters*, vol. 31, no. 7, pp. 533–538, 2010.
 - [7] M. R. Teague, "Image analysis via the general theory of moments.," *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.
 - [8] Y. Sheng and L. Shen, "Orthogonal fourier-mellin moments for invariant pattern recognition," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 11, no. 6, pp. 1748–1757, 1994.
 - [9] C. Singh and R. Upneja, "Accurate computation of orthogonal fourier-mellin moments," *Journal of Mathematical Imaging and Vision*, pp. 1–21, 2012.
 - [10] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by tchebichef moments," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1357–1364, 2001.
 - [11] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by krawtchouk moments," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1367–1377, 2003.
 - [12] H. Zhu, H. Shu, J. Zhou, L. Luo, and J. L. Coatrieux, "Image analysis by discrete orthogonal dual hahn moments," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1688–1704, 2007.
 - [13] R. Mukundan, "Radial Tchebichef Invariants for Pattern Recognition," *TENCON 2005 - 2005 IEEE Region 10 Conference*, pp. 1–6, Nov. 2005.

- [14] W. Hamilton, *Elements of Quaternions*. Longmans, Green and Co., London, 1866.
- [15] I. L. Kantor and A. S. Solodovnikov, *Hypercomplex Numbers: An Elementary Introduction to Algebras*. Springer-Verlag, New York, 1989.
- [16] S. J. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics Letters*, vol. 32, no. 21, pp. 1979–1980, 1996.
- [17] T. Ell, "Quaternion-Fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, pp. 1830–1841, 1993.
- [18] B. Chen, H. Shu, H. Zhang, G. Chen, and L. Luo, "Color image analysis by quaternion zernike moments," in *Proceedings - International Conference on Pattern Recognition*, pp. 625–628, 2010.
- [19] H. Zhu, H. Shu, T. Xia, L. Luo, and J. Louis Coatrieux, "Translation and scale invariants of tchebichef moments," *Pattern Recognition*, vol. 40, no. 9, pp. 2530–2542, 2007.
- [20] E. G. Karakasis, G. A. Papakostas, and V. D. Koulouriotis, D. E. and-Tourassis, "A unified methodology for computing accurate quaternion color moments and moment invariants," *Under revision*, 2011.
- [21] B. Xiao, M. Jian-Feng, and J.-T. Cui, "Invariant pattern recognition using radial tchebichef moments," in *2010 Chinese Conference on Pattern Recognition, CCPR 2010 - Proceedings*, pp. 1–5, 2010.
- [22] "<http://sipi.usc.edu/database/>."
- [23] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.
- [24] J. Liang, S. Yang, and A. Winstanley, "Invariant optimal feature selection: a distance discriminant and feature ranking based solution," *Pattern Recognition*, vol. 41, no. 5, pp. 1429–1439, 2008.

Chapter 6

PATTERN RECOGNITION BY BESSEL MASK AND ONE-DIMENSIONAL SIGNATURES

Selene Solorza^{1,} and Josué Alvarez-Borrego²*

¹Facultad de Ciencias, Universidad Autónoma de Baja California

²Applied Physics Division, Optics Department, CICESE

Abstract

Recently, invariant correlation digital systems to position, rotation, scale and illumination are utilized in the pattern recognition field [1-3]. Such invariants are made of by the Fourier and Fourier-Mellin transforms in conjunction with linear or nonlinear filters (k -law). In this work a new digital system invariant to position, rotation and illumination based on Fourier transform, Bessel masks, one-dimensional signatures and linear correlations are presented. Using one-dimensional signatures instead of diffraction patterns or vectorial signatures of the images reduces the computational time considerably, achieving a step toward the ultimate goal, which is to develop a simple digital system that accomplishes recognition in real time at a low cost. To achieve the invariant to translation the modulus of the Fourier transform of the image is taken. And, using a Bessel binary mask of concentric rings the invariant to rotation is obtained. The discrimination between objects is done by a linear correlation of the one-dimensional signatures assigned to each image and the target, in this manner the computational cost is reduced also. The images classification range are determined by the Fisher transformation statistic

*E-mail address: selene.solorza@gmail.com

theory. The digital system was tested using a reference image database of 21 fossil diatoms images of gray-scale and 307×307 pixel. The system has a confidence level of 95.4% or greater in the classification of the 7,560 problem images using the same illumination. Then, those problem images were altered with eight different illuminations and the system also identifies the 60,480 images with a confidence level of 95.4% or greater.

