

Advances in Computer Vision and Pattern Recognition



Marcello Pelillo *Editor*

Similarity-Based Pattern Analysis and Recognition

 Springer

The Springer logo, which is a stylized white chess knight (horse) facing left, positioned above the word "Springer" in a white serif font.

Advances in Computer Vision and Pattern Recognition

For further volumes:
www.springer.com/series/4205

Marcello Pelillo

Editor

Similarity-Based Pattern Analysis and Recognition

 Springer

Editor

Marcello Pelillo
DAIS
Ca' Foscari University
Venice, Italy

Series Editors

Sameer Singh
Rail Vision Europe Ltd.
Castle Donington
Leicestershire, UK

Sing Bing Kang
Interactive Visual Media Group
Microsoft Research
Redmond, WA, USA

ISSN 2191-6586

Advances in Computer Vision and Pattern Recognition

ISBN 978-1-4471-5627-7

DOI 10.1007/978-1-4471-5628-4

Springer London Heidelberg New York Dordrecht

ISSN 2191-6594 (electronic)

ISBN 978-1-4471-5628-4 (eBook)

Library of Congress Control Number: 2013955585

© Springer-Verlag London 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

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

For my parents, who made it possible

“Surely there is nothing more basic to thought and language than our sense of similarity. [...] And every reasonable expectation depends on resemblance of circumstances, together with our tendency to expect similar causes to have similar effects.”

Willard V.O. Quine

Foreword

The SIMBAD project was a Future and Emerging Technologies (FET) project funded by the European Commission between 2008 and 2011. It brought together an extraordinary group of talented researchers with a broad spectrum of different perspectives on the central theme of using non-Euclidean similarity functions as the basis for learning. This approach was in contrast with the use of kernel functions that had become the de facto standard at the time of the project's launch in 2008.

The SIMBAD project took a broad view of the problem of so-called non-Euclidean learning: analysing the extent to which this was essential in a particular problem, developing alternative learning strategies that could successfully learn from non-Euclidean similarity functions, developing methods of learning Euclidean representations from probabilistic models and similarity data, and so on. These approaches were not studied just in the abstract but rather were grounded in a series of concrete problems from application domains where it was known or suspected that the Euclidean assumption was unrealistic.

The number and depth of the papers that arose from this research agenda was very impressive, with significant innovations made on all of the fronts listed above. However, the research was not merely a shotgun attack on several divergent fronts, but rather represented the coherent development of the leitmotiv of the project: the use of similarity functions in learning.

Given the breadth of the reach and impact of the research, the project reviewers were fearful that this coherence might be lost in the variety of journals, conferences, and particular problems considered, hence risking that the main message become lost in the plethora of individual results.

It was therefore proposed that a book bringing together the themes of the project and its main results could help champion and communicate the SIMBAD message in one coherent volume. This carefully constructed book is the result of that proposal. It is a distillation of the main themes and results of the project into an accessible and cross-referenced volume. For those interested in learning about the potential and importance of learning from similarity functions, this work is undoubtedly the

key reference from which to begin their study and it is likely to remain so for many years to come.

Virginia Water
June 2013

John Shawe-Taylor

Preface

This book provides a thorough description of a selection of results achieved within SIMBAD, an EU FP7 project which represents the first systematic attempt at bringing to full maturation a paradigm shift that is just emerging within the pattern recognition and machine learning domains, where researchers are becoming increasingly aware of the importance of similarity information *per se*, as opposed to the classical (feature-based) approach.

SIMBAD started in April 2008 and ended in September 2011, and involved the following six partners:

- University of Venice, Italy (*scientific coordinator*)
- University of York, UK
- Delft University of Technology, The Netherlands
- Instituto Superior Tecnico, Lisbon, Portugal
- ETH Zurich, Switzerland
- University of Verona, Italy.

The very end of the project marked also the launch of the SIMBAD workshop series <http://www.dsi.unive.it/~simbad>

whose first edition was held in Venice, in September 2011, in conjunction with the project's final review meeting. These biennial workshops aim to consolidate and promote research efforts in this area and to provide an informal discussion forum for researchers and practitioners.

Within the SIMBAD project we undertook a thorough study of several aspects of purely similarity-based pattern analysis and recognition methods, from the theoretical, computational, and applicative perspective. We covered a wide range of problems and perspectives. We considered both supervised and unsupervised learning paradigms, generative and discriminative models, and our interest ranged from purely theoretical problems to real-world practical applications. The chapters collected in this book aim to provide a coherent overview of our main achievements and to serve as a starting point for graduate students and researchers interested in

this important, yet diverse subject. More details on the project's activities can be found on our website

<http://simbad-fp7.eu>

and in the published papers referenced in this book.

A project like SIMBAD could not have been done without the help and support of many people and institutions, and it is a pleasure to take this opportunity to express my gratitude to them. In the first place, I'd like to acknowledge the Future and Emerging Technology (FET) Programme of the 7th Framework Programme for Research of the European Commission which funded the SIMBAD project, and I am very grateful to our project officer, Teresa De Martino, and to the reviewers, Georgios Sakas, Christoph Schnörr and John Shawe-Taylor, whose insightful suggestions and constant encouragement have been instrumental to make SIMBAD a better project.

It has been my good fortune to collaborate for almost four years with a fantastic group of people, whose genuine enthusiasm and exceptional professional competence made SIMBAD a unique, intellectually stimulating experience. In particular, I'm grateful to my fellow principal investigators who coordinated the activities of the various research units: Joachim Buhmann, Bob Duin, Mario Figueiredo, Edwin Hancock, and Vittorio Murino; to their deputies: Manuele Bicego, Umberto Castellani, Ana Fred, Marco Loog, Volker Roth, and Richard Wilson; and to all PhD students and postdocs who have worked within the project.

In Venice, I've been helped by many people in my group, and I'd like to thank them all for their support. In particular, I wish to thank Andrea Torsello for the assistance he gave me at various stages of the project, and Veronica Giove for her valuable work concerning all administrative aspects. Special thanks are due to Samuel Rota Bulò for his constant support throughout the project and for helping me assemble this book.

I'd like to thank the editorial staff at Springer, in particular Wayne Wheeler for supporting the idea of publishing this book, and Simon Rees for his advice throughout the production of the volume and for gently tolerating my procrastinations.

My deepest gratitude, however, goes to my wife, Rosanna, and my children, Claudia and Valerio, without whose endless patience and understanding the SIMBAD project, and hence this book, would have not seen the light.

Venice
July 2013

Marcello Pelillo

Contents

1	Introduction: The SIMBAD Project	1
	Marcello Pelillo	
Part I Foundational Issues		
2	Non-Euclidean Dissimilarities: Causes, Embedding and Informativeness	13
	Robert P.W. Duin, Elżbieta Pełkalska, and Marco Loog	
3	SIMBAD: Emergence of Pattern Similarity	45
	Joachim M. Buhmann	
Part II Deriving Similarities for Non-vectorial Data		
4	On the Combination of Information-Theoretic Kernels with Generative Embeddings	67
	Pedro M.Q. Aguiar, Manuele Bicego, Umberto Castellani, Mário A.T. Figueiredo, André T. Martins, Vittorio Murino, Alessandro Perina, and Aydın Ulaş	
5	Learning Similarities from Examples Under the Evidence Accumulation Clustering Paradigm	85
	Ana L.N. Fred, André Lourenço, Helena Aidos, Samuel Rota Bulò, Nicola Rebagliati, Mário A.T. Figueiredo, and Marcello Pelillo	
Part III Embedding and Beyond		
6	Geometricity and Embedding	121
	Peng Ren, Furqan Aziz, Lin Han, Eliza Xu, Richard C. Wilson, and Edwin R. Hancock	
7	Structure Preserving Embedding of Dissimilarity Data	157
	Volker Roth, Thomas J. Fuchs, Julia E. Vogt, Sandhya Prabhakaran, and Joachim M. Buhmann	

8 A Game-Theoretic Approach to Pairwise Clustering and Matching 179
Marcello Pelillo, Samuel Rota Bulò, Andrea Torsello,
Andrea Albarelli, and Emanuele Rodolà

Part IV Applications

**9 Automated Analysis of Tissue Micro-Array Images on the Example
of Renal Cell Carcinoma** 219
Peter J. Schüffler, Thomas J. Fuchs, Cheng Soon Ong, Volker Roth,
and Joachim M. Buhmann

**10 Analysis of Brain Magnetic Resonance (MR) Scans
for the Diagnosis of Mental Illness** 247
Aydın Ulaş, Umberto Castellani, Manuele Bicego, Vittorio Murino,
Marcella Bellani, Michele Tansella, and Paolo Brambilla

Index 289

Chapter 1

Introduction: The SIMBAD Project

Marcello Pelillo

Abstract This introductory chapter describes the SIMBAD project, which represents the first systematic attempt at bringing to full maturation a paradigm shift that is just emerging within the pattern recognition and machine learning domains, where researchers are becoming increasingly aware of the importance of similarity information per se, as opposed to the classical (feature-based) approach.

1.1 Motivations

The challenge of automatic pattern analysis and recognition (or machine learning) is to develop computational methods which learn, from examples, to distinguish among a number of classes, with a view to endow artificial systems with the ability to improve their own performance in the light of new external stimuli. This ability is widely recognized to be instrumental in building next-generation artificial cognitive systems (ACSs) which, as opposed to traditional machine or computer systems, can be characterized “as systems which cope with novel or indeterminate situations, which aim to achieve general goals as opposed to solving specific problems, and which integrate capabilities normally associated with people or animals.”¹ The socio-economic implications of this scientific endeavor are enormous, as ACSs will have applications in a wide variety of real-world scenarios ranging from industrial manufacturing to vehicle control and traffic safety, to remote and on-site (environmental) sensing and monitoring, and to medical diagnostics and therapeutics.

As a matter of fact, despite their technological applications, pattern recognition and machine learning can arguably be considered as a modern-day incarnation of an endeavor which has challenged mankind since antiquity. Fundamental questions pertaining to categorization, abstraction, generalization, induction, etc. have, in fact, been on the agenda of mainstream philosophy, under different names and guises,

¹From: *Artificial Cognitive Systems in FP7: A Report on Expert Consultations for the EU Seventh Framework Programme 2007–2013 for Research and Technology Development*.

M. Pelillo (✉)
DAIS, Università Ca' Foscari, Venice, Italy
e-mail: pelillo@dais.unive.it

since its inception. Indeed, as pointed out in [7], the very foundations of pattern recognition can be traced back to Aristotle and his mentor Plato who were among the firsts to distinguish between an “essential property” from an “accidental property” of an object, so that the whole field of pattern recognition can naturally be cast as the problem of finding such essential properties of a category. As Watanabe put it [20, p. 21]: “whether we like it or not, under all works of pattern recognition lies tacitly the Aristotelian view that the world consists of a discrete number of self-identical objects provided with, other than fleeting accidental properties, a number of fixed or very slowly changing attributes. Some of these attributes, which may be called ‘features,’ determine the class to which the object belongs.” Accordingly, the goal of a pattern recognition algorithm is to discern the essences of a category, or to “carve the nature at its joints.” In philosophy, this view is known as *essentialism* and has contributed to shape mainstream machine learning research in a such a way that it seems legitimate to speak about an essentialist paradigm.

During the nineteenth and the twentieth centuries, the essentialist world-view was subject to a massive assault from several quarters and it became increasingly regarded as an impediment to scientific progress. Strikingly enough, this conclusion was arrived at independently in at least three different disciplines, namely physics, biology, and psychology. In physics, anti-essentialist positions were held (among others) by Mach, Duhem, Poincaré, and in the late 1920s Bridgman, influenced by Einstein’s achievements, put forcefully forward the notion of operational definitions precisely to avoid the troubles associated with attempting to define things in terms of some intrinsic essence [4]. For example, the (special) theory of relativity can be viewed as the introduction of operational definitions for simultaneity of events and of distance, and in quantum mechanics the notion of operational definitions is closely related to the idea of observables. This point was vigorously defended by Popper [15], who developed his own form of anti-essentialism and argued that modern science (and, in particular, physics) was able to make real progress only when it abandoned altogether the pretension of making essentialist assertions, and turned away from “what-is” questions of Aristotelian-scholastic flavor.

In biology, the publication of Darwin’s *Origin of Species* in 1859 had a devastating effect on the then dominating paradigm based on the static, Aristotelian view of species, and shattered 2000 years of research which culminated in the monumental Linnaean system of taxonomic classification. According to Mayr [14], essentialism “dominated the thinking of the western world to a degree that is still not yet fully appreciated by the historians of ideas. [...] It took more than two thousand years for biology, under the influence of Darwin, to escape the paralyzing grip of essentialism.”

More recently, motivated by totally different considerations, cognitive scientists have come to a similar discontent towards essentialist explanations. Indeed, it has become increasingly clear that the classical essentialist, feature-based approach to categorization is too restrictive to be able to characterize the intricacies and the multifaceted nature of real-world categories. This culminated in the 1970s in Rosch’s now classical “prototype theory” which is generally recognized as having revolutionized the study of categorization within experimental psychology; see [13] for an

extensive account, and the recent paper by von Luxburg et al. [19] for a machine learning perspective.

Nowadays, anti-essentialist positions are associated with various philosophical movements including pragmatism, existentialism, deconstructionism, etc., and is also maintained in mathematics by the adherents of the structuralist movement, a view which goes back to Dedekind, Hilbert and Poincaré, whose basic tenet is that “in mathematics the primary subject-matter is not the individual mathematical objects but rather the structures in which they are arranged” [16, p. 201]. Basically, for an anti-essentialist what really matters is relations, not essences. The influential American philosopher Richard Rorty nicely sums up this “panrelationalist” view with the suggestion that there are “relations all the way down, all the way up, and all the way out in every direction: you never reach something which is not just one more nexus of relations” [17]. As an aside, we note that a similar dissatisfaction with the essentialist approach can also be found in modern link-oriented approaches to network analysis [8, 12].

Now, it is natural to ask: What is the current state of affairs in pattern recognition and machine learning? As mentioned above, the fields have been dominated since their inception by the notion of “essential” properties (i.e., features) and traces of essentialism can also be found, to varying degrees, in modern approaches which try to avoid the direct use of features (e.g., kernel methods). This essentialist attitude has had two major consequences which have greatly contributed to shape the fields in the past few decades. On the one hand, it has led the community to focus mainly on feature-vector representations. Here, each object is described in terms of a vector of numerical attributes and is therefore mapped to a point in a Euclidean (geometric) vector space, so that the distances between the points reflect the observed (dis)similarities between the respective objects. On the other hand, this has led researchers to maintain a reductionist position, whereby objects are seen in isolation and which therefore tends to overlook the role of relational, or contextual, information.

Feature-vector representations are indeed extremely attractive because geometric spaces offer powerful analytical as well as computational tools that are simply not available in other representations. In fact, classical pattern recognition methods are tightly related to geometrical concepts and numerous powerful tools have been developed during the last few decades, starting from linear discriminant analysis in the 1920s, to perceptrons in the 1960s, to kernel machines in the 1990s. However, there are numerous application domains where either it is not possible to find satisfactory features or they are inefficient for learning purposes. This modeling difficulty typically occurs in cases when experts cannot define features in a straightforward way (e.g., protein descriptors vs. alignments), when data are high dimensional (e.g., images), when features consist of both numerical and categorical variables (e.g., person data, like weight, sex, eye color, etc.), and in the presence of missing or inhomogeneous data. But, probably, this situation arises most commonly when objects are described in terms of structural properties, such as parts and relations between parts, as is the case in shape recognition [3]. This led in 1960s to the development of the structural pattern recognition approach, which uses symbolic

data structures, such as strings, trees, and graphs for the representation of individual patterns, thereby, reformulating the recognition problem as a pattern-matching problem.

It is clearly open to discussion to what extent the lesson learnt from the historical development of other disciplines applies to machine learning and pattern recognition, but it looks at least like that today's research in these areas is showing an increasing propensity towards anti-essentialist/relational approaches. Indeed, in the last few years, interest around purely similarity-based techniques has grown considerably. For example, within the supervised learning paradigm (where expert-labeled training data is assumed to be available) the now famous "kernel trick" shifts the focus from the choice of an appropriate set of features to the choice of a suitable kernel, which is related to object similarities. However, this shift of focus is only partial as the classical interpretation of the notion of a kernel is that it provides an implicit transformation of the feature space rather than a purely similarity-based representation. Analogously, in the unsupervised domain, there has been an increasing interest around pairwise algorithms, such as spectral and graph-theoretic clustering methods, which avoid the use of features altogether. Other attempts include Balcan et al.'s theory of learning with similarity functions [2], and the so-called collective classification approaches, which are reminiscent of relaxation labeling and similar ideas developed in computer vision back in the 1980s (see, e.g., [18] and references therein).

Despite its potential, however, presently the similarity-based approach is far from seriously challenging the traditional paradigm. This is due mainly to the sparsity and heterogeneity of the techniques proposed so far and the lack of a unifying perspective. On the other hand, classical approaches are inherently unable to deal satisfactorily with the complexity and richness arising in many real-world situations. This state of affairs hinders the application of machine learning techniques to a whole variety of relevant, real-world problems.

The main problem with purely similarity-based approaches is that, by departing from vector-space representations, one is confronted with the challenging problem of dealing with (dis)similarities that do not necessarily possess the Euclidean behavior² or not even obey the requirements of a metric. The lack of the Euclidean and/or metric properties undermines the very foundations of traditional pattern recognition theories and algorithms, and poses totally new theoretical/computational questions and challenges. In fact, this situation arises frequently in practice. For example, non-Euclidean or non-metric (dis)similarity measures are naturally derived when images, shapes or sequences are aligned in a template matching process. In computer vision, non-metric measures are preferred in the presence of partially occluded objects [27]. Other non-metric examples include pairwise structural alignments of proteins that focus on local similarity [5], variants of Hausdorff distance [18],

²A set of distances D is said to be *Euclidean* (or *geometric*) if there exists a configuration of points in some Euclidean space whose interpoint distances are given by D . In the sequel, the terms *geometric* and *Euclidean* will be used interchangeably. The term *(geo)metric* is an abbreviation to indicate the case of a distance that satisfies either the Euclidean or the metric properties.

normalized edit-distances [5], and also some probabilistic measures such as the Kullback–Leibler divergence. As argued in [27], the violation of the metric properties is often not an artifact of poor choice of features or algorithms, and it is inherent in the problem of robust matching when different parts of objects (shapes) are matched to different images. The same argument may hold for any type of local alignments. Corrections or simplifications may therefore destroy essential information.

In summary, there is an urgent need to bring to full maturation a paradigm shift that is just emerging within the pattern recognition and machine learning domains, where researchers are becoming increasingly aware of the importance of similarity information per se, as opposed to the classical feature-based (or vectorial) approach. Indeed, the notion of similarity (which appears under different names such as proximity, resemblance, and psychological distance) has long been recognized to lie at the very heart of human cognitive processes and can be considered as a connection between perception and higher-level knowledge, a crucial factor in the process of human recognition and categorization [9, 10].

1.2 The Structure of SIMBAD

SIMBAD represented the first systematic attempt towards the goal alluded to above. Within the project, we undertook a thorough study of several aspects of similarity-based pattern analysis and recognition methods, from the theoretical, algorithmic, and applicative perspective, with a view to substantially advance the state of the art in the field and contribute towards the long-term goal of organizing this emerging field into a more coherent whole.

We focused on two main themes, which basically correspond to the two fundamental questions that arise when abandoning the realm of feature-vector representations, namely:

1. How can one *obtain* suitable similarity information from object representations that are more powerful than, or simply different from, the vectorial?
2. How can one *use* similarity information in order to perform learning and classification tasks?

Although the two issues are clearly interrelated, it is advantageous to keep them apart as this allows one to separate the similarity generation process (a data modeling issue) from the learning and classification processes (a task modeling issue). According to this perspective, the very notion of similarity becomes the pivot of non-vectorial pattern recognition in much the same way as the notion of feature-vector plays the role of the pivot in the classical (geometric) paradigm. This results in a useful modularity, which means that all interactions between the object representation and the learning algorithm are mediated by the similarities, which is where the domain knowledge comes into the scene.

An important part of the project concerned the application of the developed techniques. To this end, we focused mainly on biomedical problems, which lend themselves particularly well to similarity-based approaches. Specifically, we applied the new methods developed within the project to inference tasks in the field of medical image analysis, i.e., to Tissue Micro Array (TMA) analysis and to Magnetic Resonance (MR) brain imaging.

Accordingly, the project (and hence this book) was structured around the following strands:

- Foundational issues
- Deriving similarities for non-vectorial data
- Embedding and beyond
- Applications

which we now briefly describe.

1.2.1 Foundational Issues

One of the first objectives within SIMBAD was to explore the causes and origins of non-Euclidean (dis)similarity measures and how they influence the performance of classical classification algorithms. In particular, we distinguished between the situation where the informational content associated with the violation of the geometric properties is limited, or is simply an artifact of the measurement process, and that where this is not the case. This distinction is important as, depending on the actual situation, two different strategies can be pursued: the first attempts to impose geometricity by somehow transforming or re-interpreting the similarity data, the second does not and works directly on the original similarities. Chapter 2 provides a comprehensive summary of our findings. It also discusses several techniques to convert non-Euclidean data into Euclidean and provides real-world examples which show that the non-geometric part of the data might be essential for building good classifiers.

A second line of investigation within this strand concerned fundamental questions pertaining to the very nature of the pattern recognition endeavor. Indeed, the search for patterns in data requires a mathematical definition of structure and a comparison function to rank different structures, thereby providing insights into the invariances in the problem class at hand. Motivated by an analogy between communication and learning, Chap. 3 describes an information-theoretic perspective to the problem and attempts to address the question of model selection and validation or, in other words, the tradeoff between informativeness and robustness. According to the proposed view, the notion of a pattern is interpreted as an element of an interpretation space (the “hypothesis class”) endowed with a “natural” neighborhood system, or topology. By generalizing Shannon’s random coding concept, the framework is able to determine which hypotheses are statistically indistinguishable due to measurement noise and how much we have to coarsen the hypothesis

class. The framework is thought to be applicable to more general questions arising in computer science concerning algorithm evaluation as well as (robust) algorithm design.

1.2.2 Deriving Similarities for Non-vectorial Data

The goal here was to develop suitable similarity measures for non-vectorial data. We focused primarily on structured data (e.g., strings, graphs, etc.), because of their expressive power and ubiquity, and on geometric measures as they allow one to employ the whole arsenal of powerful techniques available in the geometric pattern recognition literature. We pursued our goal by developing suitable kernels, which are known to be in correspondence with geometric (dis)similarities and considered in particular information-theoretic kernels. These are based on the assumption that the objects of interest are generated by some probabilistic mechanism (a source, in information/coding theoretic terms) and then proceed by defining (dis)similarity measures or kernels between (or among) models of these probabilistic sources. Chapter 4 reviews a recent approach which exploits the probabilistic nature of the so-called generative embeddings, by using information-theoretic kernels defined on probability distributions. This leads to a new class of hybrid generative/discriminative methods for learning classifiers whose effectiveness has been tested on two medical applications (see also Chaps. 9 and 10).

An alternative to this “kernel tailoring” approach consists in learning good similarities directly from training data. Within SIMBAD we investigated a strategy based on the evidence accumulation clustering paradigm, which aims to combine the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of data organization. Chapter 5 describes an approach which exploits the duality of similarity-based and probabilistic interpretations of the learned co-association matrix in order to produce robust and informative consensus solutions. This leads to two clustering methods: a “hard” method which explores embeddings over learned pairwise associations, and a unified probabilistic approach that we called PEACE (Probabilistic Evidence Accumulation for Clustering Ensembles), leading to soft assignments of objects to clusters.

1.2.3 Embedding and Beyond

Within this research strand, we aimed at developing computational models that do not depend on the actual object representation and rely only on (available) similarity information. As pointed out above, the analysis carried out in Chap. 2 suggests two complementary approaches. On the one hand, when the information content of non-geometricity is limited or simply caused by measurement errors, it is a plausible strategy to perform some correction on the similarity data (or finding an alternative

vectorial representation) in an attempt to impose geometricity, and then use conventional geometric techniques. On the other hand, when the information content of non-geometricity is relevant, one needs brand new tools, as standard approaches would not work in this case.

The former approach is known as “embedding,” which is a well-established technique for vector-based representations, and is the subject of Chaps. 6 and 7. In particular, Chap. 6 focuses on two contrasting approaches to the problem. In the first part, it describes spectral methods for embedding structured data such as weighted graphs in a geometrically meaningful way. The resulting embeddings are then used to construct generative models for graph structure. To this end, the chapter explores the idea of “spherical” embedding, whereby data is embedded onto the surface of sphere of optimal radius. Instead of approximating the original (dis)similarities by Euclidean distances, the second approach tries to preserve the underlying group structure of the data. Within this context, the second part of Chap. 6 shows that a polynomial characterization derived from the Ihara zeta function leads to an embedding of hypergraphs which captures interesting structural properties.

Chapter 7 also focuses on these “structure-preserving” embeddings and restricts the discussion to the case of partition-based clustering problems. It is shown that a classical pairwise clustering cost function possesses an interesting shift-invariance property which amounts to saying that the choice of a partition is not influenced by additive constant shifts in the off-diagonal elements of the affinity matrix. An approximate version of this property is shown to hold in a more general probabilistic setting which is capable of selecting the number of clusters in a data-adaptive way. These findings raise intriguing questions concerning the role of structure-preserving embedding in the context of a theory of similarity-based pattern recognition.

When there is significant information content in the non-(geo)metricity of the data one has to resort to algorithms that work directly on the original similarity function. To this end, Chap. 8 describes an approach based on game theory which is shown to offer an elegant and powerful conceptual framework that serves well our purpose. The main point made by game theorists is to shift the emphasis from optimality criteria to equilibrium conditions, namely to the search of a balance among multiple interacting forces. Interestingly, the development of evolutionary game theory in the late 1970s offered a dynamical systems perspective, an element which was totally missing in the traditional formulation. From our perspective, one of the main attractive features of game theory is that it imposes no restriction whatsoever on the structure of the similarity function. Chapter 8 describes our attempts at formulating classical pattern recognition problems from a purely game-theoretic perspective. In particular, the chapter focuses on data clustering and structural matching and discusses some successful computer vision applications.

1.2.4 Applications

Pattern recognition and machine learning are essentially application-oriented fields with well-established validation techniques. These were used to quantitatively eval-

uate the success of the proposed research on large-scale applications with clear societal impact. In particular, within SIMBAD we devoted substantial effort towards tackling two large-scale biomedical imaging applications. With the direct involvement of leading pathologists and neuroscientists from the University Hospital Zurich and the Verona–Udine Brain Imaging and Neuropsychology Program, we contributed towards the concrete objective of providing effective, advanced techniques to assist in the diagnosis of renal cell carcinoma, one of the ten most frequent malignancies in Western countries, as well as of major psychoses such as schizophrenia and bipolar disorders. The results of our research are summarized in Chaps. 9 and 10, respectively. These problems are not amenable to be tackled with traditional machine learning techniques due to the difficulty of deriving suitable feature-based descriptions. For instance, image segmentation and shape alignment problems often produce non-(geo)metric dissimilarity data in both application domains, a feature which is indeed present in many other biomedical problems.

1.3 Conclusion and Outlook

There is an increasing awareness of the importance of similarity-based approaches to pattern recognition and machine learning, and research in this area has gone past the proof-of-concept phase and is now spreading rapidly. In fact, traditional feature-based techniques are felt as inherently unable to deal satisfactorily with the complexity and richness arising in many real-world situations, thereby hindering the application of machine learning techniques to a whole variety of relevant, real-world problems. Hence, in general, progress in similarity-based approaches will surely be beneficial for machine learning as a whole and, consequently, for the long-term enterprise of building intelligent systems.

We do believe that SIMBAD has contributed substantially towards the advancement of the state of the art in this area. In fact, we have introduced fresh perspectives to old problems, we have provided a thorough analysis of foundational issues, and we have demonstrated the applicability of our methodologies in real-world applications. In conclusion, we went far beyond our original expectations. Of course, we think there is room for improvement. In this respect, it might probably be useful to involve people from “external” fields such as cognitive psychology and/or algorithmics, thereby making the research more interdisciplinary. Also, as a matter of future work, there are promising application areas, such as chemometrics, bioinformatics, social network analysis, etc., which would certainly benefit from the work done within the project. We do hope that the availability into a single coherent book of the main results achieved within SIMBAD will foster further progress in this important emerging field.

References

1. Altschul, S.F., Gish, W., Miller, W., Meyers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990)

2. Balcan, M.F., Blum, A., Srebro, N.: A theory of learning with similarity functions. *Mach. Learn.* **72**(1–2), 89–112 (2008)
3. Biederman, I.: Recognition-by-components: a theory of human image understanding. *Psychol. Rev.* **94**, 115–147 (1987)
4. Bridgman, P.W.: *The Logic of Modern Physics*. MacMillan, New York (1927)
5. Bunke, H., Sanfeliu, A.: *Syntactic and Structural Pattern Recognition: Theory and Applications*. World Scientific, Singapore (1990)
6. Dubuisson, M.P., Jain, A.K.: Modified Hausdorff distance for object matching. In: *Proc. Int. Conf. Pattern Recognition (ICPR)*, pp. 566–568 (1994)
7. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. Wiley, New York (2000)
8. Easley, D., Kleinberg, J.: *Networks, Crowds, and Markets*. Cambridge University Press, Cambridge (2010)
9. Edelman, S.: *Representation and Recognition in Vision*. MIT Press, Cambridge (1999)
10. Goldstone, R.L., Son, J.Y.S.: In: Holyoak, K., Morrison, R. (eds.) *The Cambridge Handbook of Thinking and Reasoning*, pp. 13–36. Cambridge University Press, Cambridge (2005)
11. Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances: Image retrieval and class representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 583–600 (2000)
12. Kleinberg, J.: Authoritative sources in a hyperlink environment. In: *Proc. 9th ACM/IEEE Symposium on Discrete Algorithms*, pp. 668–677 (1998)
13. Lakoff, G.: *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. University of Chicago Press, Chicago (1987)
14. Mayr, E.: *The Growth of Biological Thought*. Harvard University Press, Cambridge (1982)
15. Popper, K.R.: *Conjectures and Refutations: the Growth of Scientific Knowledge*. Routledge, London (1963)
16. Resnik, M.D.: *Mathematics as a Science of Patterns*. Clarendon, Oxford (1997)
17. Rorty, R.: A world without substances and essences. In: *Philosophy and Social Hope*, pp. 47–71. Penguin, London (1999)
18. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. *AI Mag.* **29**(3), 93–106 (2008)
19. von Luxburg, U., Williamson, R.C., Guyon, I.: Clustering: Science or art? In: *JMLR: Workshop and Conference Proceedings*, vol. 27, pp. 65–79 (2012)
20. Watanabe, S.: *Pattern Recognition: Human and Mechanical*. Wiley, New York (1985)

Part I
Foundational Issues

Chapter 2

Non-Euclidean Dissimilarities: Causes, Embedding and Informativeness

Robert P.W. Duin, Elzbieta Pełalska, and Marco Loog

Abstract In many pattern recognition applications, object structure is essential for the discrimination purpose. In such cases, researchers often use recognition schemes based on template matching which lead to the design of non-Euclidean dissimilarity measures. A vector space derived from the embedding of the dissimilarities is desirable in order to use general classifiers. An isometric embedding of the symmetric non-Euclidean dissimilarities results in a pseudo-Euclidean space. More and better tools are available for the Euclidean spaces but they are not fully consistent with the given dissimilarities.

In this chapter, first a review is given of the various embedding procedures for the pairwise dissimilarity data. Next the causes are analyzed for the existence of non-Euclidean dissimilarity measures. Various ways are discussed in which the measures are converted into Euclidean ones. The purpose is to investigate whether the original non-Euclidean measures are informative or not. A positive conclusion is derived as examples can be constructed and found in real data for which the non-Euclidean characteristics of the data are essential for building good classifiers. (This chapter is based on previous publications by the authors, (Duin and Pełalska in Proc. SSPR & SPR 2010 (LNCS), pp. 324–333, 2010 and in CIARP (LNCS), pp. 1–24, 2011; Duin in ICEIS, pp. 15–28, 2010 and in ICPR, pp. 1–4, 2008; Duin et al. in SSPR/SPR, pp. 551–561, 2008; Pełalska and Duin in IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 38(6):729–744, 2008) and contains text, figures, equations, and experimental results taken from these papers.)

R.P.W. Duin · M. Loog (✉)

Pattern Recognition Laboratory, Delft University of Technology, Delft, The Netherlands

e-mail: m.loog@tudelft.nl

R.P.W. Duin

e-mail: r.duin@ieee.org

E. Pełalska

Manchester, UK

e-mail: ela@elapekalska.com

url: <http://www.elapekalska.com>

2.1 Introduction

Automatic recognition systems work with objects such as images, videos, time signals, spectra, and so on. They are built in the process of learning from a set of object examples labeled with the desired pattern classes. Two main steps can be distinguished in this procedure:

Representation: Individual objects are characterized by a set of suitable mathematical descriptors such as vectors, strings of symbols or graphs. A good representation is the one in which objects can easily be related to each other in order to facilitate the next step.