PACS 07.05.Pj

Keywords: Image processing algorithms, Image processing, pattern recognition

AMS Subject Classification: 68U10, 94A08, 68T10

1. The Bessel Mask

The digital system works with $n \times n$ gray-scale images. The binary mask is build using the Bessel function of first kind and first order as

$$f(x) = \frac{J_1(x - c_x)}{x - c_x}, \quad (1)$$

where $x = 1, \dots, n$ and the center pixel (c_x, c_x) of the image is given by

$$c_x = \begin{cases} \frac{n}{2} + 1, & \text{if } n \text{ is even,} \\ \lfloor \frac{n}{2} \rfloor + 1, & \text{if } n \text{ is odd,} \end{cases} \quad (2)$$

here $\lfloor z \rfloor$ rounds z to the nearest integer towards $-\infty$.

The function f is symmetric in $x = c_x$, hence

$$Z(x) = \begin{cases} 1, & \text{if } f(c_x) > 0, \\ 0, & \text{if } f(c_x) \leq 0, \end{cases} \quad (3)$$

also is symmetric. Finally, the Z function is rotating 180 degrees by the symmetric axes to generate concentric cylinders centered in (c_x, c_x) and height one. Those cylinders are mapped to the plain to obtain the Bessel mask given in Fig. 1.

2. The One-Dimensional Signature

Let $I(x, y)$ represents the intensity of the image I in the pixel (x, y) , where $x, y = 1, \dots, n$. The digital system works with the modulus of the Fourier

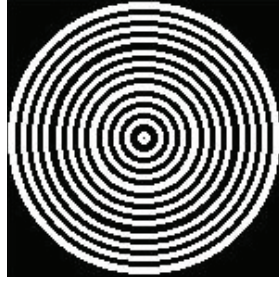


Figure 1. Bessel binary rings mask example.

transform of the image, hereafter called M , because it is invariant to translation, that is, $M = |\mathcal{F}\{I(x, y)\}| = |\mathcal{F}\{I(x + \tau, y + \chi)\}|$.

The Bessel mask, named B , filters the M image as

$$H = B * M, \quad (4)$$

where $*$ means element by element multiplication. Next, the rings in H are numbered from the center toward out-side to obtain the following set,

$$N = \{\text{ring index} \mid \text{ring index} \in \bar{n}\}, \quad (5)$$

where $\bar{n} = \{1, 2, \dots, \text{number of rings}\}$.

The addition of the intensity values into each ring of H are computed to build the function

$$\text{signature: } NA \subset \mathbb{R},$$

$$\text{signature: (ring index)} = \sum H(x, y), \text{ if } H(x, y) \text{ are in the ring.} \quad (6)$$

Because the cardinality of A always is bigger than one, the graph of the signature function is called *one-dimensional signature* of the image I .

3. The Pattern Recognition

Let R be the set of 21 diatoms reference image (RI) shown in Fig. 2 and PI a problem image to be classified. The PI could be translated and/or rotated in the Cartesian plane. The one-dimensional signatures of the RI and the PI are obtained as described in section 2.

Fig. 3 shows the signature of diatoms *Actinocyclus ingens*, *Azpeitia sp.*, *Azpeitia nodulifera* and *Actinocyclus ellipticus*, named image A, B, C and D, respectively, see Fig. 2. Although the diatoms *Azpeitia sp.* (B) and *Azpeitia nodulifera* (C) are so similar, their signatures are very different, hence we have a pattern recognition digital system.

The signatures of the PI and the k -th reference image (RI_k) are compared by the linear correlation equation,

$$C(S_{RI_k}, S_{PI}) = \mathcal{F}^{-1} \left\{ |\mathcal{F} \{S_{PI}\}| e^{i\phi_{PI}} \left| \mathcal{F} \{S_{RI_k}\} \right| e^{-i\phi_{RI_k}} \right\}, \quad (7)$$

where ϕ_{PI} and ϕ_{RI_k} are the phases of the Fourier transform for the signature of the PI and the k -th RI, respectively. If the maximum value of the magnitude for the linear correlation are significant, that is similar to the autocorrelation maximum value, hence the PI contains the RI, otherwise are different. To normalize the output results, those maximum values are scaled as

$$\frac{\max |C(S_{RI_k}, S_{PI})|}{(N-1)\sigma_{RI_k}\sigma_{PI}}, \quad (8)$$

where N is the length of the signatures. σ_{RI_k} and σ_{PI} are the standard deviations of the k -th RI and the PI signatures, respectively.