Generalization/Discrimination: The representations of the object examples should enable the mathematical modeling of object classes or class discriminants such that a good class estimate can be found for new, unseen and, thereby, unlabeled objects using the same representation.

The most popular representations, next to strings and graphs, encodes objects as vectors in Euclidean vector spaces. Instead of single vectors, also sets of vectors may be considered for representing individual objects, as studied, e.g., in [32, 33, 46, 48]. For some applications, representations defined by strings of symbols and attributed graphs are preferred over vectors as they model the objects more accurately and offer more possibilities to include domain expert knowledge [6].

On the other hand, representations in Euclidean vector spaces are well suited for generalization. Many tools are available to build (learn) models and discriminant functions from sets of object examples (also called training sets) that may be used to classify new objects into the right class. Traditionally, the Euclidean vector space is defined by a set of features. These should ideally characterize the patterns well and be relevant for class differences at the same time. Such features have to be defined by experts exploiting their knowledge of the application.

The use of features has one important drawback. Features often represent the objects just partially because they encode their limited characteristics. Consequently, different objects may have the same representation, i.e., the same feature vector, when they differ by properties that are not expressed in the chosen feature set. This results in class overlap: in some areas of the feature space, objects of different classes are represented by the same feature vectors. Consequently, they cannot be distinguished any longer, which leads to an intrinsic classification error, usually called the Bayes error.

An alternative to the feature representation is the dissimilarity representation defined on direct pairwise object comparisons. If the entire objects are taken into account in the comparison, then only identical objects will have a dissimilarity zero (if the dissimilarity measure has the property of ‘identity of indiscernibles’). For such a representation class, overlap does not exist if the objects are unambiguously labeled, which means that there are no real world objects in the application that belong to multiple classes.

Some dissimilarity measures used in practice do not have the property that a zero dissimilarity can only arise for identical objects. An example is the single-linkage distance used in clustering: the dissimilarity between two clusters is defined

as the distance between the two most neighboring vectors. This distance measure corresponds to defining the smallest distance between the surfaces of two real world objects as the distance between the objects. A zero value, however, does not imply that the objects are identical; they are just touching.

Distance measures such as the above, and many others, cannot be perfectly embedded in a Euclidean space. This means that there is no set of vectors in a vector space of any dimensionality for which the Euclidean distances between the objects are identical to the given ones. In particular, it holds for non-metric distances, which are just an example from a large set of non-Euclidean distance measures. As we want to include non-metric distances (such as the single-linkage distance) we will use the more general term of dissimilarities instead of distances. They refer to possibly improper distance measures in the mathematical sense. We will still assume that dissimilarities are non-negative and that they have a monotonic relation with object differences: if two given objects are made more different, their dissimilarity increases.

Non-Euclidean symmetric dissimilarity data can be perfectly embedded into pseudo-Euclidean spaces. A proper embedding of non-Euclidean dissimilarities and the training of classifiers in the resulting space are, however, not straightforward. There are computational as well as fundamental problems to be solved. The question thereby arises whether the use of non-Euclidean dissimilarity measures is strictly necessary. Finding the causes of such measures, see Sect. 2.2, is a first step to answer this question. This will be more extensively discussed in Sect. 2.6. We will investigate whether such measures are really informative and whether it is possible to make Euclidean corrections or approximations by which no information is lost.

Two main vectorial representations of the dissimilarity data, the dissimilarity space and the pseudo-Euclidean embedded space, are presented in Sect. 2.3. Section 2.4 discusses classifiers which can be trained in such spaces. Transformations which make the dissimilarity data Euclidean are briefly presented in Sect. 2.5. Next, numerous examples of artificial and real dissimilarity data are collected in Sect. 2.7. Oftentimes, they illustrate that linear classifiers in the dissimilarity-derived vector spaces are much more advantageous than the traditional 1-NN rule. Finally, we summarize and discuss our findings in Sect. 2.8.

The issue of informativeness of the non-Euclidean measures is the main topic of this chapter. We will present artificial and real world examples for which the use of such measures is really informative. We will, however, also make clear that for any given classifier defined in a non-Euclidean space an equivalent classifier in a Euclidean space can be constructed. It is a challenge to do this such that the training of good classifiers in this Euclidean space is feasible. In addition, we will argue that the dissimilarity space as proposed by the authors [37, 55] is a Euclidean space that preserves all non-Euclidean information and enables the design of well performing classifiers.

2.2 Causes of Non-Euclidean Dissimilarities

In this section, we shortly explain why non-Euclidean dissimilarities frequently arise in the applications. This results from the analysis of a set of real world objects. Let D be an $N \times N$ dissimilarity matrix describing a set of pairwise dissimilarities between N objects. D is Euclidean if it can be perfectly embedded into a Euclidean space. This means that there exists a Euclidean vector space with N vectors for which all Euclidean distances are identical to the given ones.

There are N^2 free parameters if we want to position N vectors in an N -dimensional space. The dissimilarity matrix D has also N^2 values. D should be symmetric because the Euclidean distance is. Still, there might be no solution possible as the relation between vector coordinates and Euclidean distances is nonlinear. More on the embedding procedures is discussed in Sect. 2.3. At this moment, we need to remember that the matrix D is Euclidean only if the corresponding vector space exists.

First, it should be emphasized how common non-Euclidean measures are. An extensive overview of such measures is given in [55], but we have often encountered that this fact is not fully recognized. Most researchers wrongly assume that non-Euclidean distances are equivalent to non-metric ones. There are, however, many metric but non-Euclidean distances, such as the city-block or ℓ_1 -norm.

Almost all probabilistic distance measures are non-Euclidean by nature. This implies that by dealing with object invariants, the dissimilarity matrix derived from the overlap between the probability density functions corresponding to the given objects is non-Euclidean. Also the Mahalanobis class distance as well as the related Fisher criterion is non-Euclidean. Consequently, many non-Euclidean distance measures are used in cluster analysis and in the analysis of spectra in chemometrics and hyperspectral image analysis as spectra can be considered as one-dimensional distributions.

Secondly, what is often overlooked is the following fact. One may compare pairs of real world objects by a (weighted) Euclidean distance, yet the complete set of N objects giving rise to an $N \times N$ dissimilarity matrix D is non-Euclidean. In short, this is caused by the fact that different parts or characteristics of objects are used per pair to define the object differences. Even if the dissimilarity is defined by the weighted sum of differences, as long as there is no single basis of reference for the comparison of *all pairs*, the resulting dissimilarity matrix D will be non-Euclidean. These types of measures often result from matching procedures which minimize the cost or path of transformation between two objects. Fundamental aspects of this important issue are extensively discussed in Sect. 2.2.2.3.

In shape recognition, various dissimilarity measures are based on the weighted edit distance, on variants of the Hausdorff distance or on nonlinear morphing. Usual parameters are optimized within an application w.r.t. the performance based on template matching and other nearest neighbor classifiers [14]. Almost all have non-Euclidean behavior and some are even non-metric [14].

In the design and optimization of the dissimilarity measures for template matching, their Euclidean behavior is not an issue. With the popularity of support vector

machines (SVMs), it has become important to design kernels (similarities) which fulfill the Mercer conditions [12]. This is equivalent to a possibility of an isometric Euclidean embedding of such a kernel (or dissimilarities). Next sections discuss reasons that give rise to violations of these conditions leading to non-Euclidean dissimilarities or indefinite kernels.

2.2.1 Non-intrinsic Non-Euclidean Dissimilarities

Below we identify some non-intrinsic causes that give rise to non-Euclidean dissimilarities. In such cases, it is not the dissimilarity measure itself, but the way it is computed or applied that causes the non-Euclidean behavior.

2.2.1.1 Numeric Inaccuracies

Non-Euclidean dissimilarities arise due to the numeric inaccuracies caused by the use of a finite word length. If the intrinsic dimensionality of the data is lower than the sample size, the embedding procedure that relies on an eigendecomposition of a certain matrix, see Sect. 2.3, may lead to numerous tiny negative eigenvalues. They should be zero in fact, but become nonzero due to numerical problems. It is thereby advisable to neglect dimensions (features) that correspond to very small positive and negative eigenvalues.

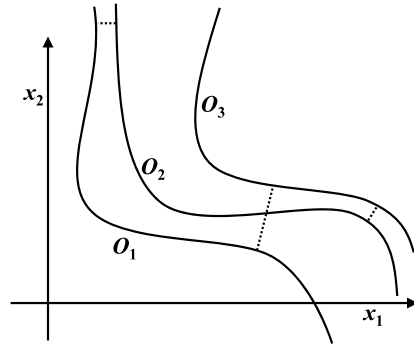
2.2.1.2 Overestimation of Large Distances

Complicated measures are used when dissimilarities are derived from raw data such as (objects in) images. They may define the distance between two objects as the length of the path that transforms one object into the other. Examples are the weighted edit distance [4] and deformable templates [31]. In the optimization procedure that minimizes the path length, the procedure may approximate the transformation costs from above. As a consequence, too large distances are found. Even if the objects are compared by a (weighted) Euclidean distance measure, the resulting set of dissimilarities in D will often become non-Euclidean or even non-metric.

2.2.1.3 Underestimation of Small Distances

The underestimation of small distances has the same result as the overestimation of large distances. It may happen when the pairwise comparison of objects is based on different properties for each pair, as it is the case, e.g., in studies on consumer preference data. Another example is the comparison of partially occluded objects in computer vision.

Fig. 2.1 Vector space with the invariant trajectories for three objects O_1 , O_2 and O_3 . If the chosen dissimilarity measure is defined as the minimum distance between these trajectories, triangle inequality can easily be violated, i.e., $d(O_1, O_2) + d(O_1, O_3) < d(O_2, O_3)$



2.2.2 Intrinsic Non-Euclidean Dissimilarities

The causes discussed in the above may be judged as accidental. They result either from computational or observational problems. If better computers and observations were available, they would disappear. Now, we will focus on dissimilarity measures for which this will not happen. There are three possibilities.

2.2.2.1 Non-Euclidean Dissimilarities

As already indicated at the start of this section, arguments can be given from the application side to use another metric than the Euclidean one. An example is the l_1 -distance between energy spectra as it is related to energy differences. Although the l_2 -norm is very convenient for computational reasons and it is rotation invariant in a Euclidean space, other distance measures may naturally arise from the demands in applications, e.g., see [47].

2.2.2.2 Invariants

A fundamental reason behind non-Euclidean dissimilarities is related to the occurrence of invariants. Frequently, one is not interested in the dissimilarity between given objects A and B , but in the dissimilarity between their equivalence classes, i.e., sets of objects $A(\theta)$ and $B(\theta)$ in which θ controls an invariant. One may define the dissimilarity between the A and B as the minimum difference between the sets defined by all their invariants (see Fig. 2.1 for an illustration of this idea):

$$d^*(A, B) = \min_{\theta_A} \min_{\theta_B} (d(A(\theta_A), B(\theta_B))). \quad (2.1)$$

This measure is non-metric: the triangle inequality may be violated as for different pairs of objects different values of θ are found minimizing (2.1).

2.2.2.3 Sets of Vectors

Complicated objects such as multi-region images may be represented by sets of vectors. Problems like this are investigated in the domain of Multi Instance Learning (MIL) [13], or Bag-of-Words (BoW) classification [52]. Distance measures between such sets have already been studied for a long time in cluster analysis. Many are non-Euclidean or even non-metric, such as the single linkage distance. This measure is defined as the distance between the two most neighboring points of the two clusters being compared. It is non-metric. It even holds that if $d(A, B) = 0$, then it does not follow that $A \equiv B$.

For the single linkage dissimilarity measure it can be understood why the dissimilarity space may be useful. Given a set of such dissimilarities between clouds of vectors, it can be concluded that two clouds are similar if the two sets of dissimilarities with all other clouds are about equal. If just their mutual dissimilarity is (close to) zero, they may still be very different.

The problem with the single linkage dissimilarity measure between two sets of vectors points to a more general problem in relating sets and even objects. In [33], an attempt has been made to define a proper Mercer kernel between two sets of vectors. Such sets are in that paper compared by the Hellinger distance derived from the Bhattacharyya's affinity between two pdfs $p_A(x)$ and $p_B(x)$ found for the two vector sets A and B :

$$d(A, B) = \left[\int (\sqrt{p_A(x)} - \sqrt{p_B(x)})^2 \right]^{1/2}. \quad (2.2)$$

The authors state that by expressing $p(x)$ in any orthogonal basis of functions, the resulting kernel K is automatically positive semidefinite (psd). This is only correct, however, if all vector sets A, B, \dots to which the kernel is applied have the same basis. If different bases are derived in a pairwise comparison of sets, the kernel may become indefinite. This occurs if the two pdfs are estimated in a subspace defined by a PCA computed from the objects of the two classes A and B only.

This makes clear that indefinite relations may arise in any pairwise comparison of real world objects if every pair of objects is first represented in some joint space in which the dissimilarity is computed. These joint spaces may be different for different pairs! Consequently, the total set of dissimilarities will likely have a non-Euclidean behavior, even if each comparison relies on the Euclidean distance, as in (2.2).

The consequence of this observation is huge for pattern recognition applications. It implies that a representation defined by pairwise dissimilarities between objects can only be Euclidean if a common basis between all objects, including the future test objects, is found for the derivation of such dissimilarities. This is naturally, by definition, the case for feature vector representations, as the joint space for all objects is already defined by the chosen set of features. For the dissimilarity representation, however, which has the advantage of potentially using the entire objects,

the consequence is that no common representation basis can be found before all objects are seen. This contradicts the idea of generalization and discrimination: being able to classify unseen objects.

We emphasize this conclusion as we judge it as very significant: Non-Euclidean object relations naturally arise for real world object recognition as no Euclidean representation can be defined before we have seen (or implicitly considered) all objects, including the ones to be recognized in future. Transductive inference [49] is the solution: include the objects to be classified in the definition of the representation.

2.3 Vector Spaces for the Dissimilarity Representation

The complete dissimilarity representation is defined as a square matrix with the dissimilarities between all pairs of objects. Traditionally, in the nearest neighbor classification scenario, just the dissimilarities between the test objects and training objects are used. For every test object, the nearest neighbors in the set of training objects are first found and used by the nearest neighbor rule. This procedure does not make use of the pairwise relations between the training objects.

The following two approaches construct a new vector space on the basis of the relations within the training set. The resulting vector space is used for training classifiers.

In the first approach, the dissimilarity matrix is considered as a set of vectors, one for every object. They represent the objects in a vector space constructed by the dissimilarity vectors whose coordinates are dissimilarities to the training objects. Usually, this vector space is treated as a Euclidean space and equipped with the standard inner product definition.

In the second approach, an attempt is made to embed the dissimilarity matrix in a Euclidean vector space such that the distances between the extracted vectors are equal to the given dissimilarities. This can only be realized without error, of course, if the original set of dissimilarities is Euclidean. If this is not the case, either an approximate procedure has to be followed or the objects should be embedded into a non-Euclidean vector space. This is a space in which the standard inner product definition and the related distance measure are changed, leading to indefinite inner products and later to indefinite kernels.

It appears that an exact embedding is possible for every symmetric $N \times N$ dissimilarity matrix D with zero self-dissimilarity, i.e., a diagonal all of zeros. The resulting space is the so-called pseudo-Euclidean space.

These two approaches are more formally defined below, using an already published description [20].

2.3.1 Dissimilarity Space

Let $\mathcal{X} = \{x_1, \dots, x_n\}$ be a training set. Given a dissimilarity function and/or dissimilarity data, we define a data-dependent mapping $D(\cdot, R) : \mathcal{X} \rightarrow \mathbb{R}^k$ from \mathcal{X} to

the so-called *dissimilarity space* (DS) [19, 26, 43]. The k -element set R consists of objects that are representative for the problem. This set is called the representation set or prototype set and it may be a subset of \mathcal{X} . In the dissimilarity space, each dimension $D(\cdot, p_i)$ describes a dissimilarity to a prototype p_i from R .

We initially choose $R := \mathcal{X}$. As a result, every object is described by an n -dimensional dissimilarity vector $D(x, \mathcal{X}) = [d(x, x_1) \cdots d(x, x_n)]^T$. The resulting vector space is endowed with the traditional inner product and the Euclidean metric.

Any dissimilarity measure ρ can be defined in the Dissimilarity Space. One of them is the Euclidean distance:

$$\rho_{\text{DS}}(x, y) = \left(\sum_{i=1}^n [d(x, x_i) - d(y, x_i)]^2 \right)^{1/2} \quad (2.3)$$

This is the distance computed on dissimilarity vectors defined by original dissimilarities. For metric dissimilarity measures ρ , it holds asymptotically that the nearest neighbor objects are unchanged by ρ_{DS} . This is, however, not necessarily true for finite data sets. In that case, the nearest neighbors in dissimilarity space might be more appropriate for classifications as the distances are defined in the context of the entire representation set.

The approaches discussed here are originally intended for dissimilarities directly computed between objects and not resulting from feature representations. It is, however, still possible to study dissimilarity representations derived from features which may yield interesting results [40]. In Fig. 2.2, an example is presented that compares an optimized radial basis SVM with a Fisher linear discriminant computed in the dissimilarity space derived from the Euclidean distances in a feature space. The example shows a large variability of the nearest neighbor distances. As the radial basis kernel used by SVM is constant it cannot be optimal for all regions of the feature space.

The Fisher linear discriminant is computed in the complete dissimilarity space, where the classes are linearly separable. Although the classifier is overtrained (the dissimilarity space is 100-dimensional and the training set has also 100 objects) it gives here the perfect result. It should be realized that this example is specifically constructed to show the possibilities of the dissimilarity space.

2.3.2 Pseudo-Euclidean Space

Before explaining the relation between pseudo-Euclidean spaces and dissimilarity representation, we start with definitions.

A Pseudo-Euclidean Space (PES) $\mathcal{E} = \mathbb{R}^{(p,q)} = \mathbb{R}^p \oplus \mathbb{R}^q$ is a vector space with a non-degenerate indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ such that $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ is positive definite on \mathbb{R}^p and negative definite on \mathbb{R}^q [24, 55]. The inner product in $\mathbb{R}^{(p,q)}$ is defined (wrt an orthonormal basis) as $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{E}} = \mathbf{x}^T \mathcal{J}_{pq} \mathbf{y}$, where $\mathcal{J}_{pq} = [I_{p \times p} \ 0; 0 \ -I_{q \times q}]$

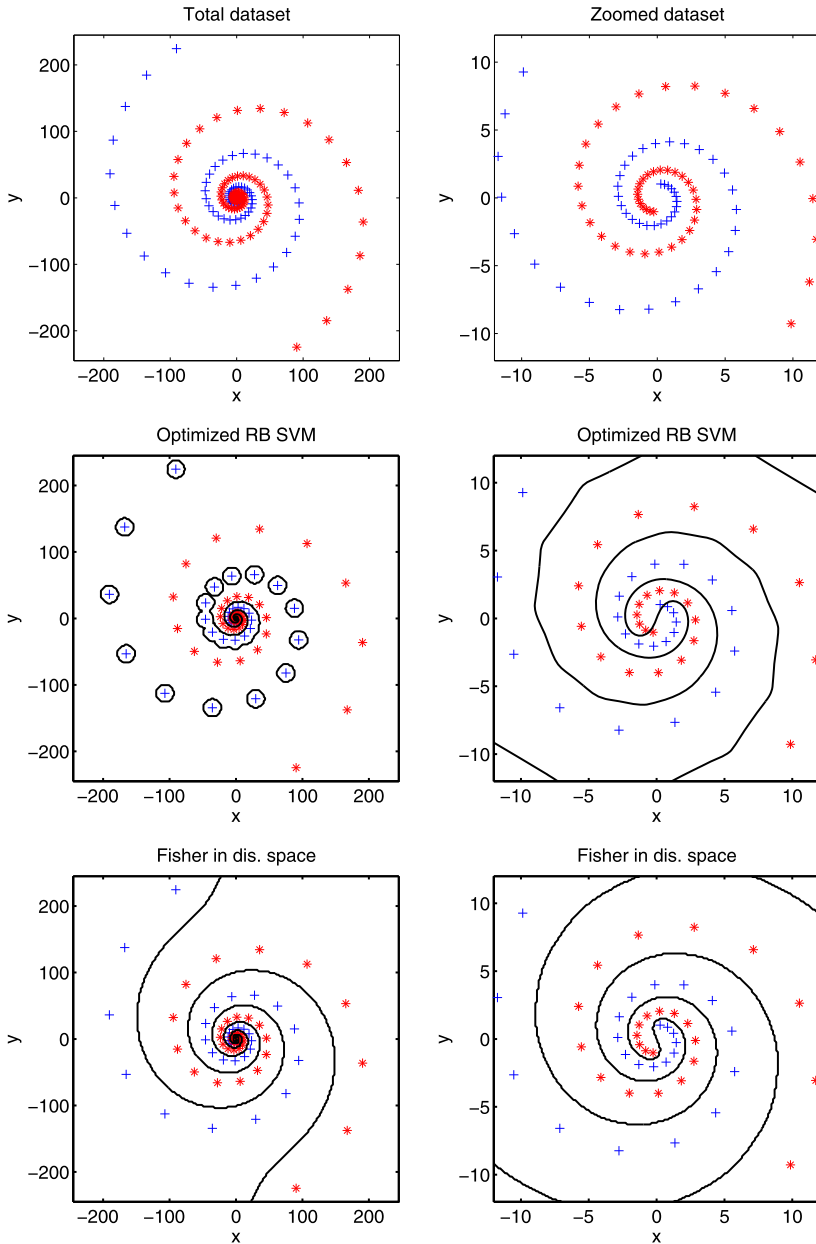


Fig. 2.2 A spiral example with 100 objects per class. *Left column* shows the complete data sets, while the *right column* presents the zoom of the spiral center. 50 objects per class, systematically sampled, are used for training. The *middle row* shows the training set and SVM with an optimized radial basis function; 17 out of 100 test objects are erroneously classified. The *bottom row* shows the Fisher Linear Discriminant (without regularization) computed in the dissimilarity space derived from the Euclidean distances. All test objects are correctly classified

and I is the identity matrix. As a result, $d_{\mathcal{E}}^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathcal{J}_{pq}(\mathbf{x} - \mathbf{y})$. Obviously, a Euclidean space \mathbb{R}^p is a special case of a pseudo-Euclidean space $\mathbb{R}^{(p,0)}$. An infinite-dimensional extension of a PES is a Kreĭn space. It is a vector space \mathcal{K} equipped with an indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}} : \mathcal{K} \times \mathcal{K} \rightarrow \mathbb{R}$ such that \mathcal{K} admits an orthogonal decomposition as a direct sum, $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$, where $(\mathcal{K}_+, \langle \cdot, \cdot \rangle_+)$ and $(\mathcal{K}_-, -\langle \cdot, \cdot \rangle_-)$ are separable Hilbert spaces with their corresponding positive and negative definite inner products.

A positive definite kernel function can be interpreted as a generalized inner product in some Hilbert space. This space becomes Euclidean when a kernel matrix is considered. In analogy, an arbitrary symmetric kernel matrix can be interpreted as a generalized inner product in a pseudo-Euclidean space. Such a PES is obviously data dependent and can be retrieved via an embedding procedure. Similarly, an arbitrary symmetric dissimilarity matrix with zero self-dissimilarities can be interpreted as a pseudo-Euclidean distance in a proper pseudo-Euclidean space.

Since in practice we deal with finite data, dissimilarity matrices or kernel matrices can be seen as describing relations between vectors in the underlying pseudo-Euclidean spaces. These pseudo-Euclidean spaces can be either determined via an embedding procedure and directly used for generalization, or approached indirectly by the operations on the given indefinite kernel. The section below explains how to find the embedded PES.

2.3.2.1 Pseudo-Euclidean Embedded Space

A symmetric dissimilarity matrix $D := D(\mathcal{X}, \mathcal{X})$ can be embedded in a Pseudo-Euclidean Space (PES) \mathcal{E} by an isometric mapping [24, 55]. The embedding relies on the indefinite Gram matrix G , derived as $G := -\frac{1}{2}HD^*2H$, where $D^{*2} = (d_{ij}^2)$ and $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. H projects the data such that X has a zero mean vector. The eigendecomposition of G leads to $G = Q\Lambda Q^T = Q|\Lambda|^{\frac{1}{2}}[\mathcal{J}_{pq}; 0]|\Lambda|^{\frac{1}{2}}Q^T$, where Λ is a diagonal matrix of eigenvalues, first decreasing p positive ones, then increasing q negative ones, followed by zeros. Q is the matrix of eigenvectors. Since $G = X\mathcal{J}_{pq}X^T$ by definition of a Gram matrix, $X \in \mathbb{R}^n$ is found as $X = Q_n|\Lambda_n|^{\frac{1}{2}}$, where Q_n consists of n eigenvectors ranked according to their eigenvalues Λ_n . Note that X has a zero mean and is uncorrelated, because the estimated pseudo-Euclidean covariance matrix $C = \frac{1}{n-1}X^T X \mathcal{J}_{pq} = \frac{1}{n-1}\Lambda_r$ is diagonal. The eigenvalues λ_i encode variances of the extracted features in $\mathbb{R}^{(p,q)}$.

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. If this space is a PES $\mathbb{R}^{(p,q)}$, $p + q = n$, the pseudo-Euclidean distance is computed as:

$$\begin{aligned} \rho_{\text{PES}}(\mathbf{x}, \mathbf{y}) &= \left(\sum_{i=1}^p [x_i - y_i]^2 - \sum_{i=p+1}^{p+q} [x_i - y_i]^2 \right)^{1/2} \\ &= \left(\sum_{i=1}^n \delta(i, p) [x_i - y_i]^2 \right)^{1/2}, \end{aligned}$$

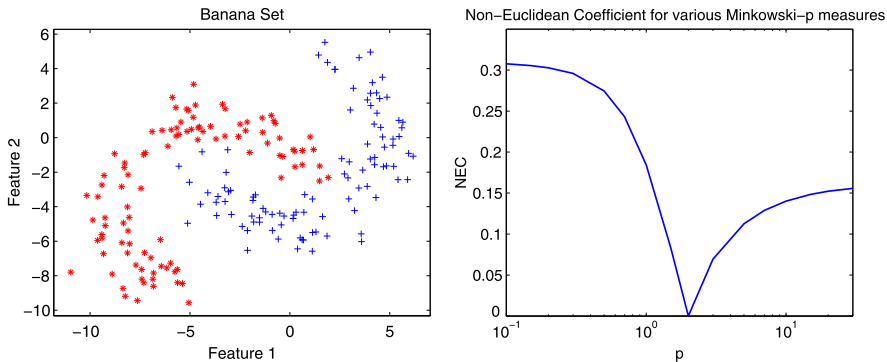


Fig. 2.3 A two-dimensional data set (*left*) and the NEF as a function of p for various Minkowski- p dissimilarity measures

where $\delta(i, p) = \text{sign}(p - i + 0.5)$. Since the complete pseudo-Euclidean embedding is perfect, $D(x, y) = \rho_{\text{PES}}(x, y)$ holds.

Other distance measures may also be defined between vectors in a PES, depending on how this space is interpreted. Two obvious choices are:

$$\rho_{\text{PES}^+}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^p [x_i - y_i]^2 \right)^{1/2}, \quad (2.4)$$

which neglects the dimensions corresponding to the negative contributions (derived from negative eigenvalues in the embedding), and

$$\rho_{\text{AES}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n [x_i - y_i]^2 \right)^{1/2}, \quad (2.5)$$

which treats the vector space \mathbb{R}^n as Euclidean \mathbb{R}^{p+q} . This means that the negative subspace of PES is interpreted as a Euclidean subspace (i.e., the negative signs of eigenvalues are neglected in the embedding procedure).

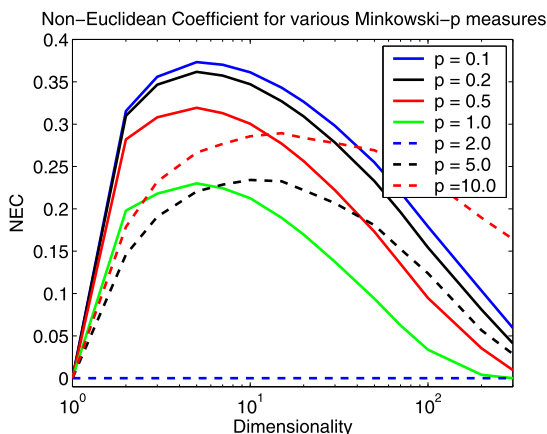
To inspect the amount of non-Euclidean influence in the derived PES, we define Negative EigenFraction (NEF) as:

$$\text{NEF} = \sum_{j=p+1}^{p+q} |\lambda_j| / \sum_{i=1}^{p+q} |\lambda_i| \in [0, 1]. \quad (2.6)$$

Figure 2.3 shows how NEF varies as a function of p of the Minkowski- p dissimilarity measure (k -dimensional spaces) for a two-dimensional example:

$$\rho_{\text{Min}_p}(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^k [x_i - y_i]^p \right)^{1/p}. \quad (2.7)$$

Fig. 2.4 The Non-Euclidean Coefficient for various Minkowski- p dissimilarity measures as a function of the dimensionality of a set of 100 points generated by a standard Gaussian distribution



This dissimilarity measure is Euclidean for $p = 2$ and metric for $p > 1$. The measure is non-Euclidean for all $p \neq 2$. The value of NEC may vary considerably with a changing dimensionality. This phenomenon is illustrated in Fig. 2.4 for 100 points generated by a standard Gaussian distribution for various values of p . The one-dimensional dissimilarities obviously fit perfectly to a Euclidean space. For a high dimensionality, the sets of dissimilarities become again better embeddable in a Euclidean space.

2.3.3 Discussion on Dissimilarity-Based Vector Spaces

Now we want to make some remarks on the two procedures for deriving vector spaces from dissimilarity matrices, as discussed in previous section.

The dissimilarity space interprets the dissimilarity vectors, defined by the dissimilarities from objects to particular prototypes from the representation set, as features. The true characteristics behind the used dissimilarity measure is not used when a general classifier is applied in a dissimilarity space. Special classifiers are needed to make use of that information. The good side of this ‘disadvantage’ is that the dissimilarity space can be used for any dissimilarity representation, including ones that are negative, asymmetric or weird, otherwise.

The embedding procedure is more restrictive. The dissimilarities are assumed to be symmetric and become zero for identical objects. A pseudo-Euclidean space is needed for a perfect embedding in case of non-Euclidean data sets. A pseudo-Euclidean space is however “broader” than the original distance measure in the sense that it allows negative square distances. Moreover, the requirements of a proper metric or well-defined distances obeying the triangle inequality are not of use as they do not guarantee a Euclidean embedding.

A severe drawback of both procedures is that they initially generate vector spaces that have as many objects as dimensions. Specific classifiers or dimension reduction

procedures are thereby needed. For the dissimilarity representation, this is more feasible than for the feature representation: features can vary greatly in their discriminative power, range, costs, or characteristics. Some features may be very good, others might be useless, or only useful in relation with particular other features.

This is not true for dissimilarities. The initial representation is based on objects which have similar characteristics. It is not beneficial to use two objects that are much alike as it leads to highly correlated dissimilarity vectors. Systematic, or even random procedures that reduce the initial representation set (in fact, prototype selection) can be very effective [38] for this reason.

A relevant topic in the comparison of both procedures is the representation of new objects in a given space derived from dissimilarities between an earlier set of objects (“projection”). For the dissimilarity space, this is simple. It is defined by the dissimilarities with the representation set used to define the space. A “projection” into a pseudo-Euclidean space is not straight forward. The space itself is found by the eigenvalue decomposition. Traditionally, new objects are projected into such a space by determining the point with the shortest distance. For pseudo-Euclidean spaces, however, this is not appropriate as distances can be negative. The projection point can thereby be chosen such that it has an arbitrarily large negative distance. The consequence is that in case new objects are considered the space has to rebuild from the combined set of old and new objects. This is directly related to the final observation made in Sect. 2.2.2.3 about the need to use transductive inference for non-Euclidean data.

2.4 Classifiers

We will discuss here a few well-known classifiers and their behavior in various spaces. This is a summary of our experiences based on numerous studies and applications. See [18, 21, 55] and their references.

In order to make a choice between the embedded pseudo-Euclidean space and the dissimilarity space for classifier training one should take into account the essential differences between these spaces. Pseudo-Euclidean embedding aims to preserve the given distances, while the dissimilarity space is not concerned about it. In addition, there is a nonlinear transformation between these spaces: the dissimilarity space can be defined by computing the distances to the prototypes in the embedded space. As a consequence, a linear classifier in the embedded space is a nonlinear classifier in the dissimilarity space. The reverse holds as well, but it should be kept in mind that the dissimilarity space is more general. As it is also defined for arbitrary, even asymmetric, dissimilarities, classifiers will relate to possible objects that do not exist in the embedded space.