To obtain the confidence level of the system, the 21 diatom images in Fig. 2 are used as RI, each of them were rotated degree by degree until complete the 360 degrees, hence the PI data base has 7,560 images. The results were box plotting by the mean of the normalized values (eq. 8) with two standard errors (2SE) for the 360 images of each diatom. For example, Fig. 4 shows the box plot graph with diatom A as RI. We see in the box plot that results are normalized and there are not overlap of the whiskers associated to the RI, in this example diatom A, with the whiskers associated to the other diatom images. Therefore, the digital system has a confidence level of 95.4% in the diatom A identification. The same statistical analysis was done for each RI to obtained that, in general, the digital system has a confidence level of 95.4%.

Once the confidence level of the system is set, to do the classification process, first of all we have to notice that data does not has a known distribution curve. Hence, we use the Fisher transformation to have a normal distribution. Then, the confidence intervals of a 95.4% are set for identify each PI. Moreover, the PI could had different illumination of that images in the reference image data base and the digital system also classified it.

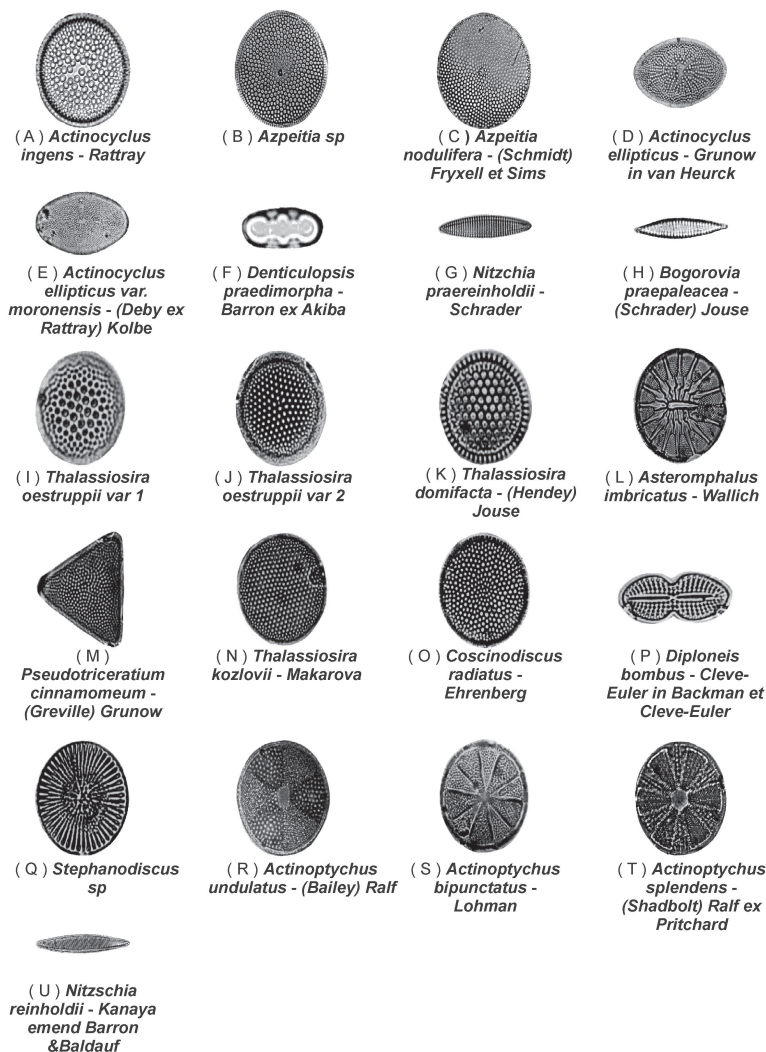


Figure 2. Reference image data base.

Conclusion

This work presents a low computational cost algorithm invariant to position, rotation and illumination. The digital system was tested using a reference im-

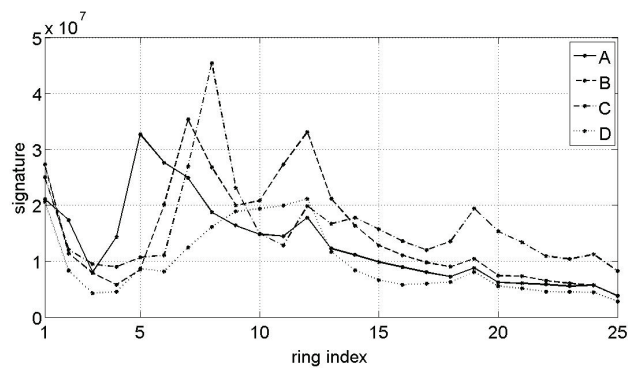


Figure 3. Signature examples.



Figure 4. Box plot example.

age database of 21 fossil diatoms gray-scale images of 307×307 pixel. The system classified 7, 560 problem images with a confidence level of 95.4% or greater. Furthermore, those problem images were altered with eight different illuminations background and the system also identifies the 60, 480 images.