It is outside the scope of this chapter, but the following observation might be helpful for some readers. If the dissimilarities are not constructed by a procedure on a structural representation of objects, but are derived as Euclidean distances in a feature space, then the pseudo-Euclidean embedding effectively reconstructs the

original Euclidean feature space (except for orthonormal transformations). So in that case a linear classifier in the dissimilarity space is a nonlinear classifier in the embedded space, which is the same nonlinear classifier in the feature space. Such a classifier, computed in a dissimilarity space, can perform very well [17, 21].

2.4.1 Nearest Neighbor Classifier

The k -nearest neighbor (k -NN) classifier in an embedded (pseudo-)Euclidean space is based on the distances computed in this space. By definition, these are the original dissimilarities (provided that the test examples are embedded together with the training objects). So without the process of embedding, this classifier can directly be applied to a given dissimilarity matrix but is simultaneously also a classifier for the embedded space. This is the classifier traditionally used by many researchers in the area of structural pattern recognition. The study of the dissimilarity representation arose because this classifier did not make use of the dissimilarities between the objects in the training set. Classification is entirely based on the dissimilarities of a test object to the objects in the training (or representation) set only.

The k -NN rule computed in the dissimilarity space relies on a Euclidean distance between the dissimilarity vectors, hence the nearest neighbors are determined by using all dissimilarities of a given object to the representation objects. As explained in Sect. 2.3.1, it is already mentioned that the distances between similar objects are small in the two spaces for large training sets and the metric distance. So, it is expected that learning curves are asymptotically identical. However, for small training sets the k -NN classifier in the dissimilarity space performs usually better than the direct k -NN rule as it uses more information.

2.4.2 Parzen Density Classifiers

The class densities computed by the Parzen kernel density procedure are based on pairwise distance computations between objects. The applicability of this classifier as well as its performance is thereby related to those of the k -NN rule. The major difference is that this classifier is smoother, depending on the choice of the smoothing parameter (kernel) and that its optimization involves the entire training set.

2.4.3 Normal Density Bayes Classifiers

Bayes classifiers assume that classes can be described by probability density functions. The expected classification error is minimized by using class priors and the Bayes' rule. In case of normal density functions, either a linear classifier (Linear

Discriminant Analysis, LDA) arises on the basis of equal class covariances, or a quadratic classifier is obtained for the general case (Quadratic Discriminant Analysis, QDA). These two classifiers are the best possible in case of (nearly) normal class distributions and a sufficiently large training set. As mean vectors and covariance matrices can be computed in a pseudo-Euclidean space, see [24, 55], these classifiers can be re-defined there as well if we forget the starting point of normal distributions. The reason is that normal distributions are not well defined in pseudo-Euclidean spaces; it is not clear what a normal distribution is unless we refer to associated Euclidean spaces.

In a dissimilarity space, the assumption of normal distributions often works very well. This is due to the fact that in many cases dissimilarity measures are based on, or related to sums of numerical differences. Under certain conditions, large sums of random variables tend to be normally distributed. It is not perfectly true for distances as we often get Weibull [8] or χ^2 distributions, but the approximations are sufficient for a good performance of LDA and QDA. The effect is emphasized if the classification procedure involves the computation of linear subspaces, e.g., by PCA. Thanks to projections the aspect of normality is emphasized even more.

2.4.4 Fisher's Linear Discriminant

In a Euclidean space, the Fisher linear discriminant (FLD) is defined as the linear classifier that maximizes the Fisher criterion, i.e., the ratio of the between-class variance to the within-class variance. For a two-class problem, the solution is equivalent to LDA (up to an added constant), even though no assumption is made about normal distributions. Since variance and covariance matrices are well defined in pseudo-Euclidean spaces, the Fisher criterion can be used to derive the FLD classifier there. Interestingly, FLD in a pseudo-Euclidean space coincides with FLD in the associated Euclidean space. FLD is a linear classifier in a pseudo-Euclidean space, but can be rewritten to FLD in the associated space; see also [29, 42].

In a dissimilarity space, which is Euclidean by definition, FLD coincides with LDA for a two-class problem. The performances of these classifiers may differ for multi-class problems as the implementations of FLD and LDA will usually vary then. Nevertheless, FLD performs very well. Due to the nonlinearity of the dissimilarity measure, FLD in a dissimilarity space corresponds to a nonlinear classifier in the embedded pseudo-Euclidean space.

2.4.5 Logistic Classifier

The logistic classifier is based on a model of the class posterior probabilities as a function of the distance to the classifier [1]. The distance between a vector and a linear hyperplane in a pseudo-Euclidean space, however, is an unsuitable concept

for classification as it can have any value in $(-\infty, \infty)$ for vectors on the same side of this hyperplane. We are not aware of a definition and an implementation of the logistic classifier for pseudo-Euclidean spaces. Alternatively, the logistic classifier can be constructed in the associated Euclidean space.

In a dissimilarity space, the logistic classifier performs well, although in practice normal density based classifiers work often better. It relaxes the demands for normality as made by LDA. It is also more robust in case of high-dimensional spaces.

2.4.6 Support Vector Machine (SVM)

The linear kernel in a pseudo-Euclidean space is indefinite (non-Mercer). The quadratic optimization procedure used to optimize a linear SVM may thereby fail [28]. An SVM can, however, be constructed if the contribution of the positive subspace of the Euclidean space is much stronger than that of the negative subspace. Mathematically, it means that the measure is only slightly deviating from the Euclidean behavior and the solution of the SVM optimization is found in the positive definite neighborhood. Various researchers have reported good results in applying this classifier, e.g., see [5]. Although the solution is not guaranteed and the algorithm (in this case LIBSVM, [10]) does not stop at the global optimum, a good classifier can be constructed.

In case of a dissimilarity space, the (linear) SVM is particularly useful for computing classifiers in the complete space in which the representations set equals the training set, $R := \mathcal{X}$, see Sect. 2.3.1. The given training set \mathcal{X} defines therefore a separable problem. The SVM classifier is well defined. It does not overtrain or only overtrains just slightly. The advantage of this procedure is that it does not demand a reduction of the representation set. By a suitable normalization of the dissimilarity matrix (such that the average dissimilarity is one), we found stable and good results in many applications by setting the trade-off parameter C in the SVM procedure [11] to $C = 100$. Hereby, additional cross-validation loops are avoided to optimize this parameter. As a result, in an application one can choose to focus on optimizing the dissimilarity measure.

2.5 Transformations

We will summarize the problem of building vector spaces from non-Euclidean dissimilarities as discussed so far:

- Non-Euclidean dissimilarities naturally arise in comparing real world objects for recognition purposes (see Sect. 2.2.2 and in particular Sect. 2.2.2.3).
- The pseudo-Euclidean space (see Sect. 2.3.2) offers a proper isometric embedding for non-Euclidean data while the dissimilarity space (see Sect. 2.3.1) postulates an Euclidean space in which just under some conditions, asymptotically, the nearest neighbor relations may be consistent with the given ones.

- The definition of classifiers in the pseudo-Euclidean space is not straightforward, and many of the standard tools developed for statistical pattern recognition and machine learning are not valid or need to be redesigned. The dissimilarity space, however, is a standard vector representation that can be used as the traditional feature space (see Sect. 2.4).
- The representation of new objects for classification purposes is for the pseudo-Euclidean space not well defined and for the dissimilarity space straightforward (see Sect. 2.3.3). The only proper existing solution for the pseudo-Euclidean space is to include these objects in construction of the space, at the cost of re-training the classifiers. This type of transductive learning [49] is fundamentally related to non-Euclidean object dissimilarities (see Sect. 2.2.2.3). For the dissimilarity spaces, transduction can be easily realized (at the cost of retraining the classifiers) or skipped (at the cost of accuracy).

Given the above, the dissimilarity space is preferred in most applications. It is easy to define and to handle. There is a one-to-one relation with the constituting dissimilarity matrix between the given objects. Any change in this matrix is reflected in a change of the representation. Moreover, this change is continuous. It can thereby be stated that there is no loss of information. The pseudo-Euclidean embedding, on the other hand, is of fundamental interest as it directly reflects the non-Euclidean aspects of the data. It is thereby a perfect place to study the question whether the non-Euclideaness contributes to the recognition performance or disturbs it.

One way to do this is to investigate transformations of the pseudo-Euclidean space that shrink or remove the non-Euclideaness. We discuss shortly a number of possibilities. See [20, 22] for more information.

2.5.1 The Dissimilarity Space (DS)

The original pseudo-Euclidean space, based on all eigenvectors, offers an isometric embedding of the given dissimilarities. So if we compute the distances in this space between all objects, the original dissimilarity matrix is obtained and thereby the dissimilarity space. If the pseudo-Euclidean space is first transformed, e.g., by rescaling or by deleting some axes (eigenvectors of the original embedding), then in a similar way a dissimilarity space can be obtained. This Euclidean space reflects all information of such a transformed pseudo-Euclidean space. As the transformation is continuous, then for any classifier in the pseudo-Euclidean space there exists a classifier in the dissimilarity space that yields the same classification. The transformation, however, is nonlinear. So a linear classifier in the pseudo-Euclidean space is nonlinear in the dissimilarity space and the other way around. Consequently, classifiers trained in these spaces before and after transformation yield different performances.

2.5.2 *The Positive Part of the Pseudo Euclidean Space (PES+)*

The most obvious correction for a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ is to neglect the negative definite subspace. This results in a p -dimensional Euclidean space \mathbb{R}^p with many-to-one mappings. Consequently, it is possible that the class overlap for the training set increases. It may, however, be worthwhile if the negative eigenvalues in the embedding procedure are mainly the result of noise and are not informative for the class separation. In that case, this correction may improve the classification.

2.5.3 *The Negative Part of the Pseudo Euclidean Space (PES-)*

In case the positive definite subspace of the pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ is neglected, a q -dimensional Euclidean space \mathbb{R}^q is obtained. It is expected for real world applications that this space will show a bad class separation. As in this space, however, all information is collected that makes the dissimilarities non-Euclidean, any separation will indicate that such useful information exists.

2.5.4 *The Associated Euclidean Space (AES)*

Since $\mathbb{R}^{(p,q)}$ is a vector space, we can equip it with the traditional inner product, which leads to the so-called associated Euclidean space \mathbb{R}^{p+q} . It means that the vector coordinates are identical to those of PES, but now we use the norm and distance measure that are Euclidean. This is consistent with the natural topology of a vector space. This solution is identical to the one obtained by classical scaling based on the magnitudes of eigenvalues [25, 55].

2.5.5 *Dissimilarity Enlargement by a Constant (DEC)*

Instead of modifying the embedding procedure, the dissimilarity matrix may be adapted such that it is embeddable into a Euclidean space. A simple way to avoid the negative eigenvalues is to increase all off-diagonal elements of the dissimilarity matrix such that $d_c^2(x_i, x_j) = d^2(x_i, x_j) + 2c, \forall i \neq j$. The value of c is chosen such that $c \geq -\lambda_{\min}$, where λ_{\min} is the smallest negative eigenvalue in the pseudo-Euclidean embedding of D . As a result, all eigenvalues are increased by c [55].

In our experiments, we set $c = -\lambda_{\min}$. Since the eigenvalues reflect the variances of the embedded data, the dimensions of the resulting Euclidean space are unevenly scaled by $\sqrt{\lambda_i + c}$. Note that the dimension with the largest negative contribution in PES has now a zero variance. In this way, dimensions related to noisy negative eigenvalues are more pronounced [55].

2.6 Are Non-Euclidean Dissimilarity Measures Informative?

The question about informativeness of non-Euclidean dissimilarity measures is different than the question whether non-Euclidean measures are better than Euclidean ones. The later question cannot be answered, in general. After studying a set of individual problems compared for a large set of dissimilarity measures, it might be found that for some problems the best measure is non-Euclidean. Such a result, however, is always temporary. A new Euclidean measure that outperforms the earlier ones may be invented later.

The question of informativeness, on the other hand, may be answered in an absolute sense. Even if a particular measure is not the best one, its non-Euclidean contribution can be judged as informative if the performance deteriorates by removing it. Should this result also be found by a classifier constructed in the non-Euclidean space? If a Euclidean correction can be found for an initially non-Euclidean representation that enables the construction of a good classifier, is the non-Euclidean dissimilarity measure then informative? We answer this question positively as any transformation can be included in the classifier and thereby effectively a classifier for the non-Euclidean representation has been found.

We will therefore state that the non-Euclidean character of a dissimilarity measure is non-informative if the classification result improves by removing its non-Euclidean contribution. The answer may be classifier dependent.

The traditional way of removing the non-Euclidean contribution is by neglecting the negative eigenvectors that define dimensions of the pseudo-Euclidean embedding. This is the PES+ defined in Sect. 2.5. The PES- can be used as a check to see whether there is any class separability in the negative part of the embedded space. The below experiments are entirely based on the dissimilarity spaces of the various spaces; see Sect. 2.5.

We analyze a set of public domain dissimilarity matrices used in various applications, as well as a few artificially generated ones. See Table 2.1 for some properties: *size* (number of objects), (number of) *classes*, *non-metric* (fraction of triangle violations, if zero the dataset is metric), *NEF* (negative eigenfraction, see Sect. 2.3.2.1) and *Rand Err* (classification error by random assignment). Every dissimilarity matrix is made symmetric by averaging with its transpose and normalized by the average off-diagonal dissimilarity. We compute the linear SVM in the dissimilarity spaces based on the original pseudo-Euclidean space (PES), the positive space (PES+) and the negative space (PES-). Error estimates are based on the leave-one-out crossvalidation. These experiments are done in a transductive way: test objects are included in the derivation of the embedded space as well as the dissimilarity representations.

The four Chickenpieces datasets are the averages of 11 dissimilarity matrices derived from a weighted edit distance between blobs [4]. FlowCyto is the average of four specific histogram dissimilarities including an automatic calibration correction. WoodyPlants is a subset of the shape dissimilarities between leaves of woody plants [30]. We used classes with more than 50 objects. Catcortex is based on the connection strength between 65 cortical areas of a cat [26]. Protein measures protein sequence differences using an evolutionary distance measure [27]. Balls3D is

Table 2.1 Classification errors of the linear SVM for several representations using the leave-one-out crossvalidation

	Size	Classes	Non-metric	NEF	Rand Err	PES \rightarrow DS	PES+ \rightarrow DS	PES- \rightarrow DS
Chickenpieces45	446	5	0	0.156	0.791	0.022	0.132	0.175
Chickenpieces60	446	5	0	0.162	0.791	0.020	0.067	0.173
Chickenpieces90	446	5	0	0.152	0.791	0.022	0.052	0.148
Chickenpieces120	446	5	0	0.130	0.791	0.034	0.108	0.148
WoodyPlants50	791	14	5e-4	0.229	0.928	0.075	0.076	0.442
CatCortex	65	4	2e-3	0.208	0.738	0.046	0.077	0.662
Protein	213	4	0	0.001	0.718	0.005	0.000	0.634
Balls3D	200	2	3e-4	0.001	0.500	0.470	0.495	0.000
GaussM1	500	2	0	0.262	0.500	0.202	0.202	0.228
GaussM02	500	2	5e-4	0.393	0.500	0.204	0.174	0.252
CoilYork	288	4	8e-8	0.258	0.750	0.267	0.313	0.618
CoilDelftSame	288	4	0	0.027	0.750	0.413	0.417	0.597
CoilDelftDiff	288	4	8e-8	0.128	0.750	0.347	0.358	0.691
NewsGroups	600	4	4e-5	0.202	0.733	0.198	0.213	0.435
BrainMRI	124	2	5e-5	0.112	0.499	0.226	0.218	0.556
Pedestrians	689	3	4e-8	0.111	0.348	0.010	0.015	0.030

an artificial dataset based on the surface distances of randomly positioned balls of two classes having a slightly different radius. GaussM1 and GaussM02 are based on two 20-dimensional normally distributed sets of objects for which dissimilarities are computed using the ℓ_p -norm (Minkowski) distances with $p = 1$ (metric, non-Euclidean) and $p = 0.2$ (non-metric). The three Coil datasets are based on the same sets of SIFT points in the COIL images compared by different graph distances. BrainMRI is the average of 182 dissimilarity measures obtained from MRI brain images. Pedestrians is a set of dissimilarities between detected objects (possibly pedestrians) in street images of the classes ‘pedestrian’, ‘car’ and ‘other’. They are based on cloud distances between sets of feature points derived from single images.

The table shows examples of non-Euclidean datasets for which the non-Euclideanness is informative, as well datasets for which it is non-informative. In all cases where the error of the PES- is significantly better than the error of random assignment, the negative space is informative. It contributes clearly to the classification performance based on the entire space for the Chickenpieces datasets as in these cases the error for just the positive space, PES+ is clearly worse than for the entire space, PES. BrainMRI is an example of a dataset for which the non-Euclideanness is non-informative as the negative part of the space does not contribute. The artificial dataset Balls3D has been successfully constructed such that all information is in the negative part of the space: classes can be entirely separated by PES- and the positive part, PES+, can be better removed.

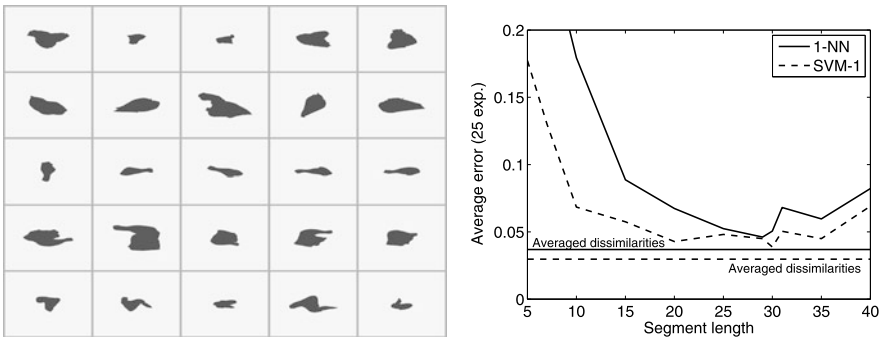


Fig. 2.5 (Left) Some examples of the Chickenpieces dataset. (Right) The error curves as a function of the segment length L

2.7 Examples

In this section, we will discuss a few examples that are typical for the use of dissimilarities in structural pattern recognition problems. They have been published by us before [18] and are repeated here as they may serve well as an illustration.

2.7.1 Shapes

A simple and clear example of a structural pattern recognition problem is the recognition of blobs: 2D binary structures. An example is given in Fig. 2.5. It is an object out of the five-class Chickenpieces dataset consisting of 445 images [2]. One of the best structural recognition procedure uses a string representation of the contour described by a set of segments of the same length [4]. The string elements are the consecutive angles of these segments. The weighted edit distances between all pairs of contours are used to compute the pairwise dissimilarities. This measure is non-Euclidean.

A (γ, L) family of problems is considered depending on the specific choice for the cost of one editing operation γ as well as for the segment's length L used in the contour description. As a result, the classification performance depends on the parameters used, as shown in Fig 2.5, right. 10-fold cross-validation errors are shown there for the 1-NN rule directly applied on the dissimilarities as well as the results for the linear SVM computed by LIBSVM (see [10]) in the dissimilarity space. In addition, the results are presented for the average of the 11 dissimilarity matrices. We can observe that the linear classifier in the dissimilarity space (SVM-1) improves the traditional 1-NN results and that combining of the dissimilarities improves the results further on.

2.7.2 Histograms and Spectra

Histograms and spectra offer very simple examples of data representations that are judged by human experts on their shape. In addition, also the sampling of the bins or wavelengths may serve as a useful vector representation for an automatic analysis. This is thanks to the fact that the domain is bounded and that spectra are often aligned. Below we give an example in which the dissimilarity representation outperforms the straightforward vector representation based on sampling because the first can correct for a wrong calibration (resulting in an imperfect alignment) in a pairwise fashion. Another reason to prefer dissimilarities for histograms and spectra over sampled vectorial data is that a dissimilarity measure encodes shape information. See the papers by Porro [44, 45] for more details.

We will consider now a dataset of 612 FL3-A DNA flowcytometer histograms from breast cancer tissues in a resolution of 256 bins. The initial data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000–2004, using the four tubes 3–6 of a DACO Galaxy flow cytometer. Histograms are labeled into three classes: aneuploid (335 patients), diploid (131) and tetraploid (146). We averaged the histograms of the four tubes thereby covering the DNA contents of about 80000 cells per patient. We removed the first and the last bin of every histogram as here outliers are collected, thereby obtaining 254 bins per histogram. Examples of histograms are shown in Fig. 2.6. The following representations are used:

Histograms. Objects (patients) are represented by the normalized values of the histograms (summed to one) described by a 254-dimensional vector. This representation is similar to the pixel representation used for images as it is based on just a sampling of the measurements.

Euclidean distances. These dissimilarities are computed as the Euclidean distances in the vector space mentioned above. Every object is represented by a vector of distances to the objects in the training set.

Calibrated distances. As the histograms may suffer from an incorrect calibration in the horizontal direction (DNA content), for every pairwise dissimilarity we compute the multiplicative correction factor for the bin positions that minimizes their dissimilarity. Here we used the ℓ_1 -distance. This representation makes use of the shape structure of the histograms and removes an invariant (the wrong original calibration).

A linear SVM with a fixed trade-off parameter C is used in learning. The learning curves for the three representations are shown in the bottom right of Fig. 2.6. They illustrate how for this classifier the dissimilarity representation leads to better results than the vector representation based on the histogram sampling. The use of the background knowledge in the definition of the dissimilarity measure improves the results further on.

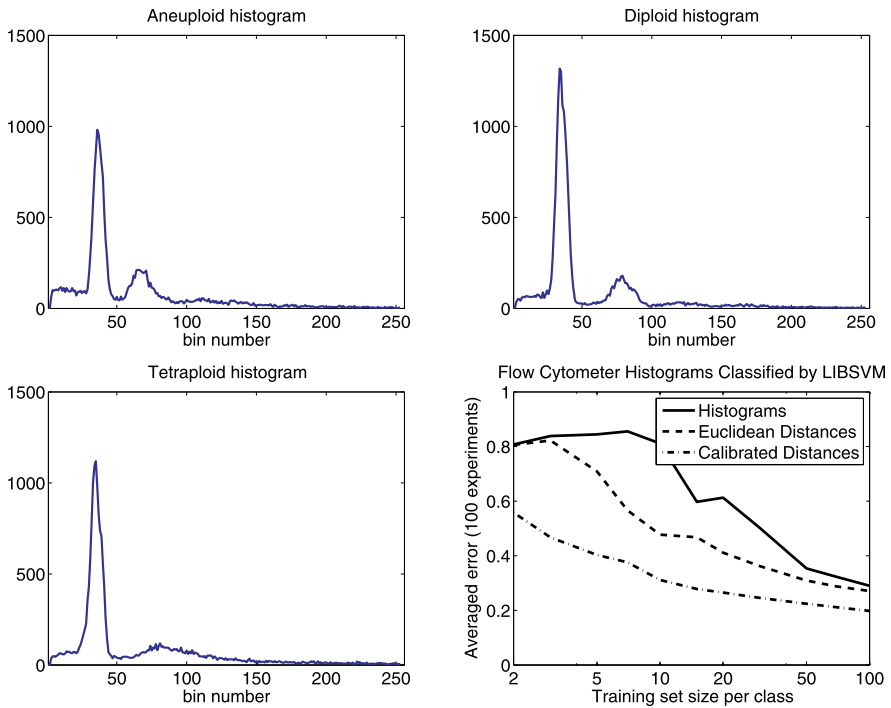


Fig. 2.6 Examples of some flowcytometer histograms: aneuploid, diploid and tetraploid. *Bottom right* shows the learning curves

2.7.3 Images

The recognition of objects on the basis of the entire image can only be done if these images are aligned. Otherwise, earlier pre-processing or segmentation is necessary. This problem is thereby a 2-dimensional extension of the histogram and spectra recognition task.

We will show an example of digit recognition by using a part of the classic NIST database of handwritten numbers [50] on the basis of random subsets of 500 digits for the ten classes 0–9. The images were resampled to 32×32 pixels in such a way that the digits fit either horizontally or vertically. Figure 2.7 shows a few examples: black is ‘1’ and white is ‘0’. The dataset is repeatedly split into training and test sets and hold-out classification is applied. In every split, the ten classes are evenly represented.

The following representations are used:

Features. We used 10 moments: the seven rotations invariant moments and the moments [00], [01], [10], measuring the total number of black pixels and the centers of gravity in the horizontal and vertical directions.

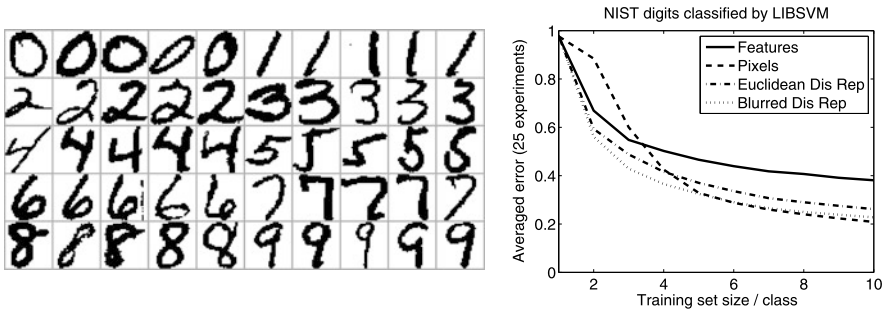


Fig. 2.7 (Left) Examples of the images used for the digit recognition experiment. (Right) The learning curves

Pixels. Every digit is represented by a vector of the intensity values in $32 * 32 = 1024$ dimensional vector space.

Dissimilarities to the training object. Every object is represented by the Euclidean distances to all objects in the training set.

Dissimilarities to blurred digits in the training set. As the pixels in the digit images are spatially connected, blurring may emphasize this. In this way, the distances between slightly rotated, shifted or locally transformed but otherwise identical digits become small.

The results are shown in Fig. 2.7 on the right. They show that the pixel representation is superior for large training sets. This is to be expected as this representation stores asymptotically the universe of possible digits. For small training sets, a suitable set of features may perform better. The moments we use here are very general features. Better ones can be found for digit description. As explained before, a feature-based description reduces the (information on the) object: it may be insensitive for some object modifications. For sufficiently large representation sets, the dissimilarity representation may see all object differences and may thereby perform better.

2.7.4 Sequences

The recognition of sequences of observations is in particular difficult if the sequences of a given class vary in length, but capture the same ‘story’ (information) from the beginning to the end. Some may run faster, or even run faster over just a part of the story and slow down elsewhere. A possible solution is to rely on Dynamic Time Warping (DTW) that relates the sequences in a nonlinear way, yet obeys the order of the events. Once two sequences are optimally aligned, the distance between them may be computed.

An example in which the above has been applied successfully is the recognition of 3-dimensional gestures from the sign language [35] based on an statistically optimized DTW procedure [3]. We took a part of a dataset of this study: the 20 classes

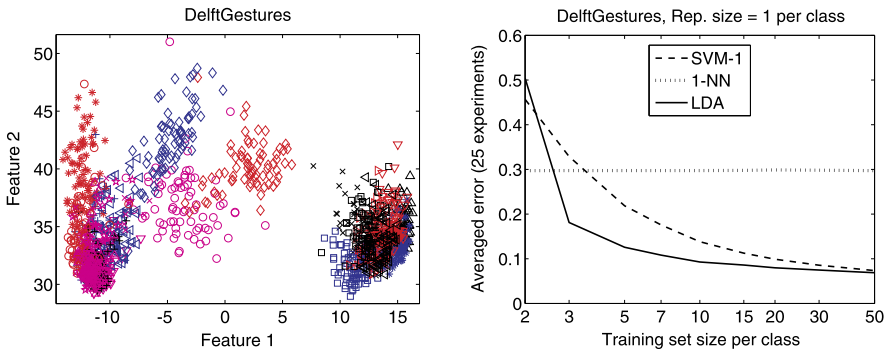


Fig. 2.8 PCA and learning curves for the 20-class Delft Gesture Dataset

(signs) that were most frequently available. Each of these classes has 75 examples. The entire dataset thereby consists of a 1500×1500 matrix of DTW-based dissimilarities. The leave-one-out 1-NN error for this dataset is 0.041, which is based on the computation of 1499 DTW dissimilarities per test object. In Fig. 2.8, left, a scatterplot is shown of the first two PCA components showing that some classes can already be distinguished with these two features (linear combinations of dissimilarities).

We studied dissimilarity representations consisting of just one randomly drawn example per class. The resulting dissimilarity space has thereby 20 dimensions. New objects have to be compared with just these 20 objects. This space is now filled with randomly selected training sets, containing between 2 and 50 objects per class. Remaining objects are used for testing. Two classifiers are studied, the linear SVM (using the LIBSVM package [10]) with a fixed trade-off parameter $C = 100$ (we used normalized dissimilarity matrices with the average dissimilarities set to 100) and LDA. The experiment was repeated 25 times and the results averaged out.

The learning curves in Fig. 2.8, right, show the constant value of the 1-NN classifier performance using the dissimilarities to the single training examples per class only, and the increasing performances of the two classifiers for a growing number of training objects. Their average errors for 50 training objects per class is 0.07. Recall that this is still based on the computation of just 20 DTW dissimilarities per object as we work in the related 20-dimensional dissimilarity space. Our experiments show that LDA reaches an error of 0.035 for a representation set of three objects per class, i.e., 60 objects in total. Again, the training set size is 50 examples per class, i.e., 1000 examples in total. For testing new objects, one needs to compute a weighted sum (linear combination) of 60 dissimilarity values giving the error of 0.035 instead of computing and ordering 1500 dissimilarities to all training objects for the 1-NN classifier leading to an error of 0.041.

2.7.5 Graphs

Graphs¹ are the main representation for describing structure in observed objects. In order to classify new objects, the pairwise differences between graphs have to be computed by using a graph matching technique. The resulting dissimilarities are usually related to the cost of matching and may be used to define a dissimilarity representation. We present here classification results obtained with a simple set of graphs describing four objects in the Coil database [36] described by 72 images for every object. The graphs are the Delaunay triangulations derived from corner points found in these images; see [51]. They are unattributed. Hence, the graphs describe the structure only. We used three dissimilarity measures:

CoilDelftSame Dissimilarities are found in a 5D space of eigenvectors derived from the two graphs by the JoEig approach; see [34].

CoilDelftDiff Graphs are compared in the eigenspace with a dimensionality determined by the smallest graph in every pairwise comparison by the JoEig approach; see [34].

CoilYork Dissimilarities are found by graph matching, using the algorithm of Gold and Rangurajan; see [23].

All dissimilarity matrices are normalized such that the average dissimilarity is 1. In addition to the three dissimilarity datasets, we used also their averaged dissimilarity matrix.

In a 10-fold cross-validation experiment, with $R := T$, we use four classifiers: the 1-NN rule on the given dissimilarities and the 1-NN rule in the dissimilarity space (listed as 1-NND in Table 2.2), LDA on a PCA-derived subspace covering 99 % of the variance and the linear SVM with a fixed trade-off parameter $C = 1$. All experiments are repeated 25 times. Table 2.2 reports the mean classification errors and the standard deviations of these means in between brackets. Some interesting observations are:

- The CoilYork dissimilarity measure is apparently much better than the two CoilDelft measures.
- The classifiers in the dissimilarity space however are not useful for the CoilYork measure, but they are for the CoilDelft measures. Apparently, these two ways of computing dissimilarities are essentially different.
- Averaging all three measures significantly improves the classifier performance in the resulting dissimilarity space, even outperforming the original best CoilYork result. It is striking that this does not hold for the 1-NN rule applied to the original dissimilarities.

¹Results presented in this section are based on a joint research with Prof. Richard Wilson, University of York, UK, and Dr. Wan-Jui Lee, Delft University of Technology, The Netherlands.

Table 2.2 10-fold cross-validation errors averaged over 25 repetitions

Dataset	1-NN	1-NND	PCA-LDA	SVM-1
CoilDelftDiff	0.477 (0.002)	0.441 (0.003)	0.403 (0.003)	0.395 (0.003)
CoilDelftSame	0.646 (0.002)	0.406 (0.003)	0.423 (0.003)	0.387 (0.003)
CoilYork	0.252 (0.003)	0.368 (0.004)	0.310 (0.004)	0.326 (0.003)
Averaged	0.373 (0.002)	0.217 (0.003)	0.264 (0.003)	0.238 (0.002)

2.8 Discussion

The dissimilarity representation discussed in this chapter is in particular useful for applications in structural pattern recognition as it is a way the represent objects in their entirety. This may result in non-Euclidean or even non-metric dissimilarities. We have presented ways how to handle them, analyzed possible causes of the non-Euclideaness, and answered the question whether such dissimilarity measures can be informative. Finally, we presented a set of examples on real world data.

We will repeat and emphasize some significant observations and additionally touch a few topics that could not be treated.

In our analysis on the causes of non-Euclidean dissimilarities, we made the observation that may be caused naturally in the process of comparing real world objects (Sect. 2.2), in particular when vector spaces are defined on just a subset of the objects of interest. This implies that objects to be classified may have to be included in the analysis together with the training set (Sect. 2.2.2.3), so called transductive inference or transductive learning [49].

The non-Euclideaness is a problem when it is attempted to build vector spaces from given dissimilarity data. This bridges the fields of structural and statistical pattern recognition [5, 16, 18, 21]. Before this problem was faced, researchers just used dissimilarities for template matching or approximated the non-Euclidean dissimilarities by Euclidean ones. In this chapter, examples are given that show that the non-Euclidean part of the data (reflected in the so-called negative part of the pseudo-Euclidean space used for an isometrical embedding the dissimilarities; see Sect. 2.3.2.1) can be informative for the classification, see Sect. 2.6.

In Sect. 2.7, a number of real world examples has been given that show that the dissimilarity approach can contribute significantly to the solution of pattern recognition problems. The use of the dissimilarity space is thereby advantageous. It avoids the computational complexity of embedding dissimilarities in a pseudo-Euclidean space as well as the Euclidean correction of this space or the problems of constructing classifiers. We judge the study of pseudo-Euclidean embedding especially of interest for studying the informativeness of the non-Euclidean characteristics. The dissimilarity space preserves all non-Euclidean information but is itself Euclidean (see Sect. 2.5). There are many interesting issues related to the dissimilarity approach discussed in this chapter. A number of them are discussed elsewhere or hardly investigated so far. A first, obvious question is that relating all objects to all other objects results into a computational explosion. Moreover, it may seem that

there is not really a need to determine the dissimilarities to vary similar objects, which will become the case for growing training sets. Prototype selection is thereby of interest to reduce to size of the representation set. See [9] for some results and earlier references. Directly related to this is the question whether dissimilarities are useful for very large training sets. How to find the optimal set of prototypes for such cases? Is it possible to guarantee some asymptotically optimal result?

At this point, it is relevant to realize the following. If objects show a zero distance if and only if they are identical and if they are labeled unambiguously then classes do not overlap and a zero-error classifier is possible. What is the best way to reach this? Most classifiers assume class overlap. The study of classifiers that make use of the fact that classes do not overlap didn't make much progress after the definition of the original perceptron rule. The assumption of non-overlapping classes may also have a significant impact on the collection of training data and the definition of classifier performance. If classes do not overlap there is no need use a statistical approach based on density distributions. The definition of class domains may be sufficient. Training sets should in that case be representative for the domains and not for the distributions. This implies that it will be allowed to ask application experts for typical examples instead of selecting an i.i.d. dataset representative for the data distribution.

For most practical applications there will be many ways to define dissimilarity measures that are zero if and only if the objects are identical. Combining such measures usually improves the results. In particular, a straightforward averaging as applied in Sect. 2.7 is very interesting as it does not introduce additional parameters but just combines different types of information resulting in dissimilarity matrices of the same size and spaces of the same dimensionality in which data is better separable.

A new and significant application domain, next to structural pattern recognition, is the design of classification procedures for sets points in a feature space representing different parts of objects to be recognized, see Sect. 2.2.2.3. It is a generalization of the Multi-Instance Learning (MIL) problem and the bag-of-words classifiers. The proper design of dissimilarity measures between two sets of feature vectors representing two objects, adapted to the characteristics of the problem at hand is a fascinating issue [13, 33, 48, 52].

Once the basic tools for dissimilarity based classification are established, the next question will be to define the basic set of dissimilarity measures for various data types like for the above mentioned sets of feature vectors. For every more general domain of objects like images, spectra, time signals, a set of basic dissimilarity measures should be available to define an initial solution for most problems. Like for the areas of feature extraction and classifiers, the optimal approach should be tuned to the application, but the availability of a set of tools and examples may contribute to good solution of the pattern recognition problem at hand.

References

1. Anderson, J.A.: Logistic discrimination. In: Krishnaiah, P.R., Kanal, L.N. (eds.) Handbook of

- Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality, pp. 169–191. North-Holland, Amsterdam (1982)
2. Andreu, G., Crespo, A., Valiente, J.M.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In: Proceedings of ICNN'97, International Conference on Neural Networks, vol. II, pp. 1341–1346. IEEE Service Center, Piscataway (1997)
 3. Bahlmann, C., Burkhardt, H.: The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(3), 299–310 (2004)
 4. Bunke, H., Bühler, U.: Applications of approximate string matching to 2D shape recognition. *Pattern Recognit.* **26**(12), 1797–1812 (1993)
 5. Bunke, H., Riesen, K.: Graph classification based on dissimilarity space embedding. In: Structural, Syntactic, and Statistical Pattern Recognition, pp. 996–1007 (2008)
 6. Bunke, H., Sanfeliu, A. (eds.): Syntactic and Structural Pattern Recognition Theory and Applications. World Scientific, Singapore (1990)
 7. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. *Pattern Recognit. Lett.* **19**(3–4), 255–259 (1998)
 8. Burghouts, G.J., Smeulders, A.W.M., Geusebroek, J.M.: The distribution family of similarity distances. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
 9. Plasencia Calana, Y., García Reyes, E.B., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: ICPR 2010, pp. 177–180 (2010)
 10. Chang, C.-C., Lin, C.-J.: LIBSVM: A Library for Support Vector Machines (2001). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
 11. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995)
 12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
 13. Dieterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **89**, 31–71 (1997)
 14. Dubuisson, M.P., Jain, A.K.: Modified Hausdorff distance for object matching. In: Int. Conference on Pattern Recognition, vol. 1, pp. 566–568 (1994)
 15. Duin, R.P.W., Pękalska, E.: Non-Euclidean dissimilarities: causes and informativeness. In: Hancock, E.R., et al. (eds.) Proc. SSPR & SPR 2010 (LNCS), vol. 6218, pp. 324–333. Springer, Berlin (2010)
 16. Duin, R.P.W.: Non-Euclidean problems in pattern recognition related to human expert knowledge. In: Filipe, J., Cordeiro, J. (eds.) ICEIS. Lecture Notes in Business Information Processing, vol. 73, pp. 15–28. Springer, Berlin (2010)
 17. Duin, R.P.W., Loog, M., Pękalska, E., Tax, D.M.J.: Feature-based dissimilarity space classification. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR Contests. Lecture Notes in Computer Science, vol. 6388, pp. 46–55. Springer, Berlin (2010)
 18. Duin, R.P.W., Pękalska, E.: The dissimilarity representation for structural pattern recognition. In: CIARP (LNCS), vol. 7042, pp. 1–24. Springer, Berlin (2011)
 19. Duin, R.P.W., de Ridder, D., Tax, D.M.J.: Experiments with object based discriminant functions; a featureless approach to pattern recognition. *Pattern Recognit. Lett.* **18**(11–13), 1159–1166 (1997)
 20. Duin, R.P.W., Pękalska, E.: On refining dissimilarity matrices for an improved NN learning. In: ICPR, pp. 1–4 (2008)
 21. Duin, R.P.W., Pękalska, E.: The dissimilarity space: between structural and statistical pattern recognition. *Pattern Recognit. Lett.* **33**, 826–832 (2012)
 22. Duin, R.P.W., Pękalska, E., Harol, A., Lee, W.-J., Bunke, H.: On Euclidean corrections for non-Euclidean dissimilarities. In: SSPR/SPR, pp. 551–561 (2008)
 23. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(4), 377–388 (1996)

24. Goldfarb, L.: A new approach to pattern recognition. In: Kanal, L.N., Rosenfeld, A. (eds.) *Progress in Pattern Recognition*, vol. 2, pp. 241–402. Elsevier, Amsterdam (1985)
25. Gower, J.C.: Metric and Euclidean Properties of Dissimilarity Coefficients. *J. Classif.* **3**, 5–48 (1986)
26. Graepel, T., Herbrich, R., Bollmann-Sdorra, P., Obermayer, K.: Classification on pairwise proximity data. In: *Advances in Neural Information System Processing*, vol. 11, pp. 438–444 (1999)
27. Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.-R., Obermayer, K., Williamson, R.: Classification on proximity data with LP-machines. In: *ICANN*, pp. 304–309 (1999)
28. Haasdonk, B.: Feature space interpretation of SVMs with indefinite kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 482–492 (2005)
29. Haasdonk, B., Pełkalska, E.: Indefinite kernel Fisher discriminant. In: *ICPR*, pp. 1–4 (2008)
30. Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with non-metric distances: image retrieval and class representation. *IEEE TPAMI* **22**(6), 583–600 (2000)
31. Jain, A.K., Zongker, D.E.: Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(12), 1386–1391 (1997)
32. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In: *Proceedings of International Conference on Machine Learning*, pp. 143–151 (1997)
33. Kondor, R.I., Jebara, T.: A kernel between sets of vectors. In: *ICML*, pp. 361–368 (2003)
34. Lee, W.J., Duin, R.P.W.: An inexact graph comparison approach in joint eigenspace. In: *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 35–44 (2008)
35. Lichtenauer, J.F., Hendriks, E.A., Reinders, M.J.T.: Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(11), 2040–2046 (2008)
36. Nene, S.A., Nayar, S.K., Murase, H.: *Columbia object image library (COIL-100)*, Columbia University (1996)
37. Pełkalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. *Pattern Recognit. Lett.* **23**(8), 943–956 (2002)
38. Pełkalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. *Pattern Recognit.* **39**(2), 189–208 (2006)
39. Pełkalska, E., Duin, R.P.W.: *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, Singapore (2005)
40. Pełkalska, E., Duin, R.P.W.: Dissimilarity-based classification for vectorial representations. In: *ICPR* (3), pp. 137–140 (2006)
41. Pełkalska, E., Duin, R.P.W.: Beyond traditional kernels: classification in two dissimilarity-based representation spaces. *IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev.* **38**(6), 729–744 (2008)
42. Pełkalska, E., Haasdonk, B.: Kernel discriminant analysis with positive definite and indefinite kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(6), 1017–1032 (2009)
43. Pełkalska, E., Paclík, P., Duin, R.P.W.: A Generalized Kernel Approach to Dissimilarity Based Classification. *J. Mach. Learn. Res.* **2**(2), 175–211 (2002)
44. Porro-Muñoz, D., Duin, R.P.W., Talavera-Bustamante, I., Orozco-Alzate, M.: Classification of three-way data by the dissimilarity representation. *Signal Process.* **91**(11), 2520–2529 (2011)
45. Porro-Muñoz, D., Talavera, I., Duin, R.P.W., Hernández, N., Orozco-Alzate, M.: Dissimilarity representation on functional spectral data for classification. *J. Chemom.* 476–486 (2011)
46. Samsudin, N.A., Bradley, A.P.: Nearest neighbour group-based classification. *Pattern Recognit.* **43**(10), 3458–3467 (2010)
47. Sebe, N., Lew, M.S., Huijsmans, D.P.: Toward improved ranking metrics. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(10), 1132–1143 (2000)
48. Tax, D.M.J., Loog, M., Duin, R.P.W., Cheplygina, V., Lee, W.-J.: Bag dissimilarities for multiple instance learning. In: *LNCS. Lecture Notes in Computer Science*, vol. 7005, pp. 222–234. Springer, Berlin (2011)

49. Vapnik, V.: *Statistical Learning Theory*. Wiley, New York (1998)
50. Wilson, C.L., Garris, M.D.: *Handprinted character database 3*. Technical report, National Institute of Standards and Technology (1992)
51. Xiao, B., Hancock, E.R.: Geometric characterisation of graphs. In: CIAP, pp. 471–478 (2005)
52. Yang, J., Jiang, Y.-G., Hauptmann, A.G., Ngo, C.-W.: Evaluating bag-of-visual-words representations in scene classification. In: Ze Wang, J., Boujemaa, N., Del Bimbo, A., Li, J. (eds.) *Multimedia Information Retrieval*, pp. 197–206. ACM, New York (2007)

Chapter 3

SIMBAD: Emergence of Pattern Similarity

Joachim M. Buhmann

Abstract A theory of patterns analysis has to suggest criteria how patterns in data can be defined in a meaningful way and how they should be compared. *Similarity-based Pattern Analysis and Recognition* is expected to adhere to fundamental principles of the scientific process that are expressiveness of models and reproducibility of their inference. Patterns are assumed to be elements of a pattern space or hypothesis class and data provide “information” which of these patterns should be used to interpret the data. The mapping between data and patterns is constructed by an inference algorithm, in particular by a cost minimization process. Fluctuations in the data usually limit the precision that we can achieve to uniquely identify a single pattern as interpretation of the data. We advocate an information-theoretic perspective on pattern analysis to resolve this dilemma where the tradeoff between *informativeness* of statistical inference and their *stability* is mirrored in the information-theoretic optimum of high information rate and zero communication error. The inference algorithm is considered as a noisy channel which naturally limits the resolution of the pattern space given the uncertainty of the data.

3.1 Pattern Theory

Ulf Grenander has started the field of general pattern theory in the 1960s to formalize the notion of patterns in precise mathematical terms [12, 13]. Patterns are perceived as regular structures behind the data sources, i.e., “*the underlying deep regular structures are descriptions of the source, which are hidden via the sensing channel*” [13, p. 2]. Grenander’s General Pattern Theory combines algebra, geometry and statistics to explain the nature of data sources and, thereby, depicts a *generative modeling* perspective on pattern analysis. This philosophy argues for a distinct generative viewpoint to infer the probability law governing the data source.

In many real-world situations, the data are generated and represented in a high dimensional space and the information processing task focusses on a low dimen-

J.M. Buhmann (✉)

Swiss Federal Institute of Technology Zurich, Zurich, Switzerland
e-mail: jbuhmann@inf.ethz.ch

sional interpretation space. The analysis of visual data like images or videos provides a very convincing example of such a situation: intensity patterns that are sensed by a camera are mathematically represented as points in a space with $\#\{\text{intensities}\}^{\#\{\text{pixels}\}}$ dimensions. When segmenting an image in semantically distinct regions, the interpretation space contains $\#\{\text{segments}\}^{\#\{\text{sites}\}}$ elements where the range of intensities significantly exceeds the number of distinct segments and the number of sites is often much smaller than the number of pixel. The reader should note that the space of admissible segmentations is still exponentially large in the number of sites, but this pattern space is much smaller than the data space. This discrepancy between the complexity of the data space and the expressiveness of the hypothesis class hints at a common situation in inference where we acquire too little information for estimating the data source, but we can harvest enough information from the data source to select a set of “desirable” patterns given the data. Consequently, we adopt a discriminative view of pattern recognition: patterns that are inferred from the data are elements of an interpretation space called hypothesis class. These patterns constitute abstractions from the data generating mechanism and they are more or less closely related to the data source. In image segmentation, for example, the segments are related to object parts in a semantically meaningful way, but they do not characterize the intensity generating mechanism of light reflecting surfaces. Furthermore, the hypothesis class often also reflects information about the aim of pattern analysis, i.e., what the patterns are used for in subsequent information processing.

Pattern analysis requires more mathematical structure than solely a hypothesis class, i.e., a set of possible patterns. In addition to a hypothesis class, we would like to derive a “natural” neighborhood system or topology for the pattern space. Furthermore, most algorithmic search procedures for patterns in data require metric information to structure the pattern space. How can we discover this topological and metric structure of pattern spaces given data? Which mathematical theory can serve as a prototypical framework for this scientific program? I am convinced that the data have to tell us which patterns are indistinguishable or are very similar and what data properties allow us to differentiate between patterns. Stochastic influences in the data generation process often erase the distinguishability between patterns in the hypothesis class and render them equivalent, thereby providing topological information on the pattern space.

The inference of patterns from data is formulated as an algorithmic search for a stable subset of the underlying hypothesis class. Stability is required to guarantee that the pattern analysis process would yield an equivalent outcome for the same structure of the data source but a different realization of the noise process. A second, antagonistic requirement of the pattern analysis process is its specificity or informativeness: a small subset of the hypothesis class and in the noise free limit, a single hypothesis should be selected which poses a tradeoff to the stability requirement. Both design principles mirror the reproducibility and specificity requirement of scientific reasoning [31].

Pattern analysis algorithms often follow an optimization principle. Desired patterns are assigned a high score or low costs and undesirable patterns are discarded

by assigning a low score or high costs. In the following, we adopt the terminology of cost minimization rather than score maximization. A cost function defines a partial order of hypotheses where the most preferred hypotheses are distinguished by minimal costs. The noise in the data, however, may introduce fluctuations in the costs, and the hypotheses with minimal costs for one realization of the data may no longer minimize costs for a second realization of the data. Therefore, we advocate to stabilize the set of cost-minimal hypotheses by expanding it to a set of hypotheses with near-optimal costs, also called approximation set. The size of such an approximation set is determined by information theoretic considerations. Hypotheses in the approximation set are considered to be statistically indistinguishable.

3.2 Statistical Learning for Pattern Analysis

3.2.1 Objects, Measurements and Hypotheses

Pattern Analysis quantifies structures in data which usually relate to a set of objects. To mathematically characterize this problem domain, we have to define what we mean by measurements and hypotheses. Given is a *set of objects* $\mathbf{O}^{(n)} = \{O_1, \dots, O_n\} \subset \mathcal{O}$, $n \in \mathbb{N}$. Individual objects can be characterized by measurements either relative to an external reference frame, e.g., a coordinate system in a feature space, or by comparison to other objects. A measurement X is defined as a mapping of an object configuration in a measurement space, i.e.,

$$X : \mathcal{O}^1 \times \dots \times \mathcal{O}^r \rightarrow \mathbb{K}, \quad (O_1, \dots, O_r) \mapsto X_{O_1, \dots, O_r}. \quad (3.1)$$

The object configurations are often specified as collections of objects taken from the same object set $\mathcal{O}^1 = \dots = \mathcal{O}^r$. The most often used measurement type are feature vectors $X : \mathcal{O} \rightarrow \mathbb{R}^d$ ($r = 1$), denoted as $X_O \in \mathbb{R}^d$. Relational data ($r = 2$, $\mathcal{O}^1 = \mathcal{O}^2$) arise often in bioinformatics applications and in network analysis problems. They are defined as $X : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{R}$, where X_{O_1, O_2} denotes a proximity/similarity value between object O_1 and O_2 , both out of the same object space. For dyadic data ($r = 2$, $\mathcal{O}^1 \neq \mathcal{O}^2$), the first and the second object set can differ $\mathcal{O}^1 \neq \mathcal{O}^2$, e.g., when we analyse user \times website or patient \times gene data sets. More complicated data structures than vectors or relations, e.g., three-way ($r = 3$) data or (hyper)graphs, are occasionally employed in various applications.

In the following, we use the generic notation $\mathbf{X}^{(n)} \in \mathcal{X}^{(n)}$ for a *set of measurements* to characterize these n objects $\mathbf{O}^{(n)}$. $\mathcal{X}^{(n)}$ denotes the corresponding measurement space of n objects. To simplify notation we omit the superscript $^{(n)}$ whenever the dependence on problem size is clear.

A *hypothesis* $c(\cdot)$ of a pattern recognition problem is a function that assigns a set of objects or a set of object configurations to a pattern out of a pattern space \mathcal{P} , i.e.,

$$c : \mathcal{O}^1 \times \dots \times \mathcal{O}^s \rightarrow \mathcal{P}, \quad (O_1, \dots, O_s) \mapsto c(O_1, \dots, O_s). \quad (3.2)$$

The definition of hypotheses does not depend on the measurements X_{O_1, \dots, O_r} , but potential patterns that are denoted by hypotheses are defined *prior* to any measurements. The reader should note that the notion of a “feasible solution” in applied mathematics and optimization often depends on constraints that are determined by measurements contrary to the definition in (3.2). Such situations can be modeled by unconstrained solution spaces with infinite costs for those solutions that violate the constraints.

The *hypothesis class* for a pattern recognition problem is defined as the set of functions assigning an object or an object configuration¹ to an element of the pattern space, i.e.,

$$\mathcal{C}(\mathbf{O}) = \{c(\mathbf{O}) : \mathbf{O} \in \mathcal{O}\}. \quad (3.3)$$

A well-known example of a hypothesis class are the space of partitions or classification functions $c : \mathcal{O} \rightarrow \{1, \dots, k\}$ which we use in classification or clustering. When clustering n objects into k clusters, then we restrict the space of all possible partition functions to $\mathcal{P}^{(n)} = \{1, \dots, k\}^n$ for the object set $\mathbf{O}^{(n)}$. The corresponding hypothesis class is denoted by $\mathcal{C}^{(n)} = \mathcal{C}(\mathbf{O}^{(n)})$. For parameter estimation problems like PCA or SVD, the patterns are possible values of the orthogonal matrices and the pattern space is a subset of the d -dimensional Euclidean rotations.

3.2.2 Empirical Risk Approximation

The hypothesis class is a set of functions that map objects or object configurations to patterns. Pattern analysis requires to assess the quality of hypotheses $c \in \mathcal{C}$. We adopt a cost function (risk) viewpoint in this paper which attributes a non-negative cost value

$$R : \mathcal{C}^{(n)} \times \mathcal{X}^{(n)} \rightarrow \mathbb{R}_+, \quad (c, \mathbf{X}^{(n)}) \mapsto R(c, \mathbf{X}^{(n)}) \quad (3.4)$$

to each hypothesis given the measurements ($\mathbb{R}_+ := [0, \infty)$). The non-negativity assumption does not restrict the choice of cost functions since we can always replace $\tilde{R}(c, \mathbf{X}^{(n)}) := R(c, \mathbf{X}^{(n)}) - \inf_{c \in \mathcal{C}} R(c, \mathbf{X}^{(n)})$ for effectively computable minimal costs.

The classical theory of statistical learning [18, 19] advocates to use the empirical minimizer as the solution of the inference problem. The best empirical pattern denoted by $c^\perp(\mathbf{X}^{(n)})$ minimizes the empirical risk (ERM) of the pattern analysis problem given the measurements $\mathbf{X}^{(n)}$, i.e.,

$$c^\perp(\mathbf{X}^{(n)}) \in \arg \min_{c \in \mathcal{C}^{(n)}} R(c, \mathbf{X}^{(n)}). \quad (3.5)$$

¹In the following, we restrict hypotheses to map an object to a pattern. The more general situation of object configurations can be analyzed in an analogous way but involves a more complex notation.

Although hypotheses map objects into a pattern space, the empirical risk minimizer $c^\perp(\mathbf{X}^{(n)})$ depends on measurements.

The ERM theory requires for learnability of classifications that the hypothesis class is not “too complex” (i.e., finite VC-dimension) and, as a consequence, the ERM solution $c^\perp(\mathbf{X}^{(n)})$ converges to the optimal solution which minimizes the expected risk. A corresponding criterion has been derived for regression [1].

This classical learning theory is not applicable when the size of the hypothesis class grows exponentially with the number of objects like in clustering or other optimization problems of combinatorial nature. Without strong regularization, we cannot hope to identify a single solution which globally minimizes the expected risk in the asymptotic limit $n \rightarrow \infty$. Hypothesis classes of combinatorial problems often have an infinite VC dimension and, therefore, are not learnable in the classical VC sense. Therefore, we replace the concept of a unique function as the solution of a learning problem with a weighted set of functions. The challenge of learning then amounts to determine a weight measure which is concentrated on few solutions to achieve precision. The weights w are defined as functions which map triplets of a hypothesis, measurements and a resolution parameter to the unit interval, i.e.,

$$w : \mathcal{C}^{(n)} \times \mathcal{X}^{(n)} \times \mathbb{R}_+ \rightarrow [0, 1], \quad (c, \mathbf{X}^{(n)}, \beta) \mapsto w_\beta(c, \mathbf{X}^{(n)}). \quad (3.6)$$

The set of weights is denoted as $\mathcal{W}_\beta(\mathbf{X}^{(n)}) = \{w_\beta(c, \mathbf{X}^{(n)}) : c \in \mathcal{C}^{(n)}\}$.

How should we choose the weights $w_\beta(c, \mathbf{X}^{(n)})$ that large weights are only assigned to functions with low costs? The partial ordering constraint

$$R(c, \mathbf{X}^{(n)}) \leq R(\tilde{c}, \mathbf{X}^{(n)}) \Leftrightarrow w_\beta(c, \mathbf{X}^{(n)}) \geq w_\beta(\tilde{c}, \mathbf{X}^{(n)}), \quad (3.7)$$

ensures that functions with minimal costs $R(c^\perp, \mathbf{X}^{(n)})$ assume the maximal weight value. Weights are normalized to one w.l.o.g., i.e., $0 \leq w_\beta(c, \mathbf{X}^{(n)}) \leq 1$. The non-negativity constraint of weights allows us to write the weights as $w_\beta(c, \mathbf{X}^{(n)}) = \exp(-\beta f(R(c, \mathbf{X}^{(n)})))$ with the monotonic function $f(x)$. Since $f(x)$ amounts to a monotone rescaling of the costs $R(c, \mathbf{X}^{(n)})$ we resort w.l.o.g. to the common choice of Boltzmann weights with the inverse computational temperature β , i.e.,

$$w_\beta(c, \mathbf{X}^{(n)}) = \exp(-\beta R(c, \mathbf{X}^{(n)})). \quad (3.8)$$

It is worth mentioning that standard approximation sets as introduced in the theory of (additive) approximation algorithms would correspond to binary weights

$$w_\beta^{\text{bin}}(c, \mathbf{X}^{(n)}) = \begin{cases} 1 & \text{if } R(c, \mathbf{X}^{(n)}) \leq R(c^\perp, \mathbf{X}^{(n)}) + 1/\beta, \\ 0 & \text{otherwise.} \end{cases} \quad (3.9)$$

The weight $w_\beta(c, \mathbf{X}^{(n)})$ of a given hypothesis c is a random variable of the measurements $\mathbf{X}^{(n)}$. We consider the quantity

$$Z_\beta(\mathbf{X}^{(n)}) := \sum_{c \in \mathcal{C}^{(n)}} w_\beta(c, \mathbf{X}^{(n)}), \quad (3.10)$$

which measures the total weight of hypotheses with low costs. The weight sum is also known as the partition function in statistical physics when we use Boltzmann weights. In case of binary weights, $Z_\beta(\mathbf{X}^{(n)})$ denotes the number of solutions that are $1/\beta$ close to the optimum.²

3.2.3 Generalization and the Two-Instance Scenario

To determine the optimal regularization of a pattern recognition method, we have to define and estimate the generalization performance of hypotheses. We adopt the two-instance scenario with training and test data described by respective object sets \mathbf{O}' , \mathbf{O}'' and corresponding measurements $\mathbf{X}', \mathbf{X}'' \sim \mathbb{P}(\mathbf{X})$. Both sets of measurements are drawn i.i.d. from the same probability distribution $\mathbb{P}(\mathbf{X})$. The training and test data $\mathbf{X}', \mathbf{X}''$ define two optimization problems $R(\cdot, \mathbf{X}')$, $R(\cdot, \mathbf{X}'')$. The two-instance scenario or two-sample-set scenario is widely used in statistics and statistical learning theory [18], i.e., to bound the deviation of empirical risk from expected risk, but also for two-terminal systems in information theory [10].

Statistical pattern analysis requires that inferred patterns have to generalize from training data to test data since noise in the data might render the ERM solution $c^\perp(\mathbf{X}') \neq c^\perp(\mathbf{X}'')$ unstable. How can we evaluate the generalization properties of solutions to a pattern recognition problem? Before we can compute the costs $R(\cdot, \mathbf{X}'')$ on test data of approximate solutions $c(\mathbf{O}') \in \mathcal{C}(\mathbf{O}')$ on training data, we have to identify a pattern $c(\mathbf{O}'') \in \mathcal{C}(\mathbf{O}'')$ which corresponds to $c(\mathbf{O}')$. A priori, it is not clear how to compare patterns $c(\mathbf{O}')$ for objects \mathbf{O}' with patterns $c(\mathbf{O}'')$ for objects \mathbf{O}'' . Therefore, we define a bijective mapping

$$\psi : \mathcal{O}' \rightarrow \mathcal{O}'', \quad \mathbf{O}' \mapsto \psi \circ \mathbf{O}'. \quad (3.11)$$

The mapping ψ allows us to identify a pattern hypothesis for training set of objects $c' \in \mathcal{C}(\mathbf{O}')$ with a pattern hypothesis for a test set of objects $c'' \in \mathcal{C}(\psi \circ \mathbf{O}')$. The reader should note that such a mapping ψ might change the object indices. In cases when the objects \mathbf{O}' , \mathbf{O}'' are elements of an underlying metric space, a natural choice for ψ is the nearest neighbor mapping.

The mapping ψ enables us to evaluate pattern costs on test data \mathbf{X}'' for patterns $c(\mathbf{O}')$ selected on the basis of training data \mathbf{X}' . Consequently, we can determine how many training patterns with large weights share also large weights on test data, i.e.,

$$\Delta Z_\beta(\mathbf{X}', \mathbf{X}'') := \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_\beta(c, \psi \circ \mathbf{X}') w_\beta(c, \mathbf{X}''). \quad (3.12)$$

A large subset of hypotheses with jointly large weights indicates that low cost hypotheses on training data \mathbf{X}' also perform with low costs on test data. The tradeoff

²For binary weights, $Z_\beta(\mathbf{X}^{(n)})$ corresponds the microcanonical partition function by assuming that almost all solutions cost close to $R(c^\perp, \mathbf{X}^{(n)}) + 1/\beta$.

between stability and informativeness for Boltzmann weights (3.8) is controlled by maximizing β for given risk function $R(\cdot, \mathbf{X})$ under the constraint of large weight overlap $\Delta Z_\beta(\mathbf{X}', \mathbf{X}'')/\sqrt{Z_\beta(\mathbf{X}')Z_\beta(\mathbf{X}'')} \approx 1$. A quantitative statement how close this ratio should approach unity requires a statistical decision theory as provided by Shannon's approach to information transmission.

3.2.4 Typicality of Instances

A natural question in statistical inference arises from asymptotic considerations in the large n -limit. What is the asymptotic behavior of the log weight sum $\log Z_\beta(\mathbf{X}^{(n)})$ dependent on the problem/instance size n ? As remarked above, the measurements $\mathbf{X}^{(n)}$ of a particular pattern recognition instance depend on the value n .

In analogy to information theory (see [9, p. 58]), we assume that the log-weight sums converge according to an asymptotic equipartition property, i.e.,

$$\mathcal{F}' := \lim_{n \rightarrow \infty} -\frac{\log Z_\beta(\mathbf{X}'^{(n)})}{\log |\mathcal{C}(\mathbf{O}'^{(n)})|}, \quad (3.13)$$

$$\mathcal{F}'' := \lim_{n \rightarrow \infty} -\frac{\log Z_\beta(\mathbf{X}''^{(n)})}{\log |\mathcal{C}(\mathbf{O}''^{(n)})|}, \quad (3.14)$$

$$\Delta \mathcal{F} := \lim_{n \rightarrow \infty} -\frac{\log \Delta Z_\beta(\mathbf{X}'^{(n)}, \mathbf{X}''^{(n)})}{\log |\mathcal{C}(\mathbf{O}''^{(n)})|}. \quad (3.15)$$

These assumptions (3.13)–(3.15) require that the log-weight sums normalized by the size of the hypothesis class converge towards deterministic limits. The quantities \mathcal{F}' , \mathcal{F}'' are known as the free energies (up to a factor β^{-1}) for the instances $R(\cdot, \mathbf{X}')$, $R(\cdot, \mathbf{X}'')$ in statistical physics. The factor $\log |\mathcal{C}^{(n)}|$ denotes the problem size of the optimization problem, i.e., it is $O(n)$ for clustering problems with maximally k^n different partitions and $O(n \log n)$ for sorting problems with $\log |\mathcal{C}^{(n)}| = \log(n!)$.