Acknowledgments

This work was partially supported by CONACyT under grant No. 102007 and 169174.

References

- [1] Solorza, S. & Álvarez-Borrego, J. (2010). Digital system of invariant correlation to position and rotation. *Optics Communications*, Vol. 283, No. 19, 3613-3630, ISSN 0030-4018.
- [2] Álvarez-Borrego, J. & Solorza, S. (2010). Comparative analysis of several digital methods to recognize diatoms. *Hidrobiológica*, Vol. 20, No. 2, 158-170, ISSN 0188-8897.
- [3] Solorza, S., Álvarez-Borrego, J. and Chaparro-Magallanez, G. (2012). Pattern recognition of digital images by one-dimensional signatures. Chapter 13, pp. 299-316. *Fourier Transform-Signal Processing*. Editor Salih Mohammed Salih. Ed. Intech.

INDEX

A

absorption spectroscopy, 7
accurate models, 92
acid, 14, 37
acidity, 31, 35
adaptability, 99
adaptation, 51
Africa, 40
agar, 38
agriculture, 2, 27, 32, 50
algorithm, viii, 9, 12, 14, 15, 18, 19, 22, 26, 28, 43, 48, 50, 54, 57, 58, 59, 60, 67, 78, 79, 81, 83, 86, 92, 98, 102, 105, 107, 111, 112, 114, 117, 126, 132, 135, 143, 166, 167, 181
Algorithmic Probability (ALP), viii, 125, 127, 140
amplitude, 79, 117
anticancer drug, 26, 34
Artificial Neural Networks, 34, 88
assessment, 28, 86
atoms, 2, 4, 6

B

bacteria, 14, 20, 37, 42
bacterial colonies, 31
bacterial pathogens, 33
bacterial strains, 39
basal cell carcinoma, 20

base, ix, 14, 40, 86, 92, 94, 95, 125, 127, 128, 132, 134, 137, 138, 140, 141, 144, 147, 152, 180, 181
beef, 28, 40
bias, 22, 24, 25, 67
biochemistry, 4, 6
biomarkers, 26
biotechnology, 4
blood, 17, 39
bonds, 6
bounds, 137
brain, 19, 48
breakdown, 20, 25, 36, 38, 39, 44

C

calibration, 2, 12, 21, 22, 23, 32, 34
cancer, 11, 19, 20, 26, 31, 33, 35, 37
cancerous cells, 87
carbon, 23, 37, 42
carcinoma, 20
carotenoids, 42
case studies, 76
categorization, 54, 84, 157, 158
category d, 57
cell differentiation, 42
cell lines, 26
cell surface, 26, 43
challenges, vii, 2, 23, 49
chemical, 2, 4, 5, 9, 11, 22, 23, 27, 41, 43, 48, 49, 54, 66, 68, 84, 85

chemical bonds, 6
 chemometrics, 2, 36, 39, 41, 44
 classes, 12, 15, 29, 30, 55, 63, 64, 77, 99,
 110, 111, 112, 119, 128, 130, 131, 143,
 145, 146, 148, 170, 171
 classical methods, 112
 classification, viii, ix, x, 2, 3, 4, 8, 14, 15,
 16, 17, 18, 19, 20, 24, 26, 28, 29, 30, 31,
 35, 38, 40, 42, 43, 47, 48, 50, 57, 61, 69,
 81, 83, 84, 87, 90, 100, 110, 128, 140,
 145, 146, 151, 152, 153, 154, 169, 170,
 172, 173, 175, 177, 178, 180
 cluster analysis, 9, 11, 12, 13, 14
 clustering, vii, 2, 3, 11, 12, 13, 14, 19, 137,
 138, 140
 clusters, 11, 13, 14, 15, 118, 138, 139
 coding, 55, 56, 127, 130, 132, 134, 135,
 137, 144, 145, 150
 coffee, 9, 10, 16, 17, 36, 41, 42
 color, vii, ix, 13, 27, 28, 33, 50, 153, 154,
 155, 162, 163, 164, 166, 167, 168, 169,
 170, 172, 173, 174, 176
 comparative analysis, 30
 complement, 9, 55, 56, 57, 92
 complex numbers, 155, 156
 complexity, vii, 1, 29, 32, 93, 98, 115, 127,
 129, 130, 132, 133, 134, 135, 136, 138,
 140, 141, 147, 150, 151, 152
 composition, 6, 25, 26, 27, 43, 48, 85, 141
 compression, 152
 computation, x, 12, 21, 166, 167, 168, 169,
 170, 174, 175
 computer, 19, 20, 68, 92, 94, 98, 100, 102,
 174, 175
 computing, 18, 68, 92, 101, 166, 176
 construction, ix, 99, 125, 126
 consumption, 49, 51, 67
 convergence, 92, 99, 141
 cooking, 28, 43
 copper, 66
 correlation coefficient, 25
 correlations, x, 3, 12, 21, 25, 177, 180
 cost, x, 22, 49, 54, 60, 84, 145, 150, 159,
 177, 181
 Costa Rica, 41

cross-validation, 18, 20, 22, 26, 63, 64, 152
 crystalline, 51
 crystallization, 51
 culture, 26
 cytogenetics, 87
 cytometry, 4