Definition 3.1 The set $A_\epsilon^{(n)}$ of jointly typical instances w.r.t. $p(\mathbf{X}'^{(n)}, \mathbf{X}''^{(n)})$ is the set of instance pairs $(\mathbf{X}'^{(n)}, \mathbf{X}''^{(n)}) \in \mathcal{X}^{(n)} \times \mathcal{X}^{(n)}$ with empirical log partition functions close to the respective free energies

$$A_\epsilon^{(n)} = \left\{ (\mathbf{X}'^{(n)}, \mathbf{X}''^{(n)}) \in \mathcal{X}^{(n)} \times \mathcal{X}^{(n)} : \left| -\frac{\log Z_\beta(\mathbf{X}'^{(n)})}{\log |\mathcal{C}(\mathbf{O}'^{(n)})|} - \mathcal{F}' \right| < \epsilon, \right.$$

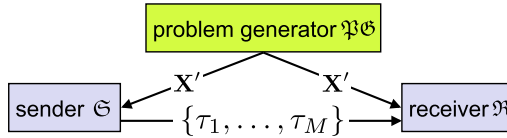


Fig. 3.1 Generation of a set of M code problems by, e.g., permuting the object indices

$$\left. \begin{aligned} \left| -\frac{\log Z_{\beta}(\mathbf{X}''^{(n)})}{\log |\mathcal{L}(\mathbf{O}''^{(n)})|} - \mathcal{F}'' \right| < \epsilon, \\ \left| -\frac{\log \Delta Z_{\beta}(\mathbf{X}'^{(n)}, \mathbf{X}''^{(n)})}{\log |\mathcal{L}(\mathbf{O}''^{(n)})|} - \Delta \mathcal{F} \right| < \epsilon \end{aligned} \right\}. \quad (3.16)$$

The reader should note that the weak law of large numbers guarantees convergence of empirical entropies towards their expectation values in Shannon's information theory. Due to the dependence of the weights $w_{\beta}(c, \mathbf{X}^{(n)})$ on the cost function $R(\cdot, \mathbf{X}^{(n)})$, convergence has to be required for a cost function. We also conjecture that cost functions which violate this convergence behavior cannot be used to define predictive models.

3.3 Coding by Approximation

In the following, we describe an information-theoretic framework to determine which hypotheses are statistically indistinguishable due to noise in the measurements, and consequently, how much we have to coarsen the hypothesis class. Shannon's random coding concept suggests a model theory to determine the maximal number of distinguishable n -bit strings in the Hamming space when the bit strings are exposed to noise in a communication channel. We develop a generalization of this idea for solution spaces of optimization problems. The weight distribution $w_{\beta}(c, \mathbf{X}^{(n)})$, $c \in \mathcal{C}$ over the hypothesis class \mathcal{C} corresponds to the subsets of bit strings assigned to a specific codebook vector in information theory. Noise perturbs the measurements, and therefore, the weight distribution fluctuates. An algorithm to approximately minimize a cost function and the measurements as input to this algorithm define a noisy channel in a hypothetical communication scenario³ with a sender \mathfrak{S} , a receiver \mathfrak{R} , and a problem generator $\mathfrak{P}\Phi$ (Fig. 3.1). The problem generator connects the sender with the receiver by posing an optimization problem given a cost function or an algorithm. Communication takes place by approximately optimizing a given cost function, i.e., by calculating weight sets $Z_{\beta}(\mathbf{X}')$, $Z_{\beta}(\mathbf{X}'')$. This coding concept will be referred to as approximation set coding (ASC) since the

³The reader should keep in mind that we are not interested in deriving a new principle for coding, but we exploit the communication metaphor to derive a quantitative criterion of how precisely we can approximate the global minimizer of a cost function by an approximation set.

weights are concentrated on approximate minimizers of the optimization problem. The noisy channel is characterized by a pattern cost function $R(c, \mathbf{X})$ which determines the channel capacity of the ASC scenario. Selection and validation of pattern recognition models are then achieved by maximizing the channel capacity over a set of cost functions $R_\theta(\cdot, \mathbf{X}), \theta \in \Theta$ where θ indexes the various cost functions or pattern recognition objectives. In a more general setting, an arbitrary algorithm which does not necessarily minimize a cost function can be considered to define a weight distribution and thereby, to play the role of a noisy channel [7] due to fluctuations in the input or in the execution path.

3.3.1 Code Design by Transformations

Before we describe the communication protocol, we have to define the code for communication. Shannon introduced his random coding theory to demonstrate the limits of asymptotically error free communication over a noisy channel. Random coding refers to the fact that messages in Shannon's random coding model are selected as a set of bit strings $\{\xi^{(j)} = (\xi_1^{(j)}, \dots, \xi_n^{(j)}), 1 \leq j \leq M\}$ with length $n = \lceil \log \mathcal{C}(\mathbf{O}^{(n)}) \rceil$ that are drawn i.i.d. according to a probability distribution $p(\xi)$. For sufficiently large n , the codewords all have mutual distances which are highly concentrated around the expected distance $2np(1-p)$ with the probability $p = \mathbb{P}(\xi^{(1)} = 1)$. In the asymptotic limit $n \rightarrow \infty$ for $p = 1/2$, the random codewords uniformly partition the Hamming space of n -bit sequences into subsets of bit strings which can be decoded without errors. In an analogous way, we cover the hypothesis class by weight distributions. To generate a uniform cover of the hypothesis class, we introduce a transformation

$$\tau : \mathcal{O} \rightarrow \mathcal{O}, \quad \mathbf{O} \mapsto \tau \circ \mathbf{O}. \quad (3.17)$$

The set of all possible transformations is denoted as \mathbb{T} . Transformations that are restricted to object sets $\mathbf{O}^{(n)}$ of n objects are denoted by $\tau^{(n)} \in \mathbb{T}^{(n)}$. A random cover of the hypothesis class is then generated by selecting a set of transformations $\mathcal{T} = \{\tau_j^{(n)} \in \mathbb{T}^{(n)} : 1 \leq j \leq M, \tau_j^{(n)} \sim P(\tau^{(n)})\}$ with a rate $\rho := \log M / \log |\mathbb{T}^{(n)}|$. A natural choice of the probability distribution for transformations is the uniform distribution $P(\tau^{(n)}) = 1/|\mathbb{T}^{(n)}|$. The intuition behind the transformations is the following:⁴ When a transformation is applied to an object set \mathbf{O} then the respective hypotheses $c(\mathbf{O})$ and the measurements $X_{\mathbf{O}}$ are transformed accordingly. Furthermore, the weights $w_\beta(c, \mathbf{X})$ are transformed by applying τ to c and \mathbf{X} , i.e., $\tau \circ w_\beta(c, \mathbf{X}) := w_\beta(\tau \circ c, \tau \circ \mathbf{X})$.

⁴Superscript ⁽ⁿ⁾ dropped for readability.

3.3.2 Typicality of Transformations

Analogous to Shannon's random coding strategy, we generate the transformations $\tau^{(n)} \sim P(\tau^{(n)})$ in a random way. The probability distribution $P(\tau^{(n)})$ is defined over the set of possible transformations $\mathbb{T}^{(n)}$. An asymptotic equipartition property depends on the entropy density of the transformation set

$$\mathcal{H}(\tau) := \lim_{n \rightarrow \infty} -\frac{\log P(\tau^{(n)})}{\log |\mathbb{T}^{(n)}|}. \quad (3.18)$$

For coding, we choose ϵ -typical transformations $\tau^{(n)} \in T_\epsilon^{(n)}$ with the typical set $T_\epsilon^{(n)}$ being defined in the following way:

Definition 3.2 The set $T_\epsilon^{(n)}$ of typical transformations w.r.t. $p(\tau^{(n)})$ is the set of transformations $\tau^{(n)} \in \mathbb{T}^{(n)}$ with the property

$$T_\epsilon^{(n)} = \left\{ \tau^{(n)} \in \mathbb{T}^{(n)} : \left| -\frac{\log P(\tau^{(n)})}{\log |\mathbb{T}^{(n)}|} - \mathcal{H}(\tau) \right| < \epsilon \right\}. \quad (3.19)$$

Special cases of such transformations $\tilde{\tau}^{(n)}$ are random permutations when optimizing combinatorial optimization cost functions like clustering models or graph cut problems. In parametric statistics, the transformations are parameter grids of, e.g., rotations when estimating the orthogonal transformations of PCA or SVD.

3.3.3 Communication Protocol

Sender \mathfrak{S} and receiver \mathfrak{R} agree on a cost function for pattern recognition $R(c, \mathbf{X}')$ and on a mapping function ψ . The following procedure is then employed to generate the code for the communication process:

1. Sender \mathfrak{S} and receiver \mathfrak{R} obtain data \mathbf{X}' from the problem generator $\mathfrak{P}\mathfrak{G}$.
2. \mathfrak{S} and \mathfrak{R} calculate the weight set $\mathcal{W}_\beta(\mathbf{X}')$.
3. \mathfrak{S} generates a set of (random) transformations $\mathcal{T} := \{\tau_1, \dots, \tau_M\}$. The transformations define a set of optimization problems $R(c, \tau_j \circ \mathbf{X}')$, $1 \leq j \leq M$ to determine weight sets $\mathcal{W}_\beta(\tau_j \circ \mathbf{X}')$, $1 \leq j \leq M$.
4. \mathfrak{S} sends the set of transformations \mathcal{T} to \mathfrak{R} who determines the set of weight sets $\{\mathcal{W}_\beta(\tau_j \circ \mathbf{X}')\}_{j=1}^M$.

The rationale behind this procedure is the following: Given the measurements \mathbf{X}' , the sender has randomly covered the hypothesis class $\mathcal{C}(\mathbf{O}')$ by respective weight sets $\{\mathcal{W}_\beta(\tau_j \circ \mathbf{X}') : 1 \leq j \leq M\}$. Communication can take place if the weight sets are stable under the stochastic fluctuations of the measurements. The criterion for

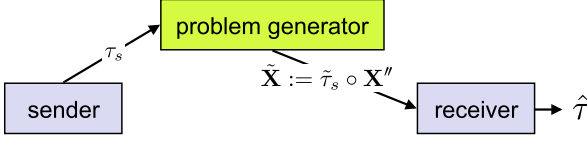


Fig. 3.2 Communication process: (i) the sender selects transformation τ_s , (ii) the problem generator draws $\mathbf{X}'' \sim \mathbb{P}(\mathbf{X})$ and applies $\tilde{\tau}_s = \psi \circ \tau_s \circ \psi^{-1}$ to it, and the receiver estimates $\hat{\tau}$ based on $\tilde{\mathbf{X}} = \tilde{\tau}_s \circ \mathbf{X}''$

reliable communication is defined by the ability of the receiver to identify the transformation which has been selected by the sender. After this setup procedure, both sender and receiver have a list of weight sets available.

How is the communication between sender and receiver organized? During communication, the following steps take place as depicted in Fig. 3.2:

1. The sender \mathfrak{S} selects a transformation τ_s as message and send it to the problem generator \mathfrak{PG} .
2. \mathfrak{PG} generates a new data set \mathbf{X}'' and establishes correspondence ψ between \mathbf{X}' and \mathbf{X}'' . \mathfrak{PG} then applies the selected transformation τ_s , yielding $\tilde{\mathbf{X}} = \psi \circ \tau_s \circ \psi^{-1} \circ \mathbf{X}''$.
3. \mathfrak{PG} send $\tilde{\mathbf{X}}$ to the receiver \mathfrak{R} without revealing τ_s .
4. \mathfrak{R} calculates the weight set $\mathcal{W}_\beta(\tilde{\mathbf{X}})$.
5. \mathfrak{R} estimates the selected transformation τ_s by using the decoding rule

$$\hat{\tau} \in \arg \max_{\tau \in \mathcal{T}} \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_\beta(c, \psi \circ \tau \circ \mathbf{X}') w_\beta(c, \tilde{\mathbf{X}}). \quad (3.20)$$

In the case of discrete hypothesis classes, then the communication channel is bounded from above by the cardinality of $\mathcal{C}(\mathbf{X})$ if two conditions hold: (i) the channel is noise free $\mathbf{X}' \equiv \mathbf{X}''$; (ii) the transformation set is sufficiently rich that every hypothesis can be selected as a global minimizer of the cost function.

3.4 Error Analysis of Approximation Set Coding

To determine the optimal approximation precision for an optimization problem $R(\cdot, \mathbf{X})$, we have to derive necessary and sufficient conditions which have to hold in order to reliably identify the transformations $\tau_s \in \mathcal{T}$. The parameter β , which controls the concentration of weights and thereby the resolution of the hypothesis class, has to be adapted to the size of the transformation set $|\mathcal{T}|$. Therefore, we analyze the error probability of the decoding rule (3.20) which is associated with a particular cost function $R(\cdot, \mathbf{X})$ and a rate ρ . The maximal value of β under the condition of zero error communication is defined as *approximation capacity* since it determines the approximation precision of the coding scheme.

A communication error occurs if the sender selects τ_s and the receiver decodes $\hat{\tau} = \tau_j, j \neq s$. To estimate the probability of this event, we introduce the weight overlaps

$$\Delta Z_{\beta}^j := \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_{\beta}(c, \psi \circ \tau_j \circ \mathbf{X}') w_{\beta}(c, \tilde{\mathbf{X}}), \quad \tau_j \in \mathcal{T}. \quad (3.21)$$

The quantity ΔZ_{β}^j measures the number of hypotheses which have jointly low costs $R(c, \psi \circ \tau_j \circ \mathbf{X}')$ and $R(c, \tilde{\mathbf{X}})$.

The probability of a communication error is given by a substantial overlap ΔZ_{β}^j induced by $\tau_j \in \mathcal{T} \setminus \{\tau_s\}, 1 \leq j \leq M, j \neq s$, i.e.,

$$\begin{aligned} \mathbb{P}(\hat{\tau} \neq \tau_s \mid \tau_s) &= \mathbb{P}\left(\max_{\substack{1 \leq j \leq M \\ j \neq s}} \Delta Z_{\beta}^j \geq \Delta Z_{\beta}^s \mid \tau_s\right) \\ &\stackrel{(a)}{\leq} \sum_{\substack{1 \leq j \leq M \\ j \neq s}} \mathbb{P}(\Delta Z_{\beta}^j \geq \Delta Z_{\beta}^s \mid \tau_s) \\ &\stackrel{(b)}{\leq} \sum_{\substack{1 \leq j \leq M \\ j \neq s}} \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \mathbb{E}_{\mathcal{T} \setminus \{\tau_s\}} \left[\frac{\Delta Z_{\beta}^j}{\Delta Z_{\beta}^s} \mid \tau_s \right] \\ &\stackrel{(c)}{=} (M-1) \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \left[\frac{\mathbb{E}_{\tau_j: j \neq s} [\Delta Z_{\beta}^j \mid \mathbf{X}', \mathbf{X}'']}{\Delta Z_{\beta}^s} \mid \tau_s \right]. \end{aligned} \quad (3.22)$$

The expectation $\mathbb{E}_{\mathcal{T} \setminus \{\tau_s\}}$ is calculated w.r.t. the set of random transformations $\tau_j, 1 \leq j \leq M, j \neq s$ where we have conditioned on the sender selected transformation τ_s . The joint probability distribution of all transformations $\mathbb{P}(\mathcal{T}) = \prod_{j=1}^M P(\tau_j)$ decomposes into product form since all transformations are randomly drawn from the set of all possible transformations $\{\tau_j\}$. It corresponds to the Shannon's random codebook design in information theory.

The inequality (a) results from the union bound, and (b) is due to Markov's inequality. The identity (c) exploits the fact that the transformations $\tau \in \mathcal{T}$ are i.i.d. drawn according to the product measure $\mathbb{P}(\mathcal{T}) = \prod_{j \leq M} P(\tau_j)$.

The expected overlap $\mathbb{E}_{\tau_j} \Delta Z_{\beta}^j, j \neq s$ with any other message $\tau_j, j \neq s$ for given training data \mathbf{X}' and test data \mathbf{X}'' conditioned on τ_s is defined by

$$\begin{aligned} &\mathbb{E}_{\tau_j: j \neq s} [\Delta Z_{\beta}^j \mid \mathbf{X}', \mathbf{X}''] \\ &= \sum_{\tau_j \in \mathbb{T}} P(\tau_j) \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_{\beta}(c, \psi \circ \tau_j \circ \mathbf{X}') w_{\beta}(c, \tilde{\mathbf{X}}) \\ &= \sum_{\tau_j \in T_{\epsilon}^{(n)}} P(\tau_j) \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_{\beta}(c, \psi \circ \tau_j \circ \mathbf{X}') w_{\beta}(c, \tilde{\mathbf{X}}) + \text{term for } \{\tau_j \notin T_{\epsilon}^{(n)}\} \end{aligned}$$

$$\begin{aligned}
& \stackrel{(d)}{\leq} \sum_{c \in \mathcal{C}(\mathbf{O}'')} w_\beta(c, \tilde{\mathbf{X}}) \exp(-\log |\mathbb{T}|(\mathcal{H}(\tau) - \epsilon)) \underbrace{\sum_{\tau_j \in \mathbb{T}} w_\beta(\tau_j^{-1} \circ \psi^{-1} \circ c, \mathbf{X}')}_{\leq Z_\beta(\mathbf{X}')} \\
& \stackrel{(e)}{\leq} \exp(-\log |\mathbb{T}|(\mathcal{H}(\tau) - \epsilon)) Z_\beta(\mathbf{X}') Z_\beta(\mathbf{X}''). \tag{3.23}
\end{aligned}$$

The inequality (d) results from the typicality of $P(\tau_j)$. The terms for atypical transformations $\tau_j \notin T_\epsilon^{(n)}$ are neglected since the probability $P(\tau_j)$ converges to the entropy $\mathcal{H}(\tau)$. The last inequality (e) holds since the set $\{\tau_j^{-1} \circ \psi^{-1} \circ c : c \in \mathcal{C}(\mathbf{O}''), \tau_j \in \mathbb{T}\} \subset \mathcal{C}(\mathbf{O}')$ and extending the sum $\sum_{\tau_j \in \mathbb{T}}$ to $\sum_{c \in \mathcal{C}(\mathbf{O}')}$ only adds positive terms. Effectively, the sum over a random transformation τ_j decouples the two sums into a product of weight sums. The expectation over the data $\mathbf{X}', \mathbf{X}''$ in Eq. (3.22) yields

$$\begin{aligned}
& \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \frac{Z_\beta(\mathbf{X}') Z_\beta(\mathbf{X}'')}{\Delta Z_\beta^s} \\
& \stackrel{(f)}{=} \mathbb{E}_{(\mathbf{X}', \mathbf{X}'')} \mathbb{I}\{(\mathbf{X}', \mathbf{X}'') \in A_\epsilon^{(n)}\} \frac{Z'_\beta Z''_\beta}{\Delta Z_\beta^s} + \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \mathbb{I}\{(\mathbf{X}', \mathbf{X}'') \notin A_\epsilon^{(n)}\} \frac{Z'_\beta Z''_\beta}{\Delta Z_\beta^s} \\
& = \exp(-\mathcal{F}' \log |\mathcal{C}(\mathbf{O}')| - (\mathcal{F}'' - \Delta \mathcal{F}) \log |\mathcal{C}(\mathbf{O}'')| \\
& \quad + \epsilon (\log |\mathbb{T}| + \log |\mathcal{C}'| + 2 \log |\mathcal{C}''|)) + \text{term for } \{(\mathbf{X}', \mathbf{X}'') \notin A_\epsilon^{(n)}\} \tag{3.24}
\end{aligned}$$

with the abbreviation $Z'_\beta = Z_\beta(\mathbf{X}')$, $Z''_\beta = Z_\beta(\mathbf{X}'')$. The equality (f) for the expectation $\mathbb{E}_{\mathbf{X}', \mathbf{X}''}$ is split into typical contributions $(\mathbf{X}', \mathbf{X}'') \in A_\epsilon^{(n)}$ and negligible terms for atypical measurements $(\mathbf{X}', \mathbf{X}'') \notin A_\epsilon^{(n)}$. The term proportional to ϵ can be neglected since it becomes arbitrarily small in the limit $\lim_{n \rightarrow \infty}$ due to the assumed⁵ asymptotic equipartition property (3.13)–(3.15).

Inserting result (3.24) into equation (3.22) yields

$$\begin{aligned}
& \mathbb{P}(\hat{\tau} \neq \tau_s | \tau_s) \\
& \leq \exp(\log |\mathbb{T}|(\rho - \mathcal{H}(\tau)) - \log |\mathcal{C}'| \mathcal{F}' - \log |\mathcal{C}''| (\mathcal{F}'' - \Delta \mathcal{F})), \tag{3.25}
\end{aligned}$$

where we have introduced the rate definition $\rho = \log M / \log |\mathbb{T}|$. Often, the assumption $|\mathcal{C}(\mathbf{O}')| = |\mathcal{C}(\mathbf{O}'')| = |\mathbb{T}|$ is justified and the bound (3.25) simplifies to

$$\begin{aligned}
& \mathbb{P}(\hat{\tau} \neq \tau_s | \tau_s) \leq \exp(-\log |\mathcal{C}|(\mathcal{I}_\beta - \rho - 4\epsilon)) \\
& \quad \text{with } \mathcal{I}_\beta := \mathcal{H}(\tau) + \mathcal{F}' + \mathcal{F}'' - \Delta \mathcal{F}. \tag{3.26}
\end{aligned}$$

The quantity \mathcal{I}_β plays the role of the mutual information in communication. Error free communication requires $\rho < \mathcal{I}_\beta$, i.e., the rate ρ should not exceed $\mathcal{H}(\tau) + \mathcal{F}' + \mathcal{F}'' - \Delta \mathcal{F}$.

⁵Please note that AEP has to be proved for a selected cost function R .

How can this upper bound (3.26) with the quantity \mathcal{J}_β be interpreted? A close look at equation (3.25) reveals that the bound depends on the term

$$\begin{aligned} & \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \frac{Z_\beta(\mathbf{X}') Z_\beta(\mathbf{X}'')}{|\mathbb{T}| \Delta Z_\beta^s} \\ &= \mathbb{E}_{\mathbf{X}', \mathbf{X}''} \exp \left(-\log \frac{|\mathbb{T}|}{Z'_\beta} - \log \frac{|\mathcal{C}''|}{Z''_\beta} + \log \frac{|\mathcal{C}''|}{\Delta Z_\beta^s} \right). \end{aligned} \quad (3.27)$$

The term $\log(|\mathbb{T}|/Z'_\beta)$ counts the number of ways you can form statistically distinguishable subset of the complete transformation class \mathbb{T} , the second term $\log(|\mathcal{C}''|/Z''_\beta)$ measures the same property on the receiver side, and the last term $\log(|\mathcal{C}''|/\Delta Z_\beta^s)$ accounts for double counting of the overlap. The three terms together define a mutual information between the selected message τ_s and the reconstructed message $\hat{\tau}$.

3.5 Information-Theoretical Model Selection

The analysis of the error probability suggests the following inference principle for controlling the appropriate regularization strengths which implements a form of model selection: The approximation precision is controlled by β which has to be maximized to derive more precise solutions or patterns. For small β the rate ρ will be low since we resolve the space of solutions only in a coarse grained fashion. For too large β the error probability does not vanish which indicates confusions between τ_j , $j \neq s$ and τ_s . The optimal β -value is given by the largest β or, equivalently, the highest approximation precision

$$\beta^* \in \arg \max_{\beta \in [0, \infty)} \mathcal{J}_\beta(\tau_s, \hat{\tau}). \quad (3.28)$$

Another choice to be made in modeling is to select a suitable cost function $R(\cdot, \mathbf{X})$ for the pattern recognition problems at hand. Let us assume that a number of cost functions $\{R_\theta(\cdot, \mathbf{X}), \theta \in \Theta\}$ are considered as candidates. The approximation capacity $\mathcal{J}_\beta(\tau_s, \hat{\tau} | R_\theta)$ depends on the cost function through the Gibbs weights. Therefore, we can rank the different models according to their $\mathcal{J}_\beta(\tau_s, \hat{\tau} | R_\theta)$ values. Robust and informative cost functions yield a higher approximation capacity than simpler or more brittle models. A rational choice is to select the cost function

$$R_{\theta^*}(c, \mathbf{X}) \quad \text{with } \theta^* \in \arg \max_{\theta \in \Theta} \mathcal{J}_\beta(\tau_s, \hat{\tau} | R_\theta), \quad (3.29)$$

where both the random variables τ_s and $\hat{\tau}$ depend on $R_\theta(c, \mathbf{X})$, $\theta \in \Theta$. The selection rule (3.29) prefers the model which is “expressive” enough to exhibit high information content (e.g., many clusters in clustering) and, at the same time, robustly resists to noise in the data set. The bits or nats which are measured in the ASC communication setting are context sensitive since they refer to a hypothesis class $\mathcal{C}(\mathbf{X})$, i.e., how finely or coarsely hypotheses can be resolved in \mathcal{C} .

3.6 Minimizing Hamming Distance

To demonstrate the approach to regularized optimization we will apply it to an almost trivial optimization problem, i.e., minimizing the Hamming distance to a reference bit string $\xi' = (\xi'_1, \xi'_2, \dots, \xi'_n) \in \{-1, 1\}^n$ of n bits.⁶ This optimization problem describes the decoding step in classical communication theory. The cost function for communication measures the difference between a bit string $s \in \{-1, 1\}^n$ and a reference codeword ξ' , i.e.,

$$R(s, \xi') = \sum_{i=1}^n \mathbb{I}\{s_i \neq \xi'_i\} = \frac{1}{2} \left(n - \sum_{i=1}^n s_i \xi'_i \right). \quad (3.30)$$

The variable s has to be optimized and the empirical minimum is $s = \xi'$. However, ξ' is exposed to channel noise and, in the spirit of approximation set coding, we should only approximate it. The weights of approximate solutions are defined by

$$\mathcal{W}_\beta(\xi') = \left\{ w_\beta(s, \xi') = \exp\left(-\frac{\beta}{2} \left(n - \sum_{1 \leq i \leq n} s_i \xi'_i \right) \right) \right\}. \quad (3.31)$$

The sender uses this measurement ξ' and permutes the bits according to one of the randomly selected transformations $\mathcal{T} := \{\tau_1, \dots, \tau_{2M}\}$. Permutations which leave ξ' invariant are excluded. This set of randomly selected transformations generates a codebook with code vectors $\{\tau_1 \circ \xi', \dots, \tau_{2M} \circ \xi'\}$.

During communication, a second bit string ξ'' is generated by the problem generator. The receiver then receives the message $\tilde{\xi} = \tau_s \circ \xi''$ when the sender decides to communicate with transformation τ_s . This process defines the approximation problem $R(s, \tilde{\xi}) = \frac{1}{2}(n - s \cdot \tilde{\xi})$ on the receiver side. Based on the data $\tilde{\xi}$, the receiver has to estimate the transformation τ_s which has been communicated by the sender.

Let us assume that the probability $\delta := \mathbb{P}(\xi'_i \neq \xi''_i)$ characterizes the communication channel. Therefore, a fraction $\hat{\delta}n$ bits are different between the first bit sequence ξ' and the second bit sequence ξ'' , i.e., $\hat{\delta} = \frac{1}{n} |\{i : \xi'_i \neq \xi''_i\}|$.

The weight sums $Z_\beta(\xi)$, $\xi \in \{\xi', \xi''\}$ are given by

$$\begin{aligned} Z_\beta(\xi) &= \sum_{s \in \mathcal{C}(\xi)} \exp\left(-\frac{\beta}{2} \left(n - \sum_{i \leq n} s_i \xi_i \right) \right) \\ &= \exp\left(-\beta \frac{n}{2}\right) \prod_{i \leq n} \sum_{s_i \in \{-1, 1\}} \exp\left(\frac{\beta}{2} s_i \xi_i\right) \\ &= \exp\left(-\frac{n\beta}{2}\right) 2^n \left(\cosh \frac{\beta}{2} \right)^n. \end{aligned} \quad (3.32)$$

⁶W.l.o.g. we use the symmetric encoding $\{-1, 1\}$ rather than $\{0, 1\}$ to simplify the calculations.

The number of jointly approximating bit strings is determined by

$$\begin{aligned}
\Delta Z_\beta &= \sum_{s \in \mathcal{C}(\xi'')} \exp\left(-\beta\left(n - \frac{1}{2} \sum_{i \leq n} s_i(\xi'_i + \xi''_i)\right)\right) \\
&= \exp(-\beta n) \prod_{i \leq n} \left(\exp\left(\frac{\beta}{2}(\xi'_i + \xi''_i)\right) + \exp\left(-\frac{\beta}{2}(\xi'_i + \xi''_i)\right)\right) \\
&= \exp(-\beta n) 2^n (\cosh \beta)^{n(1-\delta)}. \tag{3.33}
\end{aligned}$$

The mutual information (3.27) for the special case of minimizing Hamming distances is determined by

$$\begin{aligned}
\mathcal{I}_\beta &= \mathcal{H}(\tau) + \mathcal{F}' + \mathcal{F}'' - \Delta \mathcal{F} \\
&= \ln 2 - \lim_{n \rightarrow \infty} \frac{1}{n} (\ln Z_\beta(\xi') + \ln Z_\beta(\xi'') - \ln \Delta Z_\beta) \\
&= (1 - \delta) \ln \cosh \beta - 2 \ln \cosh \frac{\beta}{2} \\
&= \ln 2 + (1 - \delta) \ln \cosh \beta - \ln(\cosh \beta + 1), \tag{3.34}
\end{aligned}$$

where we have estimated the size of the set of possible random transformations as $|\mathbb{T}| = 2^n$. In the case of a biased sequence with $\pi := \mathbb{P}(\xi'_i = 1) \neq 1/2$, the cardinality of the transformation set is $|\mathbb{T}| = 2^{\mathcal{H}(\pi)}$ with the binary entropy $\mathcal{H}(\pi) = -\pi \log_2 \pi - (1 - \pi) \log_2 (1 - \pi)$.

The optimal value for β is determined by the maximum of \mathcal{I}_β , i.e.,

$$\frac{d \mathcal{I}_\beta}{d \beta} = (1 - \delta) \frac{\sinh \beta}{\cosh \beta} - \frac{\sinh \beta}{\cosh \beta + 1} = 0 \tag{3.35}$$

$$\Rightarrow \cosh \beta = \frac{1 - \delta}{\delta}, \quad \cosh \beta + 1 = \delta^{-1}. \tag{3.36}$$

Inserting these values into Eq. (3.34) yields

$$\begin{aligned}
\mathcal{I}_\beta &= \ln 2 + (1 - \delta) \ln \frac{1 - \delta}{\delta} - \ln \frac{1}{\delta} \\
&= \ln 2 + (1 - \delta) \ln(1 - \delta) + \delta \ln \delta \\
&= \ln 2 - \mathcal{H}(\delta). \tag{3.37}
\end{aligned}$$

Equation (3.37) shows that optimally approximating the Hamming distance of bit strings by approximation set coding yields the channel capacity of the binary symmetric channel with bit error probability δ .

The expected generalization error, when the bit string s is drawn from the Gibbs distribution $p(s | \xi')$, is given by

$$\begin{aligned} \mathbb{E}_{\xi', \xi''} R(\xi', \xi'') &= \mathbb{E}_{\xi', \xi''} \sum_{s \in \mathcal{C}(\xi)} p(s | \xi') \frac{1}{2} \left(n - \sum_{i=1}^n s_i \xi_i' \right) \\ &= \frac{n}{2} \left(1 - (1 - 2\delta) \tanh \frac{\beta}{2} \right) \\ \text{with } p(s | \xi') &= \prod_{i \leq n} \frac{\exp(\beta s_i \xi_i' / 2)}{2 \cosh(\beta / 2)}. \end{aligned} \quad (3.38)$$

It is worth noting that the resolution of the model with minimal expected generalization error is achieved by $\lim_{\beta \rightarrow \infty} \mathbb{E}_{\xi', \xi''} R(\xi', \xi'') = \frac{n}{2} \delta$, i.e., by the empirical risk minimizer. The information theoretically optimal solution with $\beta^* = \operatorname{arccosh} \frac{1-\delta}{\delta}$ defines a lower resolution of the hypothesis class than the optimal generalization error would suggest with $\beta \rightarrow \infty$.

3.7 Discussion and Conclusion

Pattern analysis explores the questions how similar different patterns are and how we should compare them. The underlying topology and metric of a hypothesis class are often chosen ad hoc in applications and usually do not reflect properties of the data source, e.g., characteristics of a noise model. Approximation set coding as a model validation principle establishes a notion of *pattern equivalence* by considering them as statistically indistinguishable when the pattern differences cannot be exploited for coding. Patterns with the same or similar weights are considered to be equally acceptable solutions and these weights directly depend on the objective or cost function. To justify a natural topology and metric, we have to validate the underlying objective function for the pattern analysis problem. The reader should realize that the assumption of an objective function assumes a lot of information about the hypothesis class, it essentially establishes a partial order of all hypothesis.

Model selection and validation requires to estimate the generalization ability of models from training to test data. “Good” models show a high expressiveness and they are robust w.r.t. noise in the data. This tradeoff between *informativeness* and *robustness* ranks different models when they are tested on new data and it quantitatively describes the underfitting/overfitting dilemma. In this chapter, we have explored the idea to use approximation sets of clustering solutions as a communication code. The *approximation capacity* of a cost function provides a selection criterion which renders various models comparable in terms of their respective bit rates. The number of reliably extractable bits of a pattern analysis cost function $R(\cdot, \mathbf{X})$ defines a “task sensitive information measure” since it only accounts for the fluctuations in the data \mathbf{X} which actually have an influence on identifying an individual pattern or a set of patterns.