D

data analysis, 4, 9, 15, 23, 25, 26, 85
 data processing, 60
 data set, 8, 12, 14, 16, 19, 26, 29, 48, 63
 database, x, 42, 107, 114, 176, 178, 182
 decision trees, 37
 decoding, 115
 decomposition, 93, 119
 deep learning, 145
 dendrogram, 14
 dependent variable, 4
 depth, 113, 114, 116
 derivatives, 3, 10, 93, 99
 dermatology, 37
 desorption, 10
 detection, 5, 6, 16, 19, 28, 31, 33, 35, 36,
 39, 40, 42
 deviation, 22, 96, 97, 98
 diatoms, x, 178, 179, 180, 182, 183
 diffraction, x, 177
 diffuse reflectance, 10, 23, 36, 42
 dimensionality, 8, 23, 24, 32
 discretization, 105, 116
 discriminant analysis, 3, 12, 18, 38, 39, 42,
 44
 discrimination, ix, x, 9, 14, 16, 18, 20, 33,
 39, 43, 85, 125, 130, 131, 132, 143, 149,
 150, 152, 177
 disorder, 40
 dispersion, 138
 distributed representation, 145
 distribution, 11, 26, 27, 28, 104, 106, 112,
 113, 114, 128, 129, 130, 133, 134, 137,
 138, 143, 145, 146, 148, 180
 diversity, 126
 DNA sequencing, 4
 drugs, 6, 26

E

ecology, 2, 27, 32
 electrodes, 50, 51, 66, 67, 69, 74, 80, 81, 83, 86, 87
 electrolysis, 66
 electromagnetic, 2, 4, 5, 7, 27
 electron, 4, 5, 7
 electronic systems, 48, 49
 EMG, 89
 emission, 2, 4, 5, 7
 encoding, 115, 116
 energy, 4, 6, 7, 54, 113
 engineering, 88, 112
 environment, vii, 2, 48, 53, 93, 115
 environments, 36, 53
 equality, 142
 equilibrium, 69
 equipment, viii, 48, 49, 50, 68
 ERS, 43
 erythrocytes, 16
 ESI, 10
 ESR, 4, 5
 Euclidean space, 104, 106
 evolution, 38
 experimental condition, 2
 explosives, 6
 extraction, 30

F

fabrication, 54
 false positive, 77
 feature selection, 39
 filters, x, 177, 179
 fingerprints, 18, 41
 Finland, 45
 Fisher transformation statistic theory, x
 fixation, 92, 117, 118
 fluctuations, 55
 fluorescence, 2, 4, 5, 7, 14, 19, 31, 35, 41
 food, 6, 18, 22, 28, 37, 43, 50, 51
 food industry, 54
 food products, 36, 84

food safety, 6
 forecasting, 88, 89
 formula, 12, 95, 104, 157, 160, 166
 foundations, 126, 132
 FTIR, 9, 14, 16, 17, 23, 36, 40, 41, 44
 fusion, 13, 41

G

gelation, 35
 gene expression, 26, 151
 genes, 26
 genomics, 24
 genotyping, 26
 genus, 32
 geographical origin, 18
 geology, 2, 27, 32
 geometry, 99
 glucose, 25, 44
 graph, 179, 180
 Graphical User Interface (GUI), viii, 47, 61
 grass, 28, 40
 grasses, 22, 42
 Greece, 153
 grids, 94, 95, 96, 97, 98, 99
 grouping, 26, 116
 growth, 5, 98

H

height, 178
 hemoglobin, 17
 higher education, 152
 human, 19, 26, 48, 129, 152
 human brain, 48
 humidity, 18
 hybrid, 99
 hypercube, 27, 28, 95

I

identification, 20, 26, 27, 28, 31, 36, 38, 44, 50, 86, 110, 180

illumination, x, 18, 170, 171, 173, 177, 178, 180, 181
 illusions, 84
 image, ix, x, 11, 27, 28, 31, 35, 54, 60, 137, 152, 153, 154, 155, 159, 161, 162, 163, 164, 165, 167, 168, 169, 171, 173, 174, 175, 176, 177, 178, 179, 180, 181
 image analysis, 35, 137, 154, 155, 176
 images, ix, x, 19, 27, 30, 31, 34, 36, 40, 41, 43, 45, 136, 151, 153, 155, 166, 167, 168, 169, 170, 171, 176, 177, 178, 180, 182, 183
 imitation, 93, 94, 103
 independence, 128, 142, 143
 independent variable, 4, 21
 individuals, 90
 induction, viii, 125, 127, 133, 151, 152
 induction methods, 127
 inequality, 96, 97, 116, 134
 infectious agents, 31
 information processing, 118
 infrared spectroscopy, 6, 16, 38, 39, 40, 41, 42, 44
 initial state, 107
 input signal, 67, 117
 integration, 104, 106
 intensity values, 179
 interpretability, 16, 25
 invariants, x, 154, 155, 159, 164, 166, 168, 169, 170, 172, 173, 174, 175, 176, 177
 ionization, 4, 10
 IR spectroscopy, 6
 Islam, 39
 Israel, 87
 Italy, 123
 iteration, 64

J

Jordan, 88
 jumping, 112, 117, 118

K

kidney, 11
 Kolmogorov complexity, 132, 140, 141, 151, 152

L

labeling, 54
 lactic acid, 14, 37
 laws, 112
 lead, 31, 135, 145, 165
 learning, viii, 24, 29, 32, 48, 51, 53, 54, 57, 84, 85, 125, 129, 131, 140, 141, 143, 145
 light, 2, 4, 5, 6, 13, 14, 21, 31, 33, 41
 linear correlations, x
 linear model, 18, 23, 32, 34
 lipids, 37
 localization, 5, 31
 luminescence, 5, 14, 41

M

machine learning, viii, 125, 129
 magnetic resonance, 4, 5, 7
 magnitude, 6, 180
 manifolds, 29
 mapping, 28, 99
 mass, 10, 11, 26, 86, 134, 135, 136
 mass spectrometry, 10, 11, 86
 material sciences, 6
 materials, 20, 50, 51, 54, 66
 mathematical methods, 2
 mathematical programming, 101
 matrix, 8, 9, 12, 13, 15, 18, 21, 24, 25, 61, 63, 65, 69, 71, 75, 80, 81, 137, 138, 139, 165
 measurements, 2, 5, 8, 44, 48, 50, 51, 54, 66, 67, 68, 69, 84, 110, 118
 meat, 28, 38, 85
 medical, 4, 31, 45, 87
 medicine, 2, 32
 melanoma, 20

memory, viii, 47, 49, 50, 51, 53, 54, 60, 61, 67, 77, 79, 118
 memory capacity, 53
 message length, 151
 metabolome, 39
 metals, viii, 47
 metaphor, 19
 methodology, 11, 117, 176
 microcontroller, vii, viii, 47, 49, 50, 51, 60, 61, 62, 67, 68, 75, 77, 78, 79, 80, 81, 83, 85
 microscopy, 35
 Minimum Description Length (MDL), viii, 125, 127
 Minimum Message Length (MML), viii, 125, 127
 mixture analysis, 86
 modelling, 34, 38
 models, ix, 2, 3, 8, 16, 17, 18, 23, 26, 32, 34, 42, 48, 86, 92, 93, 94, 111, 125, 126, 127, 128, 129, 130, 132, 133, 134, 137, 138, 139, 141, 143, 147, 148, 150, 152
 modules, 52
 modulus, x, 156, 177, 178
 moisture, 22
 moisture content, 22
 molecular biology, 6
 molecular structure, 20
 molecules, 2, 4, 5, 6, 7
 morphological variability, 53
 Moscow, 91, 119, 120, 121, 122, 123, 124
 multidimensional, 11, 29, 92, 94, 95, 98, 99, 108, 118
 multiple regression, 9
 multiplication, 155, 156, 163, 166, 179
 multivariate analysis, 38
 multivariate calibration, 22
 mutant, 14
 mutation rate, 31

N

navigation system, 112
 near infrared spectroscopy, 39, 40, 41, 42
 negative consequences, 126

neglect, 101
 neural network(s), vii, viii, 2, 3, 19, 37, 38, 39, 40, 43, 44, 45, 48, 49, 50, 51, 60, 67, 68, 69, 79, 84, 85, 87, 88, 89, 90, 127
 neural systems, 48
 neurons, 19, 48
 New Zealand, 35, 151
 NIR, 16, 26, 31, 35, 37, 38, 42, 44
 NIR spectra, 28
 nitrogen, 28, 37
 nodes, 55, 56, 99
 normal distribution, 15, 137, 180
 nuclear magnetic resonance (NMR), 4, 5, 7, 18, 34, 38, 39, 40
 nuclei, 19, 35
 nutrition, 36