The maximum entropy inference principle suggests that we should average over the statistically indistinguishable solutions in the optimal approximation set. Such a model averaging strategy replaces the original cost function with the free energy and, thereby, it defines a continuation methods with maximal robustness. Algorithmically, maximum entropy inference can be implemented by annealing methods [3, 15, 17]. The urgent question in many data analysis applications, which regularization term should be used without introducing an unwanted bias, is naturally answered by the entropy. The second question, how the regularization parameter should be selected, is answered by ASC: Choose the parameter value which maximizes the approximation capacity! The link to robust optimization is analyzed from a theoretical computer science viewpoint in [5].

ASC for model selection can be applied to all combinatorial or continuous optimization problems which depend on noisy data. The noise level is characterized by two sample sets \mathbf{X}' , \mathbf{X}'' . ASC has been empirically explored by model validation problems for model based clustering [4] of high dimensional Gaussian distributed data and of Boolean data. The well-known spin glass phase of maximum likelihood estimations for Gaussian sources is identified as a structure with zero information content for coding. ASC can also be used to select models for spectral clustering [8]. For a correlation matrix of gene expression data gathered from the mussel *Mytilus Galloprovincialis*, pairwise clustering produced a more informative clustering than both normalized cut and correlation clustering. In a similar spirit, Han et al. [14] used ASC to cluster graphs and to control the selection of clusters and prototypes. Furthermore, denoising of Boolean matrices guided by the generalization capacity of SVD suggests a cutoff rank for the SVD spectrum [11].

One fundamental question for computer science remains unanswered so far:

How can we validate algorithms?

The reader should realize that we only require an objective $R(\cdot, \mathbf{X})$ to define a weight distribution. Any other mechanism to arrive at such a concept of approximate solutions will serve the same purpose. In a recent PhD thesis [6], Ludwig Busse has explored approximation set coding to measure the sensitivity of sorting algorithms to erroneous computations in pairwise comparisons of items. Various sorting algorithms like MergeSort, SelectionSort, BubbleSort, InsertionSort, QuickSort show different sensitivities to errors in the comparison subroutine. Each algorithm is then characterized by a capacity [7] which specifies a bit rate of extracted information per computation step, e.g., per comparison in the case of sorting. The study clearly demonstrates that robust algorithms like BubbleSort invest their excess comparisons in averaging to compensate for fluctuations. This computational redundancy increases the capacity of the algorithm and yields an improved localization ability in the hypothesis class. Computationally efficient methods like MergeSort perform superior in the noiseless case but sacrifice capacity for computational speed in the highly noisy case.

In principle, this concept of measuring the generalization performance of algorithms can be applied to algorithm evaluation and also to robust algorithm design. It endows the space of algorithm with a topology since two algorithms are neighbors if their approximation sets for the same input distributions share a high overlap. Such

methods to measure the robustness of algorithms to errors in the computation or in the input will be in high demand to program novel hardware that trades energy consumption against precision of computation [16]. So far we are completely lacking design principles for algorithm engineering which consider this tradeoff between energy usage and correctness. We are also convinced that the information theoretic analysis of algorithms will shed new light on the relation between computational complexity and statistical complexity—the two faces of complexity science whose relation is far from being understood.

Acknowledgement This work has been partially supported by the FP7 EU project SIMBAD and by the SNF project 200021_138117. JB acknowledges very stimulating discussions with A. Busetto, L. Busse, M.H. Chehreghani, M. Frank, M. Mihalák, V. Roth, R. Sránek, W. Szpankowski and P. Widmayer.

References

1. Alon, N., Ben-David, S., Cesa-Bianchi, N., Haussler, D.: Scale-sensitive dimensions, uniform convergence, and learnability. *J. ACM* **44**(4), 615–631 (1997)
2. Buhmann, J.M.: Information theoretic model validation for clustering. In: International Symposium on Information Theory, Austin Texas. IEEE Press, New York (2010). <http://arxiv.org/abs/1006.0375>
3. Buhmann, J.M., Kühnel, H.: Vector quantization with complexity costs. *IEEE Trans. Inf. Theory* **39**(4), 1133–1145 (1993)
4. Buhmann, J.M., Chehreghani, M.H., Frank, M., Streich, A.P.: Information theoretic model selection for pattern analysis. In: Guyon, I., Dror, G., Lemaire, V., Taylor, G., Silver, D. (eds.) *ICML 2011 Workshop on “Unsupervised and Transfer Learning”*, Bellevue, Washington, vol. 27, pp. 51–65 (2012). Clearwater Beach, Florida, *JMLR: W&CP* 5
5. Buhmann, J.M., Mihalák, M., Sránek, R., Widmayer, P.: Robust optimization in the presence of uncertainty. In: *Inventions in Theoretical Computer Science 2013*, Berkeley. ACM 2013, pp. 505–514 (2012). doi:[10.1145/2422436.2422491](https://doi.org/10.1145/2422436.2422491)
6. Busse, L.: Information in orderings (learning to order). Ph.D. thesis, # 20600, ETH Zurich, CH-8092 Zurich, Rämistrasse (2012)
7. Busse, L.M., Chehreghani, M.H., Buhmann, J.M.: The information content in sorting algorithms. In: International Symposium on Information Theory, pp. 2746–2750. IEEE Press, Cambridge (2012)
8. Chehreghani, M.H., Giovanni Busetto, A., Buhmann, J.M.: Information theoretic model validation for spectral clustering. In: *AISTATS 2012*, La Palma. *J. Mach. Learn. Res. (W&CP)*, vol. 22, pp. 495–503 (2012)
9. Cover, T.M., Thomas, J.A.: *Elements of Information Theory*, 2nd edn. Wiley, New York (1991)
10. Csiszár, I., Körner, J.: *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, New York (1981)
11. Frank, M., Buhmann, J.M.: Selecting the rank of SVD by maximum approximation capacity. In: International Symposium on Information Theory, St. Petersburg, pp. 1036–1040. IEEE Press, New York (2011)
12. Grenander, U.: *General Pattern Theory: a Mathematical Study of Regular Structures*. Oxford University Press, Oxford (1994)
13. Grenander, U., Miller, M.I.: *Pattern Theory: from Representation to Inference*. Oxford University Press, Oxford (2007)

14. Han, L., Rossi, L., Torsello, A., Wilson, R.C., Hancock, E.R.: Information theoretic prototype selection for unattributed graphs. In: Gimel'farb, G.L., Hancock, E.R., Imiya, A., Kuijper, A., Kudo, M., Omachi, S., Winderatt, T., Yamada, K. (eds.) *Structural, Syntactic, and Statistical Pattern Recognition*. Lecture Notes in Computer Science, vol. 7626, pp. 33–41. Springer, Berlin (2012)
15. Hofmann, T., Buhmann, J.M.: Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(1), 1–14 (1997)
16. Lingamneni, A., Krishna Muntimadugu, K., Enz, C., Karp, R.M., Palem, K.V., Piguët, C.: Algorithmic methodologies for ultra-efficient inexact architectures for sustaining technology scaling. In: *Proceedings of the 9th Conference on Computing Frontiers, CF'12*, pp. 3–12. ACM, New York (2012)
17. Rose, K., Gurewitz, E., Fox, G.: Vector quantization by deterministic annealing. *IEEE Trans. Inf. Theory* **38**(4), 1249–1257 (1992)
18. Vapnik, V.N.: *Estimation of Dependencies Based on Empirical Data*. Springer, New York (1982)
19. Vapnik, V.N., Chervonenkis, A.Ya.: On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16**, 264–280 (1971)

Part II
Deriving Similarities for Non-vectorial
Data

Chapter 4

On the Combination of Information-Theoretic Kernels with Generative Embeddings

Pedro M.Q. Aguiar, Manuele Bicego, Umberto Castellani, Mário A.T. Figueiredo, André T. Martins, Vittorio Murino, Alessandro Perina, and Aydın Ulaş

Abstract Classical methods to obtain classifiers for structured objects (e.g., sequences, images) are based on generative models and adopt a classical generative Bayesian framework. To embrace discriminative approaches (namely, support vector machines), the objects have to be mapped/embedded onto a Hilbert space; one way that has been proposed to carry out such an embedding is via generative models (maybe learned from data). This type of hybrid discriminative/generative approach has been recently shown to outperform classifiers obtained directly from the generative model upon which the embedding is built.

Discriminative approaches based on generative embeddings involve two key components: a generative model used to define the embedding; a discriminative

P.M.Q. Aguiar

Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa, Portugal

e-mail: aguiar@isr.ist.utl.pt

M. Bicego · U. Castellani · V. Murino · A. Ulaş

Dipartimento di Informatica, University of Verona, Verona, Italy

M. Bicego

e-mail: manuele.bicego@univr.it

U. Castellani

e-mail: umberto.castellani@univr.it

V. Murino

e-mail: vittorio.murino@univr.it

A. Ulaş

e-mail: mehmetaydin.ulas@univr.it

M.A.T. Figueiredo (✉) · A.T. Martins

Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal

e-mail: mario.figueiredo@lx.it.pt

A.T. Martins

e-mail: andre.t.martins@gmail.com

A. Perina

Microsoft Research, Redmond, WA, USA

e-mail: alperina@microsoft.com

learning algorithms to obtain a (maybe kernel) classifier. The literature on generative embedding is essentially focused on defining the embedding, and some standard off-the-shelf kernel and learning algorithm are usually adopted. Recently, we have proposed a different approach that exploits the probabilistic nature of generative embeddings, by using information-theoretic kernels defined on probability distributions. In this chapter, we review this approach and its building blocks. We illustrate the performance of this approach on two medical applications.

4.1 Introduction

Most approaches to the statistical learning of classifiers belong to one of two classical paradigms: *generative* and *discriminative* [2, 53], also known in the statistics literature as *sampling* and *diagnostic*, respectively [3]. Generative approaches are based on probabilistic class models and *a priori* class probabilities, learnt from training data and combined via Bayes law to yield posterior probability estimates. Discriminative methods aim at learning class boundaries or posterior class probability estimates directly from data, without the intermediate step of learning generative class models.

Discriminative approaches received a formidable boost with the introduction of kernel-based methods, namely the support vector machine (SVM), in the early 1990s [4]. Kernels had a great impact in machine learning and were used for many learning tasks besides classification, including regression, principal component analysis, clustering, and many others [5, 6]. Their great popularity derives mainly from two facts: (i) kernels extend linear methods (e.g., classifiers) that depend only on inner products between pairs of objects to the nonlinear realm, by replacing each inner product by a kernel evaluation; (ii) kernels considerably widen the applicability of many learning algorithms from the classical vector spaces to a much wider range of sets of objects (images, sequences, trees, functions, probability distributions) [6].

In the past decade, several hybrid generative–discriminative approaches have been proposed with the goal of combining the best of both paradigms [8, 39]. In a nutshell, the idea is to take into account, when defining/building a kernel, any available knowledge/model about how objects are generated. In this context, the so-called *generative embeddings* (or generative score space methods) have exploited generative models to map the objects to be classified into a feature space, where discriminative techniques (e.g., kernel-based SVMs) can be used. This is particularly well suited for dealing with non-vectorial data, since it maps objects that may have different dimensions (e.g., strings of different lengths) into a (fixed, maybe infinite-dimensional) Hilbert space.

The seminal work on generative embeddings is due to Jaakola and Haussler [39], where the so-called *Fisher score* was introduced. In that work, the features of a given object are the derivatives of the log-likelihood function under the assumed generative model, with respect to the model parameters, computed at that object. Other examples of generative embeddings can be found in the work of Bicego et al. [9],

Bosch et al. [12], and Perina et al. [57]. In this chapter, and following recent work on generative embeddings, we focus on the use of the so-called pLSA (*probabilistic latent semantic analysis*) as a generative model, the usefulness of which has been recently shown in several applications [12–14, 56].

Typically, the vectorial representations resulting from the generative embedding are used with some standard kernel-based classifier with a simple linear or radial basis function (RBF) kernel. We have recently proposed an alternative route [15–17]: instead of relying on standard kernels, we use the information-theoretic (IT) kernels introduced by Martins et al. [52] as a similarity measure between objects in the generative embedding space. The main idea is that the IT kernels are well suited to the probabilistic nature of the generative embeddings, thus expected to improve the performance of hybrid approaches. The kernels proposed by Martins et al. [52] extend and subsume previous IT kernels based on Shannon’s information theory [19, 21, 24], by adopting a non-extensive version of information theory, and are defined on both unnormalized or normalized (i.e., probability) measures. Those kernels were successfully used in text categorization tasks, based on multinomial text representations (e.g., bags-of-words, character n -grams) [52].

We illustrate the performance of the proposed methodology on two different medical applications: colon cancer detection from gene expression data and renal cell cancer classification from tissue microarray data. The experimental results testify for the adequacy and state-of-the-art performance of the combination of IT kernels with generative embeddings.

The remaining sections of this chapter are organized as follows. In Sect. 4.2, the fundamental ideas of generative embeddings are reviewed, together with the basics of the schemes herein investigated. Section 4.3 reviews the IT kernels to be used in combination with the generative embeddings. The proposed way of using the IT kernels with the generative embeddings is formalized in Sect. 4.4. Details on applications and experimental results are reported in Sect. 4.5, and Sect. 4.6 concludes the chapter.

4.2 Generative Embeddings

The underlying motivation for pursuing principled hybrid discriminative–generative classifiers is their clear complementarity and the fact that, asymptotically, the classification error of discriminative methods is lower than that of generative ones [53]. On the other hand, generative methods are effective in handling scarce data and allow for an easier handling of missing data and inclusion of prior knowledge about the data. Among hybrid generative–discriminative methods, “generative embeddings” (also called generative score spaces) have seen considerable interest in recent years, as is testified by an increasing literature on this class of methods [11, 12, 39, 49, 56, 71, 79].

Carrying out a generative embedding involve three steps: (i) a generative model (or a family thereof) is adopted and learned from the data; (ii) this learned model

is used to obtain a mapping between the set of original objects and a Hilbert space (often called the *score space*); (iii) the objects in the training set are mapped into the score space and fed into some discriminative learning technique. The key idea is to map objects (e.g., sequences, possibly with different lengths) into fixed-dimensional feature vectors, using a model of how these objects are generated. This opens the door to the use of discriminative learning techniques, such as SVMs or logistic regression [26], and has been shown to achieve higher classification accuracy than purely generative or discriminative approaches [56].

Once a generative embedding is obtained, in order to use a kernel-based discriminative learning approach, it is necessary to adopt a kernel expressing similarity between pairs of points in the score space, maybe also derived from the generative model. The most famous example is the *Fisher kernel* [39], which is simply a Riemannian inner product, using the inverse Fisher matrix of the generative model as the underlying metric. In this chapter, we will use kernels defined on the score space that are independent of the generative model.

The following sections review the generative embeddings considered in this chapter, and the pLSA generative model based on which they are built.

4.2.1 Probabilistic Latent Semantic Analysis (pLSA)

Consider a set of *documents*¹ $\mathcal{D} = \{d_1, \dots, d_{|\mathcal{D}|}\}$, each containing an arbitrary number of words, all taken from a vocabulary of $\mathcal{W} = \{w_1, \dots, w_{|\mathcal{W}|}\}$. Without loss of generality, we may simply refer to the documents and words by their indices, thus we simplify the notation by writing $\mathcal{D} = \{1, \dots, |\mathcal{D}|\}$ and $\mathcal{W} = \{1, \dots, |\mathcal{W}|\}$. The collection \mathcal{D} is summarized by a bag-of-words description (i.e., ignoring word order) into a $|\mathcal{W}| \times |\mathcal{D}|$ occurrence matrix $\mathbf{C} = [C_{ij}, i = 1, \dots, |\mathcal{W}|, j = 1, \dots, |\mathcal{D}|]$, where C_{ij} is the number of occurrences of the i th word in the j th document.

Introduced by Hofmann [38], pLSA is a generative mixture model for matrix \mathbf{C} , where the presence of each word in each document is mediated by a latent random variable, $Z \in \mathcal{Z} = \{1, \dots, |\mathcal{Z}|\}$ (known as the *topic* or *aspect* variable). More specifically, pLSA is a mixture model for the joint distribution of the pair of random variables $D \in \mathcal{D}$ and $W \in \mathcal{W}$, where the event $(W = i, D = j)$ means that there is an occurrence of the i th word in the j th document; pLSA expresses the joint probability distribution $\mathbb{P}(W = i, D = j)$ as a mixture of distributions such that, in each component of the mixture (i.e., for each *topic*), the random variables W and D are independent (i.e., $\mathbb{P}(W = i, D = j | Z = z) = \mathbb{P}(W = i | Z = z) \mathbb{P}(D = j | Z = z)$);

¹We use the term *document* to refer to a finite sequence of objects from some finite set, simply because LSA and pLSA have their roots in the field of natural language processing (NLP). Recently, pLSA has been used, not only in NLP, but in other areas, such as computer vision, bioinformatics, and image analysis [10, 12, 28]. In image analysis problems, the idea is to use pLSA to model the occurrence of image features (*visual words*) [12, 28].

formally,

$$\mathbb{P}(W = i, D = j) = \sum_{z=1}^{|\mathcal{Z}|} \mathbb{P}(Z = z) \mathbb{P}(W = i | Z = z) \mathbb{P}(D = j | Z = z). \quad (4.1)$$

The pLSA model is parameterized by a set of $1 + 2|\mathcal{Z}|$ multinomial distributions: the distribution of the latent topic variable ($\mathbb{P}(Z = 1), \dots, \mathbb{P}(Z = |\mathcal{Z}|)$) $\in \Delta_{|\mathcal{Z}|}$ (where Δ_K denotes the standard probability simplex in \mathbb{R}^K); the distributions of words ($\mathbb{P}(W = 1 | Z = z), \dots, \mathbb{P}(W = |\mathcal{W}| | Z = z)$) $\in \Delta_{|\mathcal{W}|}$, for each $z \in \{1, \dots, |\mathcal{Z}|\}$; and the distributions of documents ($\mathbb{P}(D = 1 | Z = z), \dots, \mathbb{P}(D = |\mathcal{D}| | Z = z)$) $\in \Delta_{|\mathcal{D}|}$, for each $z \in \{1, \dots, |\mathcal{Z}|\}$. Let us write these parameters compactly in a vector $\mathbf{p} = [p_1, \dots, p_{|\mathcal{Z}|}]$, where $p_z \equiv \mathbb{P}(Z = z)$ and a pair of matrices \mathbf{Q} and \mathbf{R} , where $Q_{zw} \equiv \mathbb{P}(W = w | Z = z)$ and $R_{zd} \equiv \mathbb{P}(D = d | Z = z)$. Of course, both \mathbf{Q} and \mathbf{R} are stochastic matrices: $Q_{zw} \geq 0$, $R_{zd} \geq 0$, $\sum_{w=1}^{|\mathcal{W}|} Q_{zw} = 1$, and $\sum_{d=1}^{|\mathcal{D}|} R_{zd} = 1$.

Given a collection of N independent samples $(w_1, d_1), \dots, (w_N, d_N)$ from this generative model, it is easy to show that the log-likelihood function (based on which the parameters \mathbf{p} , \mathbf{Q} , and \mathbf{R} can be estimated) can be written as

$$\begin{aligned} \mathcal{L}(\mathbf{p}, \mathbf{Q}, \mathbf{R}) &= \log \mathbb{P}((w_1, d_1), \dots, (w_N, d_N) | \mathbf{p}, \mathbf{Q}, \mathbf{R}) \\ &= \sum_{w=1}^{|\mathcal{W}|} \sum_{d=1}^{|\mathcal{D}|} C_{wd} \log \sum_{z=1}^{|\mathcal{Z}|} p_z Q_{zw} R_{zd}, \end{aligned} \quad (4.2)$$

where C_{wd} is the number of times the pair (w, d) co-occurs in the set of observations. This shows that matrix \mathbf{C} contains the sufficient statistics to estimate the parameters of the pLSA model. Of course, maximizing (4.2) with respect to \mathbf{p} , \mathbf{Q} , and \mathbf{R} cannot be done in closed form, but can be naturally addressed via the EM algorithm [38].

Given estimates of the model parameters, $\hat{\mathbf{p}}$, $\hat{\mathbf{Q}}$, and $\hat{\mathbf{R}}$, it is possible to estimate quantities such as the probability that a given topic is present in a given document:

$$\hat{\mathbb{P}}(Z = z | D = d) = \frac{\hat{\mathbb{P}}(D = d | Z = z) \hat{\mathbb{P}}(Z = z)}{\sum_{s=1}^{|\mathcal{Z}|} \hat{\mathbb{P}}(D = d | Z = s) \hat{\mathbb{P}}(Z = s)} = \frac{\hat{R}_{zd} \hat{p}_z}{\sum_{s=1}^{|\mathcal{Z}|} \hat{R}_{sd} \hat{p}_s}. \quad (4.3)$$

It is important to note that the random variable D takes values exclusively in the list of documents in the training set. For this reason, pLSA is not a full generative model of documents, in the sense that it has no way to assign a probability to a previously unseen document.

4.2.2 pLSA-Based Generative Embeddings

Generative embeddings based on latent variable models (such as pLSA) can be divided into two families: those based on the model parameters and those based on the latent/hidden variables.

4.2.2.1 Parameter-Based Generative Embeddings

In this section, we review three of the best-known generative embeddings based on the generative model parameters

- The *Fisher score* (FS), or Fisher embedding, was the first proposal of a generative embedding [39]; it consists in using as feature vector the tangent of the log-likelihood with respect to the model parameters. For the pLSA model [37], each document $d \in \{1, \dots, |\mathcal{D}|\}$ is mapped into the gradient of its log-probability w.r.t. the model parameters, which we collect into a vector $\theta \equiv (\mathbf{p}, \mathbf{Q}, \mathbf{R})$. The log-probability of a document $d \in \{1, \dots, |\mathcal{D}|\}$, denoted as $l(d | \theta)$, is obtained by marginalization,

$$l(d | \theta) = \log \sum_{w=1}^{|\mathcal{W}|} \mathbb{P}(W = w, D = d | \theta) = \log \sum_{w=1}^{|\mathcal{W}|} \sum_{z=1}^{|\mathcal{Z}|} p_z Q_{zw} R_{zd}. \quad (4.4)$$

The pLSA-based Fisher score maps each document d into a vector of dimension $|\mathcal{Z}| - 1 + |\mathcal{Z}|(|\mathcal{D}| + |\mathcal{W}| - 2)$ (the number of free parameters in the pLSA model), containing the derivatives of $l(d | \theta)$ w.r.t. to the elements of θ . In this score space, we may define the kernel simply as the Euclidean inner space. Alternatively (although we do not consider that choice here), the kernel could be defined as the Riemannian inner product, using the inverse Fisher matrix as the metric [39].

- The *TOP* (*Tangent Of Posterior log-odds*) embedding [79], was designed for two-class problems and is based on the gradient of the posterior log-odds ratio. Given parameter estimates of two pLSA models for two classes, $\theta^{(-1)}$ and $\theta^{(+1)}$, a given document d is mapped into the gradient of the posterior log-odds ratio $\log \mathbb{P}(C = +1 | d, \theta) - \log \mathbb{P}(C = -1 | d, \theta)$ w.r.t. $\theta = (\theta^{(-1)}, \theta^{(+1)})$.
- The LLR (*log-likelihood ratio*) embedding [72] is similar to the Fisher score, except that it uses one generative model per class, rather than a single model. Formally, for a C -class problem, the LLR embedding maps a given document d into the concatenation of the gradients of $\log \mathbb{P}(d | \theta^{(1)}), \dots, \log \mathbb{P}(d | \theta^{(C)})$ w.r.t. the respective parameters. Consequently, the dimensionality of the LLR embedding is C times that of the Fisher embedding.

4.2.2.2 Latent-Variable-Based Embeddings

This class of methods was proposed by Perina et al. [56] and is based on the hidden variables of the model, rather than on its parameters.

- The *free energy score space* (FESS) is based on the observation that the free energy bound on the complete log-likelihood decomposes into a sum of terms [56]; the mapping of a given document is the vector containing the terms in this decomposition. The details of the free energy bound and the resulting embedding are too long to include here, so the reader is referred to the work of Perina et al. [56] for a detailed description.
- The *posterior divergence* (PD) embedding is a modification of the FESS embedding [49], which also takes into account how much each sample affects the model. Details on the pLSA-based PD embedding and on its relationship with FESS case can be found in the work of Li, Lee, and Liu [49].
- The *mixture of topics* (MT) embedding simply maps a given document d into the $|\mathcal{Z}|$ -dimensional vector containing the conditional probabilities $\mathbb{P}(Z = 1 \mid D = d), \dots, \mathbb{P}(Z = |\mathcal{Z}| \mid D = d)$. Recall that these probabilities (given the parameter estimates) are computed according to (4.3).

4.2.2.3 Some Remarks

One obvious question is how to select the number of topics $|\mathcal{Z}|$. In our applications, we estimate this number using the well-known *Bayesian information criterion* (BIC) [67], which penalizes the likelihood with a term that depends on the number of model parameters. In the pLSA model, the number of free parameters is $|\mathcal{Z}| - 1 + |\mathcal{Z}|(|\mathcal{D}| + |\mathcal{W}| - 2)$. Thus, the number of topics is chosen as the minimizer w.r.t. $|\mathcal{Z}|$ of the penalized log-likelihood:

$$-\mathcal{L}(\mathbf{p}, \mathbf{Q}, \mathbf{R}) + [|\mathcal{Z}| - 1 + |\mathcal{Z}|(|\mathcal{D}| + |\mathcal{W}| - 2)] \log(\sqrt{N}).$$

In our experiments, we consider two versions of the FESS and MT embeddings. In the first version, we train one pLSA model per class and concatenate the resulting feature vectors (we will refer these as FESS-1 and MT-1); in the second version, we train a pLSA model for the whole data, ignoring the class label (we will refer these as FESS-2 and MT-2). In summary, we will consider eight different generative embeddings: MT-1, MT-2, FESS-1, FESS-2, LLR, FS, TOP, and PD.

4.3 Information-Theoretic Kernels

This section briefly reviews the information theoretic kernels proposed by Martins et al. [52] and introduces relevant notation. These kernels will be combined with the generative embeddings described in the previous section.

4.3.1 Positive Definite Kernels

We start by very briefly recalling the definition of positive definite (pd) kernel (for comprehensive accounts on kernel theory and methods, see, e.g., the books by

Schölkopf and Smola [5] and Shawe-Taylor and Cristianini [6]); in the following, X denotes a nonempty set.

Definition 4.1 Let $\varphi : X \times X \rightarrow \mathbb{R}$ be a symmetric function (i.e., satisfying $\varphi(y, x) = \varphi(x, y)$, for all $x, y \in X$). φ is called a *positive definite* (pd) *kernel* if and only if

$$\sum_{i=1}^n \sum_{j=1}^n c_i c_j \varphi(x_i, x_j) \geq 0$$

for all $n \in \mathbb{N}$, $x_1, \dots, x_n \in X$ and $c_1, \dots, c_n \in \mathbb{R}$.

The fact that pd kernels “correspond” to inner products, via embedding in a Hilbert space (as expressed by the next theorem), is at the heart of the use of kernels in machine learning.

Theorem 4.1 Let $\varphi : X \times X \rightarrow \mathbb{R}$ be a symmetric function. The function φ corresponds to an inner product in an Hilbert space \mathcal{H} , in the sense that

$$\varphi(x, y) = \langle \psi(x), \psi(y) \rangle,$$

where $\psi : X \rightarrow \mathcal{H}$ is the feature map (or Hilbert space embedding) and $\langle \cdot, \cdot \rangle$ denotes inner product, if and only if φ is a pd kernel.

4.3.2 Suyari’s Entropies

As proposed by Suyari [33], both the classical Shannon–Boltzmann–Gibbs (SBG) entropy [34] and the Tsallis entropy [35] are particular cases of functions $S_{q,\phi}$ that obey a certain set of axioms. Let Δ_n be the standard probability simplex in \mathbb{R}^n and $q \geq 0$ be a fixed quantity (the so-called *entropic index*). The function $S_{q,\phi} : \Delta_n \rightarrow \mathbb{R}$ has the form

$$S_{q,\phi}(p_1, \dots, p_n) = \begin{cases} \frac{k}{\phi(q)}(1 - \sum_{i=1}^n p_i^q) & \text{if } q \neq 1, \\ -k \sum_{i=1}^n p_i \ln p_i & \text{if } q = 1, \end{cases} \quad (4.5)$$

where $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a continuous function with certain properties [33], and $k > 0$ is an arbitrary constant, henceforth set to $k = 1$. As is clear in (4.5), for $q = 1$, we recover the SBG entropy, while setting $\phi(q) = q - 1$ yields the Tsallis entropy

$$S_q(p_1, \dots, p_n) = \frac{1}{q-1} \left(1 - \sum_{i=1}^n p_i^q \right) = - \sum_{i=1}^n p_i^q \ln_q p_i, \quad (4.6)$$

where

$$\ln_q(x) = (x^{1-q} - 1)/(1 - q)$$

is called the q -logarithm function.

4.3.3 Jensen–Shannon (JS) Divergence

Consider two measure spaces $(\mathcal{X}, \mathcal{M}, \nu)$, and $(\mathcal{T}, \mathcal{J}, \tau)$, where the second is used to index the first. Let H denote the SBG entropy, and consider the random variables $T \in \mathcal{T}$ and $X \in \mathcal{X}$, with densities $\pi(t)$ and $p(x) \triangleq \int_{\mathcal{T}} p(x | t)\pi(t)$. The Jensen divergence [52] is defined as

$$J^\pi(p) \triangleq J_H^\pi(p) = H(\mathbb{E}[p]) - \mathbb{E}[H(p)]. \quad (4.7)$$

When \mathcal{X} and \mathcal{T} are finite with $|\mathcal{T}| = m$, $J_H^\pi(p_1, \dots, p_m)$ is called the *Jensen–Shannon (JS) divergence* of p_1, \dots, p_m , with weights π_1, \dots, π_m [36, 37]. In particular, if $|\mathcal{T}| = 2$ and $\pi = (1/2, 1/2)$, p may be seen as a random distribution whose value in $\{p_1, p_2\}$ is chosen by tossing a fair coin. In this case, $J^{(1/2, 1/2)} = \text{JS}(p_1, p_2)$, where

$$\text{JS}(p_1, p_2) \triangleq H\left(\frac{p_1 + p_2}{2}\right) - \frac{H(p_1) + H(p_2)}{2},$$

which will be used in Sect. 4.3.5 to define JS kernels.

4.3.4 Jensen–Tsallis (JT) q -Differences

As is clear in (4.6), Tsallis' entropy can be written as $S_q(X) = -\mathbb{E}_q[\ln_q p(X)]$, where \mathbb{E}_q denotes the *unnormalized q -expectation*, which, for a discrete random variable $X \in \mathcal{X}$ with probability mass function $p : \mathcal{X} \rightarrow \mathbb{R}$, is defined as

$$\mathbb{E}_q[X] \triangleq \sum_{x \in \mathcal{X}} xp(x)^q;$$

(of course, $\mathbb{E}_1[X]$ is the standard expectation).

As in Sect. 4.3.3, consider two random variables $T \in \mathcal{T}$ and $X \in \mathcal{X}$, with densities $\pi(t)$ and $p(x) \triangleq \int_{\mathcal{T}} p(x | t)\pi(t)$. The Jensen q -difference is the nonextensive analogue of (4.7) [52],

$$T_q^\pi(p) = S_q(\mathbb{E}[p]) - \mathbb{E}_q[S_q(p)].$$

If \mathcal{X} and \mathcal{T} are finite with $|\mathcal{T}| = m$, $T_q^\pi(p_1, \dots, p_m)$ is called the *Jensen–Tsallis (JT) q -difference* of p_1, \dots, p_m , with weights π_1, \dots, π_m . In particular, if $|\mathcal{T}| = 2$ and $\pi = (1/2, 1/2)$, define $T_q = T_q^{1/2, 1/2}$ as

$$T_q(p_1, p_2) = S_q\left(\frac{p_1 + p_2}{2}\right) - \frac{S_q(p_1) + S_q(p_2)}{2},$$

which will be used in Sect. 4.3.5 to define the so-called JT kernels. Naturally, the JT 1-difference T_1 coincides with the JS divergence.

4.3.5 Jensen–Shannon and Jensen–Tsallis Kernels

The JS and JT differences underlie the kernels proposed by Martins et al. [52], which apply to normalized or unnormalized measures.

Definition 4.2 (Weighted Jensen–Tsallis (WJT) kernels) Let μ_1 and μ_2 be two (not necessarily probability) measures; the kernel \tilde{k}_q is defined as

$$\tilde{k}_q(\mu_1, \mu_2) \triangleq (S_q(\pi) - T_q^\pi(p_1, p_2))(\omega_1 + \omega_2)^q,$$

where $p_1 = \frac{\mu_1}{\omega_1}$ and $p_2 = \frac{\mu_2}{\omega_2}$ are the normalized counterparts of μ_1 and μ_2 (which have total masses ω_1 and ω_2 , respectively), and $\pi = (\omega_1 + \omega_2)^{-1}[\omega_1, \omega_2]$. The kernel k_q is defined as

$$k_q(\mu_1, \mu_2) \triangleq S_q(\pi) - T_q^\pi(p_1, p_2).$$

Notice that if $\omega_1 = \omega_2$, \tilde{k}_q and k_q coincide up to a scale factor. In the particular case of $q = 1$, k_1 is the so-called Jensen–Shannon kernel, $k_{JS}(p_1, p_2) = \ln 2 - JS(p_1, p_2)$.

The following proposition (proved by Martins et al. [52]) characterizes the positive definiteness these kernels.

Proposition 4.1 *The kernel \tilde{k}_q is positive definite, for $q \in [0, 2]$. The kernel k_q is positive definite, for $q \in [0, 1]$. The kernel k_{JS} is pd.*

4.4 Proposed Approach

The approach herein proposed consists in defining a kernel between two observed objects x and y as the composition of a generative embedding with one of the information theoretic kernels presented in the previous section. Formally,

$$k(x, y) = k_q(\phi(x), \phi(y)), \quad (4.8)$$

where k_q one of the Jensen–Tsallis kernels defined in the previous section and ϕ is one of the generative embeddings defined in Sect. 4.2.

We consider two types of kernel-based classifiers: K -NN and SVM. Recall that positive definiteness is a key condition for the applicability of a kernel in SVM learning. Standard results from kernel theory [69, Proposition 3.22] guarantee that the kernel k defined in (4.8) inherits the positive definiteness of k_q , thus can be safely used in SVM learning algorithms.

Table 4.1 Summary of two datasets and the corresponding numbers of “words” and “documents”

Problem	# classes	# documents	# words
Renal cancer classification	2	474	168
Colon cancer classification	2	62	500

4.5 Experimental Evaluation

We have applied the proposed approach to two (binary) classification problems in the medical domain: cancer detection in tissue microarray (TMA) images, and colon cancer detection in gene expression microarray data (see Chaps. 9 and 10, respectively, for more details). All the accuracies are computed using the averaged hold out cross validation (30 repetitions). The value of parameter q of the IT kernels is estimated using 5-fold cross validation on the training set. As a baseline, we consider also the linear kernel (which is the most common choice when using generative embeddings). As classifiers, we use *support vector machines* (SVM), with the well-known parameter C adjusted by 5-fold cross validation on the training set, as well as the K -nearest neighbors classifier, with $K = 1$, i.e., the nearest neighbor (NN) rule. When possible, the classifiers have been applied also in the original domain (i.e., without the generative embedding)—this will be made clear in each application.

4.5.1 Application Details

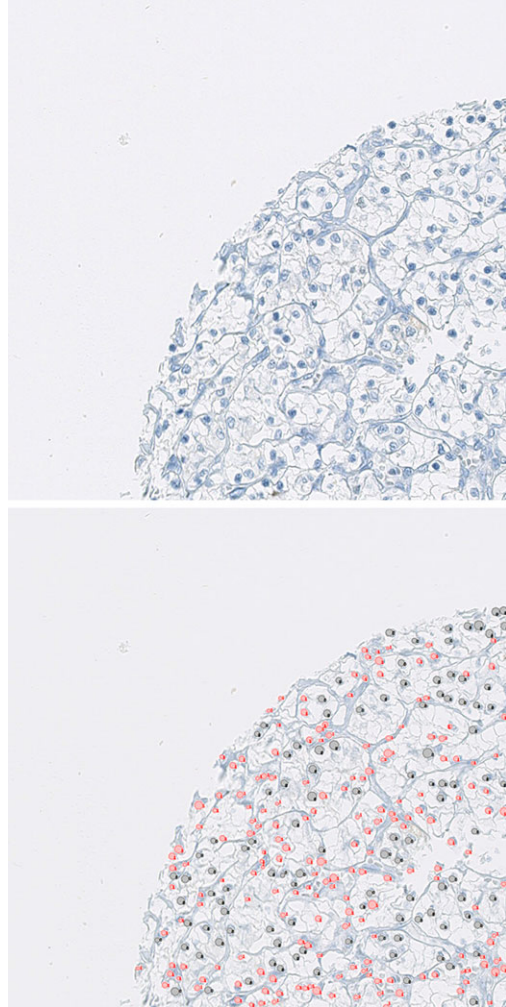
We will now describe the two applications in more detail. In particular, we will describe how the pLSA model is used in each problem, that is, what is the meaning of terms “words” and “documents” in each particular type of data. The datasets used in the experiments are summarized in Table 4.1.

4.5.1.1 Renal Cancer Classification via Tissue Microarray

In the first application, the aim is to analyze tissue microarray (TMA) images in order to identify whether a given renal cell nucleus is malignant or benign. For this purpose, the TMA images are normalized and segmented for nuclei; finally the true labels are assigned by a pool of pathologists (see Fig. 4.1). To build the “visual words,” features are extracted from the segmented nuclei (as in the work of Schüffler et al. [39]) and then quantized into 168 bins. In particular, we used the *pyramid histograms of oriented gradients* (PHOG, see the work of Bosch et al. [27] for details) computed over a 2-level pyramid of patches.

In our experiments, we use a set of three patients (more details can be found in the work of Schüffler et al. [39]), from which 474 nuclei (i.e., “documents” in pLSA terms) were segmented; 321 (67 %) benign and 153 (33 %) malignant.

Fig. 4.1 (Top) One quadrant (1500 × 1500 pixels) of a TMA spot image; (Bottom) A pathologist exhaustively labeled all cell nuclei and classified them into malignant (black) and benign (red)



4.5.1.2 Colon Cancer Classification from Gene Expression Microarray Data

In the second application, the goal is to analyze gene expression microarray data in order to distinguish between healthy people and people affected by colon cancer. The starting point is a microarray gene expression matrix, where the element at position (i, j) represents the expression level of the i th gene in the j th subject/sample. Topic models (of which pLSA is an instance) have been recently and successfully applied in this context [10, 41]. Actually, it is possible to establish an analogy between a word–document pair and a gene–sample pair; it seems reasonable to interpret samples as documents and genes as words. In this way, the gene expression levels in a sample may be interpreted as the word counts in a document. Consequently, we can simply take a gene expression matrix and (of course, after a preprocessing

Table 4.2 Accuracy rates on the renal cancer classification task; see the main text for details. The best overall result is shown in *bold*; accuracies within 5 % of the best are shown in *italic*

Embedding	Linear		Jen–Shan		Jen–Tsal		W–Jen–Tsal	
	NN	SVM	NN	SVM	NN	SVM	NN	SVM
MT-1	0.646	0.690	0.648	0.742	0.612	0.741	0.632	0.742
MT-2	0.644	0.735	0.660	0.742	0.595	0.733	0.625	0.743
FESS-1	0.643	0.709	0.643	0.706	0.619	0.688	0.630	0.702
FESS-2	0.655	0.737	0.653	0.736	0.609	0.743	0.625	0.744
LLR	0.641	0.713	0.640	0.765	0.577	0.765	0.607	0.763
FS	0.651	0.740	0.660	0.760	0.581	0.745	0.611	0.754
TOP	0.637	0.694	0.632	0.684	0.616	0.686	0.620	0.687
PD	<i>0.973</i>	<i>0.976</i>	0.987	<i>0.986</i>	0.425	<i>0.984</i>	0.652	0.987
ORIG	0.631	0.734	0.640	0.742	0.627	0.736	0.607	0.734

step, for example, to remove possibly negative numbers [10]) interpret it as a count matrix \mathbf{C} from which a pLSA model can be estimated.

The experiments were carried out on the dataset of Alon et al. [42], which is composed of 40 colon tumor cases and 22 normal colon tissue samples, each characterized by the expression level of 2000 genes. As is common in gene expression microarray data analysis, a beneficial effect may be obtained by selecting a subgroup of genes, using prior knowledge that genes varying little across samples are less likely to be informative. Hence, we decided to perform the experiments by retaining the top 500 genes ranked by decreasing variance [41].

4.5.2 Results and Discussion

The results are displayed in Tables 4.2 and 4.3, where NN and SVM indicate the results of the nearest neighbor and SVM classifiers, respectively. “Linear” denotes the linear kernel, whereas “Jen–Shan”, “Jen–Tsal” and “W–Jen–Tsal” stands for the Jensen–Shannon, Jensen–Tsallis and weighted Jensen–Tsallis kernels, respectively, as described in Sect. 4.3. The acronyms of the generative embeddings follow the notation described in Sect. 4.2.2: “MT-1” is the mixture topics embedding for a single pLSA, “MT-2” is the posterior topic mixture with one pLSA per class, “FESS-1” is the *free energy score space* for a single pLSA, while “FESS-2” is the FESS using one pLSA per class, “LLR” is the log-likelihood ratio embedding, “FS” is the Fisher space, “TOP” refers to the TOP kernel, and “PD” is the posterior divergence embedding. Finally, “ORIG” refers to the results obtained without generative embedding. The standard errors of means, in all runs, were all less than 0.0032 and 0.0179, for the renal cancer and the colon cancer classification tasks, respectively.

From the tables different observations may be done:

Table 4.3 Accuracy rates on the colon cancer classification task; see the main text for details. The best overall results is shown in *bold*; accuracies within 5 % of the best are shown in *italic*

Embedding	Linear		Jen–Shan		Jen–Tsal		W–Jen–Tsal	
	NN	SVM	NN	SVM	NN	SVM	NN	SVM
MT-1	0.732	0.624	0.775	0.816	0.739	<i>0.861</i>	<i>0.768</i>	<i>0.857</i>
MT-2	0.773	<i>0.842</i>	0.774	<i>0.862</i>	0.772	<i>0.868</i>	0.800	0.878
FESS-1	0.720	0.709	0.711	0.675	0.683	0.635	0.700	0.670
FESS-2	0.748	0.829	0.744	0.822	0.717	0.826	0.726	0.830
LLR	0.722	0.682	0.713	0.778	0.676	0.755	0.688	0.774
FS	0.771	<i>0.852</i>	<i>0.777</i>	<i>0.862</i>	0.773	<i>0.856</i>	0.800	<i>0.875</i>
TOP	0.700	0.704	0.705	0.669	0.672	0.676	0.692	0.674
PD	0.812	0.814	0.814	<i>0.863</i>	0.743	<i>0.862</i>	0.859	<i>0.863</i>
ORIG	0.760	0.753	0.758	0.769	0.660	0.842	0.659	0.816

Table 4.4 Comparison with the state-of-the-art: renal cancer classification on TMA images

Method/Reference	Protocol	Accuracy
ITK on ORIG (Jen–Shan+SVM)	Hold out	0.742
Lin on GE (PD+SVM)	Hold out	0.976
ITK on GE (PD+W–Jen–Tsal+SVM)	Hold out	0.987
[43]	10-fold CV	0.797

- In almost all the cases, the use of IT kernels with generative embeddings outperforms the linear kernel over the same embeddings; the difference is quite clear in some cases.
- Using a generative embedding is almost always beneficial with respect to the use of linear and IT kernels on the original space.
- It is clear from the tables that the best generative embedding is the very recent *posterior divergence* (PD), which is outperformed only in one case by the MT and FESS embeddings. Moreover, it seems that this generative embedding has a slight preference to be used with the IT kernels.
- There is no significant difference among the various IT kernels, even if it may be argued that the weighted Jensen–Tsallis seems to have a slight advantage over the others.
- A summary of the best combination over the different schemes, together with some state-of-the-art results, is reported in Tables 4.4 and 4.5. Even if the other results were obtained with a different protocol, it is evident that the proposed approach is in line with the best results reported in the literature.

Table 4.5 Comparison with the state-of-the-art methods: colon cancer classification

Method/Reference	Protocol	Accuracy
ITK on ORIG (Jen–Tsal+SVM)	Hold out	0.842
Lin on GE (FS+SVM)	Hold out	0.852
ITK on GE (MT2+W-Jen–Tsal+SVM)	Hold out	0.878
[44]	10-fold CV	0.888
[45]	Leave One Out	0.887
[46]	Leave One Out	0.935
[47]	0.7/0.3 CV	0.873

4.6 Conclusions

This chapter reviewed our recent proposal of combining several generative embeddings with information theoretical kernels, to obtain a new class of hybrid generative/discriminative methods for learning classifiers. The generative embeddings herein considered are based on pLSA (probabilistic latent semantic analysis), whereas the information theoretic kernels are based on a non-extensive version of information theory. We have tested the proposed approach on two medical classification problems; the reported experimental results are competitive with other state-of-the-art methods, showing that the proposed approach is promising and deserves further research.

Acknowledgements We acknowledge support from the FET programme (EU FP7), under the SIMBAD project (contract 213250).

References

1. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: *In Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 841–848. MIT Press, Cambridge (2002)
2. Dan Rubinstein, Y., Hastie, T.: Discriminative vs informative learning. In: *International Conference on Knowledge Discovery and Data Mining, KDD'1997*, pp. 49–53. AAAI Press, Menlo Park (1997)
3. Ripley, B.: *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge (1996)
4. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, Berlin (1995)
5. Schölkopf, B., Smola, A.: *Learning with Kernels*. MIT Press, Cambridge (2002)
6. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
7. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 11, pp. 487–493. MIT Press, Cambridge (1998)
8. Lasserre, J., Bishop, C., Minka, T.: Principled hybrids of generative and discriminative models. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 87–94 (2006)

9. Bicego, M., Murino, V., Figueiredo, M.: Similarity-based classification of sequences using hidden Markov models. *Pattern Recognit.* **37**(12), 2281–2291 (2004)
10. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via pLSA. In: *European Conference on Computer Vision (ECCV)*, pp. 517–530 (2006)
11. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: A hybrid generative/discriminative classification framework based on free-energy terms. In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2058–2065 (2009)
12. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: Free energy score space. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 22, pp. 1428–1436. MIT Press, Cambridge (2009)
13. Chandalia, G., Beal, M.J.: Using fisher kernels from topic models for dimensionality reduction. In: *NIPS Workshop on Novel Applications of Dimensionality Reduction* (2006)
14. Chappelier, J.-C., Eckard, E.: PLSI: The true Fisher kernel and beyond. In: *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pp. 195–210 (2009)
15. Figueiredo, M., Aguiar, P., Martins, A., Murino, V., Bicego, M.: Information theoretical kernels for generative embeddings based on hidden Markov models. In: *Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition—S+SSPR'2010*, Izmir, Turkey (2010)
16. Bicego, M., Perina, A., Murino, V., Martins, A., Aguiar, P., Figueiredo, M.: Combining free energy score spaces with information theoretic kernels: application to scene classification. In: *IEEE International Conference on Image Processing—ICIP'2010*, Hong Kong (2010)
17. Bicego, M., Ulaş, A., Schüffler, P., Castellani, U., Mirtuono, P., Murino, V., Martins, A., Aguiar, P., Figueiredo, M.: Renal cancer cell classification using generative embeddings and information theoretic kernels. In: *International Conference on Pattern Recognition in Bioinformatics (PRIB)* (2011)
18. Martins, A., Smith, N., King, E., Aguiar, P., Figueiredo, M.: Nonextensive information theoretic kernels on measures. *J. Mach. Learn. Res.* **10**, 935–975 (2009)
19. Cuturi, M., Vert, J.-P.: Semigroup kernels on finite sets. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 329–336. MIT Press, Cambridge (2005)
20. Cuturi, M., Fukumizu, K., Vert, J.-P.: Semigroup kernels on measures. *J. Mach. Learn. Res.* **6**, 1169–1198 (2005)
21. Moreno, P., Ho, P., Vasconcelos, N.: Kullback–Leibler divergence based kernel for SVM classification in multimedia applications. In: *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, Cambridge (2003)
22. Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenburg, S., Müller, K.-R.: A new discriminative kernel from probabilistic models. *Neural Comput.* **14**, 2397–2414 (2002)
23. Smith, N., Gales, M.: Speech recognition using SVMs. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, pp. 1197–1204. MIT Press, Cambridge (2002)
24. Li, X., Lee, T.S., Liu, Y.: Hybrid generative-discriminative classification using posterior divergence. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2713–2720 (2011)
25. Bicego, M., Pekalska, E., Tax, D.M.J., Duin, R.P.W.: Component-based discriminative classification for hidden Markov models. *Pattern Recognit.* **42**, 2637–2648 (2009)
26. Krishnapuram, B., Carin, L., Figueiredo, M.A.T., Hartemink, A.J.: Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 957–968 (2005)
27. Bicego, M., Lovato, P., Oliboni, B., Perina, A.: Expression microarray classification using topic models. In: *ACM Symposium on Applied Computing*, pp. 1516–1520 (2010)
28. Castellani, U., Perina, A., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., Brambilla, P.: Brain morphometry by probabilistic latent semantic analysis. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 177–184 (2010)

29. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.* **42**(1–2), 177–196 (2001)
30. Hofmann, T.: Learning the similarity of documents: an information-geometric approach to document retrieval and categorization. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 914–920. MIT Press, Cambridge (2000)
31. Smith, N., Gales, M.: Using SVMs to classify variable length speech patterns. Technical Report CUED/F-INFENG/TR–412, Cambridge University Engineering Department (2002)
32. Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **6**, 461–464 (1979)
33. Suyari, H.: Generalization of Shannon–Khinchin axioms to nonextensive systems and the uniqueness theorem for the nonextensive entropy. *IEEE Trans. Inf. Theory* **50**(8) (2004)
34. Cover, T., Thomas, J.: *Elements of Information Theory*. Wiley, New York (1991)
35. Tsallis, C.: Possible generalization of Boltzmann–Gibbs statistics. *J. Stat. Phys.* **52**, 479–487 (1988)
36. Burbea, J., Rao, C.: On the convexity of some divergence measures based on entropy functions. *IEEE Trans. Inf. Theory* **28**(3) (1982)
37. Lin, J.: Divergence measures based on Shannon entropy. *IEEE Trans. Inf. Theory* **37** (1991)
38. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
39. Schüffler, P., Fuchs, T., Ong, C.S., Roth, V., Buhmann, J.: Computational TMA analysis and cell nucleus classification of renal cell carcinoma. In: *32nd DAGM Conference on Pattern Recognition*, pp. 202–211. Springer, Berlin (2010)
40. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: *6th ACM International Conference on Image and Video Retrieval (CIVR)*, pp. 401–408 (2007)
41. Rogers, S., Girolami, M., Campbell, C., Breitling, R.: The latent process decomposition of cdna microarray data sets. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2**(2), 143–156 (2005)
42. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.* **96**(12), 6745–6750 (1999)
43. Ulaş, A., Schüffler, P., Bicego, M., Castellani, U., Murino, V.: Hybrid generative-discriminative nucleus classification of renal cell carcinoma. In: Pelillo, M., Hancock, E. (eds.) *International Workshop on Similarity-Based Pattern Analysis (SIMBAD)*. LNCS, vol. 7005, pp. 77–88. Springer, Berlin (2011)
44. Deegalla, S., Bostrom, H.: Fusion of dimensionality reduction methods: a case study in microarray classification. In: *Proc. Int. Conf. on Information Fusion*, pp. 460–465 (2009)
45. German, D., Afsari, B., Choon, T.A., Naiman, D.Q.: Microarray classification from several two-gene expression comparisons. In: *Proc. Int. Conf. on Machine Learning and Applications*, pp. 583–585 (2008)
46. Liu, H., Liu, L., Zhang, H.: Ensemble gene selection by grouping for microarray data classification. *J. Biomed. Inform.* **43**(1), 81–87 (2010)
47. Wang, L., Zhu, J., Zou, H.: Hybrid Huberized support vector machines for microarray classification and gene selection. *Bioinformatics* **24**(3), 412–419 (2008)

Chapter 5

Learning Similarities from Examples Under the Evidence Accumulation Clustering Paradigm

Ana L.N. Fred, André Lourenço, Helena Aidos, Samuel Rota Bulò, Nicola Rebagliati, Mário A.T. Figueiredo, and Marcello Pelillo

Abstract The SIMBAD project puts forward a unified theory of data analysis under a (dis)similarity based object representation framework. Our work builds on the duality of probabilistic and similarity notions on pairwise object comparison. We address the Evidence Accumulation Clustering paradigm as a means of learning pairwise similarity between objects, summarized in a co-association matrix. We show the dual similarity/probabilistic interpretation of the co-association matrix and exploit these for coherent consensus clustering methods, either exploring embeddings over learned pairwise similarities, in an attempt to better highlight the clustering

A.L.N. Fred (✉) · H. Aidos · M.A.T. Figueiredo
Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
e-mail: afred@lx.it.pt

H. Aidos
e-mail: haidos@lx.it.pt

M.A.T. Figueiredo
e-mail: mtf@lx.it.pt

A. Lourenço
Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
e-mail: arlourenco@lx.it.pt

A. Lourenço
Instituto de Telecomunicações, Lisbon, Portugal

S. Rota Bulò
Fondazione Bruno Kessler, Povo, Trento, Italy
e-mail: samyrota@gmail.com

N. Rebagliati
VTT Technical Research Centre of Finland, Espoo, Finland
e-mail: nicola.rebagliati@gmail.com

M. Pelillo
DAIS, Università Ca' Foscari, Venezia, Italy
e-mail: pelillo@dais.unive.it

M. Pelillo (ed.), *Similarity-Based Pattern Analysis and Recognition*,
Advances in Computer Vision and Pattern Recognition,
DOI [10.1007/978-1-4471-5628-4_5](https://doi.org/10.1007/978-1-4471-5628-4_5), © Springer-Verlag London 2013

structure of the data, or by means of a unified probabilistic approach leading to soft assignments of objects to clusters.

5.1 Introduction

The goal of clustering algorithms is to organize a set of unlabeled objects into groups or clusters such that objects within a cluster are more similar than objects in distinct clusters. Clustering techniques require the definition of a similarity measure between patterns, geometrical or probabilistic, which is not easy to specify in the absence of any prior knowledge about cluster shapes and structure. On the other hand, clustering solutions unveil or induce pairwise similarity, when grouping objects in a same cluster. Given the diversity of clustering algorithms, each one with its own approach for estimating the number of clusters, imposing a structure on the data, and validating the resulting clusters, we are faced with a myriad of potential similarity learners.

Clustering ensemble methods obtain consensus solutions from a set of base clustering algorithms, thus constituting a step towards the goal of assumption-free clustering. Several authors have shown that these methods tend to reveal more robust and stable cluster structures than the individual clusterings in the Clustering Ensemble (CE) [9, 10, 39].

There is a close connection between the concepts of pairwise similarity and probability in the context of unsupervised learning. It is a common assumption that, if two objects are similar, it is very likely that they are grouped together by some clustering algorithm; the higher the similarity, the higher the probability of co-occurrence in a cluster. Conversely, if two objects co-occur very often in the same cluster (high co-occurrence probability), then it is very likely that they are very similar. This duality and correspondence between pairwise similarity and pairwise probability within clusters forms the core idea of the clustering ensemble approach known as *Evidence Accumulation Clustering* (EAC) [9, 11].

The idea of evidence accumulation clustering is to combine the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of data organization. This evidence is accumulated in a co-associations matrix, the intrinsic learned pairwise similarity, which constitutes the core of the method. A consensus solution is obtained by applying a clustering algorithm over this matrix.

In this chapter, we build on the EAC paradigm, exploring the duality of similarity-based and probabilistic interpretations of the learned co-association matrix in order to produce robust and informative consensus solutions. Interpreting co-associations as new data representations, we propose to use embeddings over this matrix, as an intermediate step in the consensus clustering process, in order to extract relevant information into lower dimensional spaces. Consensus (hard) data partitions are obtained from the later by applying hierarchical clustering algorithms. By assuming a probabilistic re-interpretation of the co-association matrix, we then propose a fully probabilistic formulation of the clustering problem, leading to soft

consensus solutions. The method, that we denote as PEACE (*Probabilistic Evidence Accumulation for Clustering Ensembles*), obtains probabilistic cluster assignments through an optimization process that maximizes the likelihood of observing the empirical co-associations given the underlying object to cluster probabilistic assignment model.

The chapter is organized as follows. We start with a brief review of related work on clustering ensemble methods in Sect. 5.2. The notation and basic definitions are provided in Sect. 5.3. The EAC paradigm is reviewed in Sect. 5.4, while the proposed methods based on embeddings and probabilistic modeling are presented in Sects. 5.5 and 5.6, respectively. Results of the application of these methods to real and synthetic benchmark data, in a comparative study with the baseline EAC method, is provided in Sect. 5.7. Conclusions are drawn in a final section.

5.2 Related Work

Clustering is one of the central problems in Pattern Recognition and Machine Learning. Hundreds of clustering algorithms exist, handling differently issues such as cluster shape, density, noise. k -means is one of the most studied and used algorithms [17, 18, 41].

Recently, taking advantage of the diversity of clustering solutions produced by clustering algorithms over the same dataset, an approach known as *Clustering Ensemble methods*, has been proposed and gained an increasing interest [2, 9, 22, 39]. Given a set of data partitions—a clustering ensemble (CE)—these methods propose a consensus partition based on a combination strategy, having in general a leveraging effect over the single data partitions in the CE.

The topic of clustering combination and consensus clustering are completing the first decade of research.

Different paradigms were followed in the literature: (i) similarity between objects, induced by the clustering ensemble [9, 11, 39]; (ii) similarity between partitions [2, 7, 33, 42–44]; (iii) combining similarity between objects and partitions [8]; (iv) probabilistic approaches to cluster ensembles [42, 45, 46].

Strehl and Ghosh [39] formulated the clustering ensemble problem as an optimization problem based on the maximal average mutual information between the optimal combined clustering and the clustering ensemble exploring graph theoretical concepts, and presenting three algorithms to solve it: Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and Meta CLustering Algorithm (MCLA). CSPA, uses a graph partitioning algorithm, METIS [20], for extracting a consensus partition from the co-association matrix. In [33], this approach was extended to allow soft clusterings on the clustering ensemble. Hyper Graph Partitioning Algorithm (HGPA) and Meta CLustering Algorithm (MCLA) are based on hyper-graphs, where vertices correspond to objects, and the hyperedges correspond to the clusters of the clustering ensemble. HGPA obtains the consensus solution using an hyper-graph partitioning algorithm, HMETIS [21]; MCLA uses another heuristic which allows clustering clusters.

Topchy et al. [43, 44] proposed the Quadratic Mutual Information Algorithm (QMI) based on similarities between the partitions on the ensemble rather than similarities between objects. It is based on the notion of median partition defined as the partition that best summarizes the partitions of the ensemble and is optimized using an algorithm based on a squared error criterion.

Ayad and Kamel [2], following [7], proposed the idea of cumulative voting as a solution for the problem of aligning the cluster labels. Each clustering of the clustering ensemble is transformed into a probabilistic representation with respect to a common reference clustering. Three voting schemes are presented: Un-normalized fixed-Reference Cumulative Voting (URCV), fixed-Reference Cumulative Voting (RCV), and Adaptive Cumulative Voting (ACV).

Fern and Brodley [8] proposed the Hybrid Bipartite Graph Formulation (HBGF), where both data points and clusters of the ensemble are modeled as vertices retaining all of the information provided by the clustering ensemble, and allowing to consider the similarity among data points and clusters. The partitioning of this bipartite graph is produced using the multi-way spectral graph partitioning algorithm proposed by Ng et al. [32], which seeks to optimize the normalized cut criterion [37], or as alternative a graph partitioning algorithm, METIS [20].

In [42, 44], Topchy et al. proposed a probabilistic interpretation of the clustering combination problem, formulation the problem as a multinomial mixture model (MM) over the labels of the clustering ensembles. In Wang et al. [45], this idea was extended, introducing a Bayesian version of the multinomial mixture model, entitled Bayesian cluster ensembles (BCE). Using a strategy very similar to *Latent Dirichlet Allocation* (LDA) models [38], but applied to a different input space, features are now the labels of the ensembles, the posterior distribution being approximated using variational inference or Gibbs sampling. More recently, a nonparametric version of BCE was proposed [46].

5.3 Notation and Definitions

Sets are denoted by uppercase calligraphic letters (e.g., \mathcal{O} , \mathcal{E} , ...) except for \mathbb{R} and \mathbb{R}_+ which represent the sets of real numbers and nonnegative real numbers, respectively. The *cardinality* of a set is written as $|\cdot|$. We denote *vectors* with lowercase boldface letters (e.g., \mathbf{x} , \mathbf{y} , ...) and *matrices* with uppercase boldface letters (e.g., \mathbf{X} , \mathbf{Y} , ...). The i th component of a vector \mathbf{x} is denoted as x_i and the (i, j) th component of a matrix \mathbf{Y} is written as y_{ij} . The *transposition* operator is given by the symbol \top . The ℓ_p -norm of a vector \mathbf{x} is written as $\|\mathbf{x}\|_p$ and we implicitly assume a ℓ_2 (or Euclidean) norm, where p is omitted. We denote by $\mathbf{1}_n$ an n -dimensional column vector of all 1's and by $\mathbf{e}_n^{(j)}$ the j th column of the n -dimensional identity matrix. The *trace* of matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is given by $\text{Tr}(\mathbf{M}) = \sum_{i=1}^n m_{ii}$.

A *probability distribution* over a discrete set $\{1, \dots, K\}$ is an element of the *standard simplex* Δ_K , which is defined as

$$\Delta_K = \{\mathbf{x} \in \mathbb{R}_+^K : \|\mathbf{x}\|_1 = 1\}.$$

The *support* $\sigma(\mathbf{x})$ of a probability distribution $\mathbf{x} \in \Delta_K$ is the set of indices corresponding to positive components of \mathbf{x} , i.e.,

$$\sigma(\mathbf{x}) = \{i \in \{1, \dots, K\} : x_i > 0\}.$$

Random variables (r.v.) are represented by uppercase letters (e.g., X), and realizations of the later by corresponding lowercase letters. The probability on an event is denoted as $\Pr(\cdot)$. The *expected value* of a random variable X is denoted by $E(X)$.

The *entropy* of a probability distribution $\mathbf{x} \in \Delta_K$ is given by

$$H(\mathbf{x}) = - \sum_{j=1}^K x_j \log(x_j)$$

and the *Kullback–Leibler divergence* between two distributions $\mathbf{x}, \mathbf{y} \in \Delta_K$ is given by

$$D_{\text{KL}}(\mathbf{x} \parallel \mathbf{y}) = \sum_{j=1}^K x_j \log\left(\frac{x_j}{y_j}\right),$$

where we assume $\log 0 \equiv -\infty$ and $0 \log 0 \equiv 0$.

Let $\mathcal{S} = \{s_1, \dots, s_n\}$ denote a data set with n objects or samples. Let $\mathcal{O} = \{1, \dots, n\}$ be the indices of the set of n objects, and let $\mathcal{O}_u \subseteq \mathcal{O}$ represent a subsampling (without replacement) from \mathcal{O} , with $|\mathcal{O}_u| < n$. When objects are represented in vector form in a d -dimensional feature space, we denote by $\mathbf{O} = [\mathbf{o}_1, \dots, \mathbf{o}_n]$ the $d \times n$ matrix of object vectors, column i corresponding to the vector representation, \mathbf{o}_i , of the i th object. An alternative to the feature representation is the (dis)similarity representation defined on direct pairwise object comparisons. We denote the dissimilarity representation by a $n \times n$ matrix \mathbb{D} , where $d_{ij} = d(s_i, s_j)$ is the dissimilarity value between samples i and j .

The goal of clustering is to organize the objects into K groups or clusters. We distinguish between *hard* and *soft* clusterings. A *hard* clustering is a function $p_u : \mathcal{O}_u \rightarrow \{1, \dots, K_u\}$ assigning a class label, out of K_u available ones, to data points in $\mathcal{O}_u \subseteq \mathcal{O}$. The result of this clustering is a data partition, written as a vector $\mathbf{p}^{(u)} = p_u(\mathcal{O}_u) = [p_i^{(u)}]_{i=1:n}$, $p_i^{(u)} = \mathbf{p}^{(u)}(i) \in \{1, \dots, K_u\}$, or alternatively, on cluster sets representation: $\mathcal{P}_u = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{K_u}\}$, where \mathcal{C}_l denotes the l th cluster (the set of object indices composing cluster l), each object belonging to only one cluster. A *soft* clustering is a function s_u mapping each object $i \in \mathcal{O}_u$ into a probability distribution $\gamma_i^{(u)} \in \Delta_{K_u}$, $\gamma_i^{(u)}$ denoting the soft assignment or degree of membership of object i to each of the K_u clusters. The result of a soft clustering s_u is thus a matrix $\gamma^{(u)} = [\gamma_{kj}^{(u)}]_{k=1:K_u}^{j=1:n}$, $\gamma_{kj}^{(u)}$ denoting the degree of membership of object j to cluster k in clustering u .