O

obstacles, 103
 olfaction, 84
 olive oil, 14, 18, 37, 41, 85
 one-dimensional signatures, vii, x
 on-line mode processing facilities, vii, 1
 operations, 52, 92
 optimization, vii, viii, 14, 20, 25, 91, 94, 100, 103, 104, 105, 106, 107, 110, 111, 115, 126, 127, 137
 optimization method, 93, 101, 102
 orthogonality, 157, 158, 159
 overlap, 110, 118, 119, 180

P

Pacific, 34, 151
 parallel, 26, 84, 92, 99
 Pareto, 103, 104, 106
 Partial Least Squares (PLS), 3, 18, 21, 23, 26, 28, 32, 35, 37, 39, 44, 86
 partial least squares regression, 3, 18, 26, 44
 pasteurization, 50, 51
 pathogens, 31, 33, 41
 pathology, 31

pattern recognition, vii, viii, ix, x, 2, 4, 19,
 23, 30, 32, 49, 53, 84, 85, 86, 90, 91,
 109, 125, 126, 128, 130, 133, 134, 135,
 137, 138, 140, 142, 144, 150, 152, 153,
 175, 176, 177, 180
 PCR, 3, 21, 23
 pharmaceutical, 22, 39, 42
 phosphorescence, 5
 photons, 6
 physical treatments, 50, 68
 physiological, 37
 physiology, 19
 plasticity, 49
 polar, 154, 155, 157, 158, 159, 161, 162,
 163, 164
 polarization, 6
 pollen, 50
 pollutants, 6
 polymer materials, 20
 poultry, 31, 36
 predictor variables, 18
 preparation, 92
 Principal Components Analysis (PCA), vii,
 2, 3, 8, 9, 10, 12, 15, 21, 24, 28, 32, 40
 principles, viii, 125, 127, 150
 prior knowledge, 8
 probability, vii, 16, 112, 113, 115, 116, 117,
 126, 127, 128, 129, 130, 132, 134, 138,
 140, 141, 142, 143, 144, 150, 151, 152
 probability density function, 138
 probability distribution, 113, 128, 134
 probability theory, 152
 probe, 6, 95
 producers, 50
 programming, 31, 61, 101
 project, 83
 propagation, 54
 prostate cancer, 31, 33
 proteins, 35, 37
 prototype, 20
 pruning, 43

Q

quadratic programming, 31

quality control, 50, 51
 quantification, 22, 31, 50
 quantitative analysis, vii, 2, 3, 33
 quaternion, vii, x, 154, 155, 156, 157, 163,
 164, 165, 166, 167, 168, 169, 170, 172,
 173, 174, 176

R

radiation, 2, 4, 5, 7
 radio, 2, 4, 112, 113, 114
 radioactive tracer, 5
 radius, 109, 161, 167
 Raman spectra, 20
 Raman spectroscopy, 6, 7, 14, 16, 20, 23,
 33, 36, 37, 38, 42, 43, 44
 reagents, 22
 real numbers, 166
 real time, x, 31, 177
 Receiver Operating Characteristic (ROC),
 80, 81
 reception, 117, 118
 recognition, vii, viii, ix, x, 2, 3, 4, 19, 23,
 30, 31, 32, 49, 53, 54, 60, 61, 62, 63, 64,
 65, 71, 74, 75, 76, 77, 80, 81, 83, 84, 85,
 86, 88, 89, 90, 91, 110, 111, 118, 119,
 125, 126, 128, 130, 131, 133, 134, 135,
 137, 138, 140, 142, 144, 145, 147, 148,
 149, 150, 152, 175, 178, 183
 reconstruction, x, 25, 154, 166
 recurrence, 160
 redundancy, 21, 54, 154
 reflectance spectra, 10
 regression, vii, 2, 3, 4, 9, 18, 21, 22, 23, 24,
 25, 26, 29, 32, 36, 44, 45, 129, 151
 regression analysis, 4, 22
 regression method, viii, 2, 23, 26
 regression model, 3, 4, 21
 rejection, 145, 150
 reliability, 89, 113, 114
 remote sensing, 27, 28, 30, 32, 34, 40, 88
 repairation, 92
 repetitions, 68
 resolution, 34, 67
 resonator, 112

respiratory syncytial virus, 14, 42
 response, 4, 18, 21, 22, 24, 26, 53, 56, 68, 117
 restrictions, 100, 101, 102, 105, 114, 139
 rings, x, 177, 179
 risk, ix, 22, 126
 ROI, 28
 root-mean-square, 26
 routine analysis, vii, 1, 22
 routines, 78
 rules, 108, 156
 Russia, 91, 125

S

safety, 6, 28, 37
 Salmonella, 32
 scaling, 3, 154, 162, 165, 170
 scatter, 4, 9, 24, 31
 scatter plot, 9
 scattering, 2, 4, 5, 7, 31, 33, 38, 41
 scripts, 50
 secondary information, 118
 security, 54
 selectivity, 22
 sensing, 27, 28, 30, 32, 34, 40, 88
 sensitivity, 2, 20, 22, 26, 30, 31, 34, 63, 76, 170, 173
 sensors, 25, 42, 48, 51, 66
 sequencing, 4, 26
 shock, 51
 showing, 7, 51
 signals, 48, 67, 88, 117, 118
 signal-to-noise ratio, 115, 116
 signs, 20
 simple quaternion algebra, x
 Simplified Fuzzy ARTMAP network, viii, 47
 simulation, vii, viii, 19, 35, 91, 93, 94, 100, 101, 102, 103, 104, 105, 106, 107, 110, 111, 114, 115, 116, 117
 simulations, 93, 117
 skin, 31, 36
 smoothing, 3
 smoothness, 99

solid phase, 2, 4
 solution, viii, ix, 8, 25, 93, 94, 99, 102, 103, 104, 107, 108, 109, 111, 112, 115, 125, 129, 131, 137, 143, 149, 150, 176
 solution space, viii, ix, 125
 South Africa, 40
 Spain, 47, 85
 spatial information, 11
 species, 32
 spectral dimension, 27
 spectroscopy, vii, 1, 2, 4, 5, 6, 7, 14, 16, 17, 18, 20, 23, 24, 25, 26, 27, 32, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 48, 84
 spin, 4, 5
 stability, 30, 49
 standard deviation, 180
 states, 4, 5, 6, 7, 22, 30, 57, 93, 116, 127, 130, 150
 statistics, 117, 118
 storage, 51, 85
 stress, 28, 37
 structure, 8, 15, 20, 30, 37, 93, 99, 100, 102, 103, 115
 subgroups, 39
 substrate, 6, 18
 surface chemistry, 2, 4
 synthesis, 49

T

techniques, vii, viii, 1, 2, 3, 4, 5, 7, 8, 9, 11, 18, 25, 29, 32, 36, 41, 42, 44, 47, 48, 50, 85, 126, 128
 technology, 19, 28
 temperature, 18
 test data, 64, 171, 172
 testing, vii, 1, 63, 74, 92
 textural character, 35
 texture, 31
 thermal treatment, 77
 tin oxide, 86
 tissue, 28, 31, 37, 41, 43
 tracks, 117
 training, viii, ix, 3, 8, 15, 17, 18, 19, 25, 29, 30, 32, 48, 54, 58, 60, 61, 63, 64, 65, 67,

68, 71, 72, 74, 75, 79, 81, 83, 126, 127,
128, 129, 133, 138, 139, 142, 144, 145,
147, 148, 150
transformation, x, 8, 20, 29, 43, 95, 99, 154,
159, 165, 177, 180
translation, x, 154, 164, 175, 177, 179
transmission, 112, 113, 114, 115
treatment, vii, 1, 17, 69, 77
tumors, 31, 36
two-dimensional space, 98, 108, 109

U

UK, 42
universality, 133
USA, 34
UV, 2, 4, 5, 7, 14, 18, 37, 42, 47

V

validation, 3, 8, 16, 18, 19, 20, 22, 25, 26,
34, 39, 63, 64, 65, 71, 73, 74, 75, 77, 83,
152
variables, vii, 1, 3, 4, 8, 12, 14, 15, 18, 21,
22, 23, 24, 25, 26, 32, 34, 63, 64, 65, 71,
74, 75, 81, 95, 117, 133, 161

variance-covariance matrix, 12
variations, 6, 43
vector, viii, ix, 2, 3, 9, 18, 19, 23, 31, 40, 52,
53, 55, 56, 57, 60, 61, 63, 67, 79, 94, 95,
100, 103, 104, 105, 106, 107, 111, 125,
128, 130, 131, 137, 152, 154, 162, 166,
168, 169, 170, 172, 174
Viagra, 14, 36
vibration, 6
vision, 174, 175
visualization, 26

W

water, 7, 28, 45
water quality, 28, 45
watershed, 43
wavelengths, 6, 21, 23, 26, 27
wavelet, 151
wires, 66

Y

yield, 28, 43, 107, 147, 149