In this chapter, pairwise similarities are to be learnt from clustering committees. Without loss of generality, we will consider committees of hard clusterings. We define $\mathcal{E} = \{p_u\}_{u=1}^N = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_N\}$ a clustering ensemble, i.e., a set of N clusterings (partitions) obtained by applying different algorithms (i.e., different

parameterizations and/or initializations) on (possibly) sub-sampled versions of the objects set.

Since each clustering in the ensemble may stem from a sub-sampled version of the original data set \mathcal{O} , some pairs of objects may not appear in all clusterings. Let $\Omega_{ij} \subseteq \{1, \dots, N\}$ denote the set of clustering indices where both objects i and j have been clustered, i.e., $(u \in \Omega_{ij}) \Leftrightarrow ((i \in \mathcal{O}_u) \wedge (j \in \mathcal{O}_u))$, and let $N_{ij} = |\Omega_{ij}|$ denote its cardinality.

According to the EAC paradigm, and following the vector notation for the representation of partitions, the ensemble of clusterings is summarized in the $n \times n$ *co-association matrix* $\mathbb{C} = [c_{ij}]$, where

$$c_{ij} = \sum_{l \in \Omega_{ij}} \mathbf{1}_{p_i^{(l)} = p_j^{(l)}}, \quad c_{ij} \in \{0, \dots, N_{ij}\} \quad (5.1)$$

is the number of times objects i and j are co-assigned the same cluster label over the ensemble \mathcal{E} ($\mathbf{1}_p$ is the indicator function, giving 1 if p holds true, and 0 otherwise). An alternative summarization is the *normalized co-association matrix*, $\hat{\mathbb{C}} = [\hat{c}_{ij}]$, where

$$\hat{c}_{ij} = \frac{c_{ij}}{N_{ij}}, \quad \hat{c}_{ij} \in [0, 1] \quad (5.2)$$

represents the percentage of times objects i and j are gathered in a same cluster over the clustering ensemble.

5.4 The Evidence Accumulation Paradigm (EAC)

The EAC paradigm can be summarized in the following three steps method:

EAC

1. *Build a clustering ensemble \mathcal{E} .* A diversity of clustering solutions is achieved by running several algorithms, or the same algorithm with different parameter values and/or initializations, on possibly sub-sampled versions of the data set.
2. *Accumulate evidence from \mathcal{E} in a pairwise co-association matrix.* Evidence on pairwise associations are accumulated from the individual clusterings in \mathcal{E} . The summary of these associations are given either by:
 - Computing \mathbb{C} and $\{N_{ij}\}$, as given in Sect. 5.3, Eq. (5.1);
 - Determining $\hat{\mathbb{C}}$ using Eq. (5.2).

This voting mechanism is the key issue of the method, subsuming the problem of class correspondence in consensus clustering.

3. *Extract the consensus clustering from the co-associations.* By applying a clustering algorithm over the learned pairwise associations between objects, a consensus clustering is obtained.

The object of the EAC method is the CE, on which it is built, not the actual objects. As such, it is a clustering method that intrinsically preserves data privacy: Individual descriptions of the underlying data are not required in order to produce a clustering combination solution. Furthermore, it effectively fuses information from multiple views of the data, exploring single or hybrid representations, either feature-based or similarity-based. Some of its steps and characteristics are detailed next.

5.4.1 Building Clustering Ensembles

Clustering ensembles can be generated by following two main approaches: (i) choice of data representation and (ii) choice of clustering algorithms or algorithmic parameters.

In the first approach, different partitions of the objects under analysis may be produced by (a) employing different preprocessing and/or feature extraction mechanisms, which ultimately lead to different pattern representations (vectors, strings, graphs, correlations, dissimilarities, etc.) in different feature spaces, or dissimilarity spaces, (b) exploring subspaces of the same data representation, such as using subsets of features, or embeddings, and (c) perturbing the data, such as in bootstrapping techniques (like bagging), or sampling approaches, as, for instance, using a set of prototype samples to represent huge data sets.

In the second approach, we can generate clustering ensembles by (i) applying different clustering algorithms, exploring different concepts of clustering structure, (ii) using the same clustering algorithm with different parameters or initializations, and (iii) exploring different dissimilarity measures for evaluating inter-pattern relationships, within a given clustering algorithm.

A combination of these two main mechanisms for producing clustering ensembles leads to exploration of distinct views of inter-pattern relationships. From a computational perspective, clustering results produced in an “independent way” facilitate efficient data analysis by utilizing distributed computing, and reuse of results obtained previously.

5.4.2 Properties of the Normalized Co-association Matrix \hat{C}

Given the overall general formulation of the EAC paradigm, the method explicitly produces as intermediate result a matrix accumulating evidence on pairwise associations. The later can be given different interpretations, as presented next.

5.4.2.1 EAC as a Kernel Method

The most direct and intuitive interpretation of the normalized co-association matrix, \hat{C} , is as a measure of pairwise similarity between objects, as put in evidence

in pairwise associations provided by the individual clusterings in the ensemble \mathcal{E} . In fact, it is expected that very similar objects are very often put in a same cluster by clustering algorithms. The use of different algorithms and/or parameter configurations for each clustering algorithm enables the derivation of similarity between patterns without the use of a priori information about the number of clusters or the tuning of parameter values. As such, the EAC method, mapping the individual evidence of pairwise similarity in the clustering ensemble into a learned similarity matrix, i.e., by computing a similarity between objects, further used within some consensus clustering algorithm, can be formalized as a kernel method in supervised learning.

5.4.2.2 Co-associations as Pairwise Stability Indices and Multi-EAC

Data subsampling has largely been explored in clustering ensemble methods with the purpose of increasing diversity in the CE, as well as a means to handle the problem of missing data; however, it can also be used as a mechanism for data perturbation in order to evaluate the stability of clustering solutions.

When a clustering ensemble is produced by applying the same clustering algorithm (with the same parameter(s) value(s)) over subsampled versions of the original data, the matrix \hat{C} summarizes the replicability of clustering solutions in terms of stability of pairwise associations, measured in the interval $[0; 1]$.

Taking as basic premise that spurious clusters generated by a clustering algorithm are not likely to be stable, the pairwise stability interpretation of \hat{C} , under these CE construction conditions, has been explored in an extension of the EAC methodology, known as *Multi-EAC*, that incorporates diverse criteria clustering ensembles in a selective combination strategy at the cluster level, as opposed to the overall partition level. This approach has proven to better unveil the intrinsic data organization in the learned pairwise similarity [12], leading to better consensus clustering solutions [27].

5.4.2.3 \hat{C} as a Pairwise Probability Estimator

Let us denote by X_{ij} a random variable indicating if objects i and j belong to the same cluster. X_{ij} is a Bernoulli distributed r.v. with parameter $\theta_{ij} = E(X_{ij})$:

$$X_{ij} = \begin{cases} 1 & \text{with probability } \theta_{ij}, \\ 0 & \text{with probability } (1 - \theta_{ij}). \end{cases} \quad (5.3)$$

For each pair of objects i and j , we collect from \mathcal{E} , the clustering ensemble, N_{ij} independent realizations $x_{ij}^{(u)}$ of X_{ij} , given by

$$x_{ij}^{(u)} = \begin{cases} 1 & \text{if } p_i^{(u)} = p_j^{(u)} \\ (\text{objects } i \text{ and } j \text{ have the same cluster label in partition } \mathcal{P}_u), \\ 0 & \text{otherwise,} \end{cases} \quad (5.4)$$

for $u \in \Omega_{ij}$. The maximum likelihood (ML) estimate $\hat{\theta}_{ij}$ of the parameter θ_{ij} of each r.v. X_{ij} is given by the empirical mean \bar{x}_{ij} , i.e.,

$$\hat{\theta}_{ij} = \bar{x}_{ij} = \frac{1}{N_{ij}} \sum_{u \in \Omega_{ij}} x_{ij}^{(u)} \equiv \frac{c_{ij}}{N_{ij}} \equiv \hat{c}_{ij}. \quad (5.5)$$

Thus, the normalized co-association matrix, $\hat{\mathbb{C}}$, corresponds to the maximum likelihood estimate of the probability of pairs of objects being in the same cluster, as assessed by the clustering committee \mathcal{E} .

5.4.3 From Co-associations to Consensus Clustering

As delineated in Sect. 5.4, consensus clustering solutions are obtained by applying a clustering algorithm over the (normalized) co-association matrix. Given the possible different interpretations of the normalized co-association matrix, $\hat{\mathbb{C}}$, as described in Sect. 5.4.2, different classes of algorithms can be explored for deriving the consensus solution. We categorize them according to the underlying assumption about data representation:

- *(Dis)similarity-based Data Representation* The interpretation of $\hat{\mathbb{C}}$ as a similarity representation of objects, where intrinsic structure is enhanced through the evidence accumulation process, enables the determination of consensus partitions through a variety of clustering algorithms that explicitly use similarities as input, such as in graph-based techniques (e.g., hierarchical linkage methods). Examples of these have largely been explored in the literature, as in the seminal work [9].
- *Vector-based Object Description* The consensus matrix $\hat{\mathbb{C}}$ can also be used as data, rather than as similarity, each line i in the matrix corresponding to a feature vector representation of object i , as its similarity to all objects in the data set. It has been noted [23] that consensus solutions based on this interpretation of $\hat{\mathbb{C}}$ often lead to better results, as compared to similarity-based counterparts.
- *Co-occurrence Probability* The probabilistic interpretation of matrix $\hat{\mathbb{C}}$ as a ML estimate of the probability of pairs of objects being in the same cluster forms the basis of a new class of probabilistic consensus clustering solutions. Starting from the observation that co-occurrences are a special type of dyads, the work in [29] proposes a generative aspect model for dyadic data, as for the normalized co-association matrix; building on the framework of learning from dyadic data by statistical mixture models [16], the authors further explore this generative model for devising consensus clustering solutions under the EAC paradigm. Assuming a multi-labeling framework, where each object has an (unknown) probability of being assigned to each cluster, and exploring $\hat{\mathbb{C}}$ as empirical co-association matrix, the work in [34] formalizes the problem of consensus clustering as an optimization in probability domain, thus obtaining the soft class assignments. The later basic probabilistic formulation is further explored in Sect. 5.6, proposing a new objective function and optimization mechanism.

The above similarity-based and vector-based data descriptors interpretations of co-associations can be explored as input for a clustering algorithm to extract the consensus solution. In addition, they can be seen as data representations in high dimensional spaces, the structure of interest possibly being better described on an embedded manifold. This leads to the application of embedding techniques over the matrix $\hat{\mathbf{C}}$, as an additional intermediate step in the process of deriving a consensus clustering. This approach was first put forward in [1], being further explored in Sect. 5.5.

5.5 Finding Consensus Data Partitions by Exploring Embeddings

We propose to apply embedding methods, also called dimensionality reduction (DR) methods, over the normalized co-association matrix, $\hat{\mathbf{C}}$, interpreting it in two ways: (i) as a feature space, and (ii) as a similarity space. In the first case, we reduce the dimensionality of the feature space; in the second case, we obtain a representation constrained to the similarity matrix $\hat{\mathbf{C}}$. The overall consensus clustering method, hereafter named as DR-EAC, produces consensus solutions by applying a clustering algorithm over the embedded space.

5.5.1 Embedding Methods

In the following, we assume that objects are represented in d -dimensional feature spaces, a data set being represented by the matrix \mathbf{O} . The goal is to find a new data representation, \mathbf{X} , assuming that the data of interest lie on an embedded linear or nonlinear manifold within the higher-dimensional space. To perform embeddings we will use several unsupervised dimensionality reduction (DR) methods, namely Locality Preserving Projections (LPP) [14], Neighborhood Preserving Projections (NPE) [15], Sammon’s mapping [36], Curvilinear Component Analysis (CCA) [6], Isomap [40], Curvilinear Distance Analysis (CDA) [25], Locally Linear Embedding (LLE) [35] and Laplacian Eigenmap (LE) [3] (see Chaps. 2, 6 and 7 for other approaches). We now briefly introduce each of these algorithms.

5.5.1.1 Nonlinear Methods

Locally Linear Embedding (LLE) The working hypothesis of LLE [35] is that the data manifold is smooth and sampled densely enough such that, in the neighborhood of each data point, the manifold can be well approximated by its tangent hyperplane. This hyperplane will usually be dependent of the point on which one is approximating the manifold, hence the word *Locally* Linear Embedding. It should be noted that the name can be misleading—this method is nonlinear.

LLE makes a locally linear approximation of the whole data manifold; it begins by estimating a local coordinate system for each object i , represented by the vector \mathbf{o}_i , from its k -nearest neighbors. To produce the embedding, LLE finds low-dimensional coordinates that preserve the previously estimated local coordinate systems as well as possible.

Technically, LLE first minimizes the reconstruction error $e(\mathbf{W}) = \sum_i \|\mathbf{o}_i - \sum_j w_{ij} \mathbf{o}_j\|^2$ with respect to the coefficients w_{ij} , under the constraints that $w_{ij} = 0$ if i and j are not neighbors, and $\sum_j w_{ij} = 1$. After finding these weights, the low-dimensional configuration of points is next found by minimizing $e(\mathbf{X}) = \sum_i \|\mathbf{x}_i - \sum_j w_{ij} \mathbf{x}_j\|^2$ with respect to the low-dimensional representation \mathbf{x}_i of each object.

Laplacian Eigenmap (LE) The *Laplacian Eigenmap* [3] uses a graph embedding approach. It begins by constructing a graph where each data point is a node, and each node is connected to k other nodes corresponding to the k nearest neighbors of that point. Points i and j are connected by an edge with weight $w_{ij} = 1$ if j is among the k nearest neighbors of i , otherwise the edge weight is set to zero; this simple weighting method has been found to work well in practice [3].

To find a low-dimensional embedding of the graph, the algorithm tries to put points that are connected in the graph as close to each other as possible and does not care about what might happen to the other points.

Technically, LE minimizes $\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 w_{ij} = \text{Tr}(\mathbf{X}^T \mathbf{L} \mathbf{X})$ with respect to the low-dimensional object representations \mathbf{x}_i , where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and \mathbf{D} is a diagonal matrix with elements $\mathbf{d}_{ii} = \sum_j w_{ij}$. This cost function has an undesirable trivial solution: having all points in the same position would have a cost of zero, which would be a global minimum of the cost function. To avoid this problem, the low-dimensional configuration is found by solving the generalized eigenvalue problem $\mathbf{L} \mathbf{x}_i = \lambda_i \mathbf{D} \mathbf{x}_i$ [3]. The smallest eigenvalue corresponds to the trivial solution, but the next smallest eigenvalues yield the desired LE solution (\mathbf{X} being the matrix with the corresponding eigenvectors).

Isomap *Isomap* [40] is a variant of Multidimensional Scaling (MDS) [24], which attempts to find output coordinates that match a given distance matrix. This distance matrix is not computed using simple Euclidean distances; instead, *geodesic distances* along the manifold of the data are used.¹

Given these geodesic distances, the output coordinates are found by standard linear MDS.

Let \mathbf{o}_i and \mathbf{x}_i denote the coordinates of point i on the input (high-dimensional) space and output (low-dimensional) space, respectively. MDS attempts to find the \mathbf{x}_i for all i which minimizes the squared difference between distances in the input space and output space: $\sum_{i,j} (d(\mathbf{o}_i, \mathbf{o}_j) - d(\mathbf{x}_i, \mathbf{x}_j))^2$. In simple terms, MDS is attempting to find the low-dimensional representation of the data which makes the distances between data points as close as possible to the distances in the original space.

¹Technically, these distances are computed along a graph formed by connecting all k -nearest neighbors.

Curvilinear component analysis CCA [6] is a variant of MDS [24] that tries to preserve only distances between points that are near each other in the embedding. This is achieved by weighting each term in the MDS cost function by a coefficient that depends on the corresponding pairwise distance in the embedding; this coefficient is simply 1 if the distance is below a predetermined threshold and 0 if it is larger. This approach is similar to Isomap, but the determination of whether two points are neighbors is done in the output space in CCA, rather than in the input space as in Isomap.

Curvilinear distance analysis CDA [25] is a variant of CCA. Whereas MDS measures distances in the original space using the Euclidean distance, in CDA distances in the original space are measured with geodesic distances, like in Isomap. In all other aspects, CDA is similar to CCA.

5.5.1.2 Linear Methods

Locality Preserving Projections LPP [14] is a linear dimensionality reduction method which attempts to preserve local neighborhood information. It shares many properties of nonlinear techniques such as Laplacian Eigenmaps or Locally Linear Embedding, since it is a linear approximation of the nonlinear Laplacian Eigenmaps.

Neighborhood Preserving Projections NPE [15] is a linear dimensionality reduction method that preserves the local structure of the data. It has similar properties to LPP, but it is a linear approximation of Locally Linear Embedding (LLE).

5.5.2 The DR-EAC Method

We now present the proposed methodology called *Dimensionality Reduction in Evidence Accumulation Clustering* (DR-EAC). It extends the three step EAC method described previously (see Sect. 5.4) with an additional intermediate step: instead of applying a clustering algorithm directly to the normalized co-association matrix, we apply a DR technique to it. As detailed below, we propose two ways to do this, depending on how one interprets the co-association matrix. This DR technique outputs a low-dimensional data representation, which is then fed into a clustering algorithm, deriving the consensus partition. The DR-EAC method is thus summarized in the following four steps:

DR-EAC

1. *Build the clustering ensemble \mathcal{E} .* As discussed before (see Sect. 5.4.1), this can be accomplished in a variety of ways.
2. *Obtain the normalized co-association matrix, \hat{C} ,* as per expression (5.2)—see Sect. 5.3. Then, we interpret this matrix in one of two possible ways (see Sect. 5.4.2):

- *Co-associations viewed as Features*: the i th row of $\hat{\mathbb{C}}$ represents a new set of features for the i th object, an idea originally proposed by Kuncheva et al. [13]. Each object is now represented by the percentage of times it was grouped together with each of the other objects.
 - *Co-associations viewed as Similarities*. Since many DR methods can take as input a matrix of pairwise distances (or dissimilarities), if we transform this similarity matrix $\hat{\mathbb{C}}$ into a matrix of dissimilarities \mathbb{D} , we can exploit this property. Since the elements of $\hat{\mathbb{C}}$ take values in the interval $[0, 1]$, we use a very simple transformation: the new dissimilarity matrix \mathbb{D} has the element d_{ij} given by $1 - \hat{c}_{ij}$.
3. *Apply Dimensionality Reduction techniques*. We apply DR techniques, according to either of the interpretations above, to obtain a new representation of the data, preserving the topology of the original data.
 4. *Extract the consensus partition*. After we get the embedded data, we apply a clustering algorithm to the later in order to extract the consensus solution.

For the DR methods, in step 3, we need to choose a target dimension to reduce the data to and, in some cases, we also have to choose a parameter of the method (usually the number of nearest neighbors to consider). The target dimension is chosen using a Maximum Likelihood Estimator [26]. This MLE assumes that the data points follow a Poisson process (i.e., they are drawn independently from a uniform distribution over the data manifold) and constructs hyperspheres of growing radii r . It then checks how quickly the number of neighbors inside that hypersphere grows with r ; this dependence conveys information about the intrinsic dimension of the data.

For example, if the data lies on a 2-dimensional manifold, the number of neighbors inside a hypersphere of radius r should grow approximately with r^2 , even if the input space has a higher dimension $d \gg 2$.

In all cases, we let each algorithm choose the most suitable parameter of the DR method by an intrinsic criterion. This intrinsic criterion can be the value of the cost function that each algorithm has to minimize, or the reconstruction error. For example, in Isomap we chose the parameter (which is the number of nearest neighbors used to construct a graph) which minimizes the residual variance [40]. It is beyond the scope of this chapter to detail how these parameters should be chosen; the relevant information can be found in the references cited in Sect. 5.5.1.

5.6 PEACE: Probabilistic Evidence Accumulation for Clustering Ensembles

In this section, we propose a probabilistic formulation and solution of the consensus clustering extraction that fully exploits the probabilistic interpretation of the normalized co-association matrix, $\hat{\mathbb{C}}$, presented in Sect. 5.4.2.3.

5.6.1 Problem Formulation

Consider a general probabilistic multi-labeling framework, where each object has an (unknown) probability of being assigned to each cluster. Define the vector

$$\mathbf{y}_i = [y_{1i}, \dots, y_{Ki}]^T \in \Delta_K \quad (5.6)$$

representing the probability distribution over the set of class labels $\{1, \dots, K\}$ which characterizes object $i \in \mathcal{O}$, that is, $y_{ki} = \Pr(i \in \mathcal{C}_k)$, where \mathcal{C}_k denotes the k th cluster. Let $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n] \in \Delta_K^n$ be a $K \times n$ matrix collecting all objects class labels probability distributions.

In our model, we assume that objects are assigned to clusters independently, i.e., $\Pr(i \in \mathcal{C}_k, j \in \mathcal{C}_k) = \Pr(i \in \mathcal{C}_k) \Pr(j \in \mathcal{C}_k)$. Following this independence assumption and definition (5.6), the probability of objects i and j being assigned to the same cluster is given by

$$\sum_{k=1}^K \Pr(i \in \mathcal{C}_k, j \in \mathcal{C}_k) = \sum_{k=1}^K y_{ki} y_{kj} = \mathbf{y}_i^\top \mathbf{y}_j. \quad (5.7)$$

Let C_{ij} be a binomial random variable (r.v.) representing the number of times that objects i and j are co-clustered; from the modeling assumptions above, we have that $C_{ij} \sim \text{Binomial}(N_{ij}, \mathbf{y}_i^\top \mathbf{y}_j)$, that is,

$$\Pr(C_{ij} = c \mid \mathbf{y}_i, \mathbf{y}_j) = \binom{N_{ij}}{c} (\mathbf{y}_i^\top \mathbf{y}_j)^c (1 - \mathbf{y}_i^\top \mathbf{y}_j)^{N_{ij}-c}.$$

Each element c_{ij} of the co-association matrix \mathbb{C} is interpreted as a sample of the r.v. C_{ij} , and the different C_{ij} 's are all assumed independent. Consequently, the probability of observing \mathbb{C} , given the class probabilities \mathbf{Y} , is given by

$$\Pr(\mathbb{C} \mid \mathbf{Y}) = \prod_{\substack{i, j \in \mathcal{O} \\ i \neq j}} \binom{N_{ij}}{c_{ij}} (\mathbf{y}_i^\top \mathbf{y}_j)^{c_{ij}} (1 - \mathbf{y}_i^\top \mathbf{y}_j)^{N_{ij}-c_{ij}}.$$

We therefore formulate the probabilistic consensus clustering problem as an estimation of the unknown class assignments \mathbf{Y} , by maximizing the log-likelihood $\log \Pr(\mathbb{C} \mid \mathbf{Y})$ with respect to \mathbf{Y} . This yields the following maximization problem

$$\hat{\mathbf{Y}} = \arg \max_{\mathbf{Y} \in \Delta_K^n} f(\mathbf{Y}), \quad (5.8)$$

where

$$f(\mathbf{Y}) = \sum_{\substack{i, j \in \mathcal{O} \\ i \neq j}} c_{ij} \log(\mathbf{y}_i^\top \mathbf{y}_j) + (N_{ij} - c_{ij}) \log(1 - \mathbf{y}_i^\top \mathbf{y}_j) \quad (5.9)$$

(constant terms have been dropped).

It's interesting to notice that $f(\mathbf{Y})$ can be written in terms of the Kullback–Leibler divergence $D_{\text{KL}}(\cdot \parallel \cdot)$ as

$$f(\mathbf{Y}) = - \sum_{\substack{i,j \in \mathcal{O} \\ i \neq j}} N_{ij} [H(\mathbf{z}_{ij}) + D_{\text{KL}}(\mathbf{z}_{ij} \parallel \mathbf{w}_{ij}(\mathbf{Y}))],$$

where $\mathbf{z}_{ij} = (c_{ij}/N_{ij}, 1 - (c_{ij}/N_{ij}))^\top \equiv (\hat{c}_{ij}, 1 - \hat{c}_{ij})^\top \in \Delta_2$, $\mathbf{w}_{ij}(\mathbf{Y}) = (\mathbf{y}_i^\top \mathbf{y}_j, 1 - \mathbf{y}_i^\top \mathbf{y}_j)^\top \in \Delta_2$, \hat{c}_{ij} are elements of the normalized co-association matrix $\hat{\mathbf{C}}$, and $H(\cdot)$ is the entropy.

5.6.2 Optimization Algorithm

The optimization method described in this chapter belongs to the class of primal line-search procedures. This method iteratively finds a direction which is *feasible*, i.e., satisfying the constraints, and *ascending*, i.e., guaranteeing a (local) increase of the objective function, along which a better solution is sought. The procedure is iterated until it converges, or a maximum number of iterations is reached.

The first part of this section describes the procedure to determine the search direction in the optimization algorithm. The second part is devoted to determining an optimal step size to be taken in the direction found.

5.6.2.1 Computation of a Search Direction

Consider the Lagrangian of (5.8):

$$\mathcal{L}(\mathbf{Y}, \boldsymbol{\lambda}, \mathbf{M}) = f(\mathbf{Y}) + \text{Tr}(\mathbf{M}^\top \mathbf{Y}) - \boldsymbol{\lambda}^\top (\mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n),$$

where $\mathbf{M} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_n) \in \mathbb{R}_+^{K \times n}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$ are the Lagrangian multipliers (related to positiveness and simplex constraints), and $\mathbf{Y} \in \text{dom}(f)$. By differentiating \mathcal{L} with respect to \mathbf{y}_i and λ and considering the complementary slackness conditions, we obtain the first order Karush–Kuhn–Tucker (KKT) conditions [30] for local optimality:

$$\begin{cases} g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_n + \boldsymbol{\mu}_i = \mathbf{0}, & \forall i \in \mathcal{O}, \\ \mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n = \mathbf{0}, \\ \text{Tr}(\mathbf{M}^\top \mathbf{Y}) = 0, \end{cases} \quad (5.10)$$

where $g_i(\mathbf{Y})$, the partial derivative of $f(\mathbf{Y})$ with respect to \mathbf{y}_i , is given by

$$g_i(\mathbf{Y}) = \sum_{j \in \mathcal{O} \setminus \{i\}} c_{ij} \frac{\mathbf{y}_j}{\mathbf{y}_i^\top \mathbf{y}_j} - (N_{ij} - c_{ij}) \frac{\mathbf{y}_j}{1 - \mathbf{y}_i^\top \mathbf{y}_j},$$

and $\mathbf{1}_n$ denotes a n -dimensional column vector of all 1's. We can express the Lagrange multipliers λ in terms of \mathbf{Y} by noting that

$$\mathbf{y}_i^\top [g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_n + \boldsymbol{\mu}_i] = 0,$$

yields $\lambda_i = \mathbf{y}_i^\top g_i(\mathbf{Y})$ for all $i \in \mathcal{O}$.

Let $r_i(\mathbf{Y})$ be given as

$$r_i(\mathbf{Y}) = g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_K = g_i(\mathbf{Y}) - \mathbf{y}_i^\top g_i(\mathbf{Y}) \mathbf{1}_K,$$

and let $\sigma(\mathbf{y}_i)$ denote the support of \mathbf{y}_i , i.e., the set of indices corresponding to (strictly) positive entries of \mathbf{y}_i . An alternative characterization of the KKT conditions, where the Lagrange multipliers do not appear, is

$$\begin{cases} [r_i(\mathbf{Y})]_k = 0, & \forall i \in \mathcal{O}, \forall k \in \sigma(\mathbf{y}_i), \\ [r_i(\mathbf{Y})]_k \leq 0, & \forall i \in \mathcal{O}, \forall k \notin \sigma(\mathbf{y}_i), \\ \mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n = \mathbf{0}. \end{cases} \quad (5.11)$$

The two characterizations (5.11) and (5.10) are equivalent. This can be verified by exploiting the non-negativity of both matrices \mathbf{M} and \mathbf{Y} , and the complementary slackness conditions.

The following proposition plays an important role in the selection of the search direction. Hereafter, we denote by $(\mathbf{y}_j)_k$ the k th component of cluster assignment \mathbf{y}_j .

Proposition 5.1 *Assume $\mathbf{Y} \in \text{dom}(f)$ to be feasible for (5.8), i.e., $\mathbf{Y} \in \Delta_K^n \cap \text{dom}(f)$. Consider*

$$j \in \arg \max_{i \in \mathcal{O}} \{ [g_i(\mathbf{Y})]_{k_i^+} - [g_i(\mathbf{Y})]_{k_i^-} \},$$

where

$$k_i^+ \in \arg \max_{k \in \{1 \dots K\}} [g_i(\mathbf{Y})]_k \quad \text{and}$$

$$k_i^- \in \arg \min_{k \in \sigma(\mathbf{y}_j)} [g_i(\mathbf{Y})]_k.$$

Then the following holds:

- $[g_j(\mathbf{Y})]_{k_j^+} \geq [g_j(\mathbf{Y})]_{k_j^-}$ and
- \mathbf{Y} satisfies the KKT conditions for (5.8) if and only if $[g_j(\mathbf{Y})]_{k_j^+} = [g_j(\mathbf{Y})]_{k_j^-}$.

Proof We prove the first point by simple derivations as follows:

$$[g_j(\mathbf{Y})]_{k_j^+} \geq \mathbf{y}_j^\top g_j(\mathbf{Y}) = \sum_{k \in \sigma(\mathbf{y}_j)} (\mathbf{y}_j)_k [g_j(\mathbf{Y})]_k$$

$$\geq \sum_{k \in \sigma(\mathbf{y}_j)} (\mathbf{y}_i)_k [g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^-}.$$

By subtracting $\mathbf{y}_j^\top g_j(\mathbf{Y})$, we obtain the equivalent relation

$$[r_j(\mathbf{Y})]_{k_j^+} \geq 0 \geq [r_j(\mathbf{Y})]_{k_j^-}, \quad (5.12)$$

where equality holds if and only if $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$.

As for the second point, assume that \mathbf{Y} satisfies the KKT conditions. Then $[r_j(\mathbf{Y})]_{k_j^-} = 0$ because $k_j^- \in \sigma(\mathbf{y}_j)$. It follows by (5.12) that also $[r_j(\mathbf{Y})]_{k_j^+} = 0$ and therefore $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$. On the other hand, if we assume that $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$ then, by (5.12) and by definition of j , we have that $[r_i(\mathbf{Y})]_{k_i^+} = [r_i(\mathbf{Y})]_{k_i^+} = 0$ for all $i \in \mathcal{O}$. By exploiting the definition of k_i^+ and k_i^- , it is straightforward to verify that \mathbf{Y} satisfies the KKT conditions. \square

Given \mathbf{Y} a non-optimal feasible solution of (5.8), we can determine the indices k_j^+ , k_j^- and j as stated in Proposition 5.1. The next proposition shows how to build a feasible and ascending search direction by using these indices. Later on, we will point out some desired properties of this search direction. We denote by $\mathbf{e}_n^{(j)}$ the j th column of the n -dimensional identity matrix.

Proposition 5.2 *Let $\mathbf{Y} \in \Delta_K^n \cap \text{dom}(f)$ and assume that the KKT conditions do not hold. Let $\mathbf{D} = (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})(\mathbf{e}_n^j)^\top$, where j , $k^+ = k_j^+$ and $k^- = k_j^-$ are computed as in Proposition 5.1. Then, for all $0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}$, we have that $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon \mathbf{D}$ belongs to Δ_K^n , and for all small enough, positive values of ε , we have $f(\mathbf{Z}_\varepsilon) > f(\mathbf{Y})$.*

Proof Let $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon \mathbf{D}$. Then for any ε ,

$$\mathbf{Z}_\varepsilon^\top \mathbf{1}_K = (\mathbf{Y} + \varepsilon \mathbf{D})^\top \mathbf{1}_K = \mathbf{Y}^\top \mathbf{1}_K + \varepsilon \mathbf{D}^\top \mathbf{1}_K = \mathbf{1}_n + \varepsilon \mathbf{e}_n^j (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})^\top \mathbf{1}_K = \mathbf{1}_n.$$

As ε increases, only the (k^-, j) th entry of \mathbf{Z}_ε , which is given by $(\mathbf{y}_j)_{k^-} - \varepsilon$, decreases. This entry is nonnegative for all values of ε satisfying $\varepsilon \leq (\mathbf{y}_j)_{k^-}$. Hence, $\mathbf{Z}_\varepsilon \in \Delta_K^n$ for all positive values of ε not exceeding $(\mathbf{y}_j)_{k^-}$ as required.

As for the second point, the Taylor expansion of f at \mathbf{Y} gives, for all small enough positive values of ε :

$$\begin{aligned} f(\mathbf{Z}_\varepsilon) - f(\mathbf{Y}) &= \varepsilon \left[\lim_{\varepsilon \rightarrow 0} \frac{d}{d\varepsilon} f(\mathbf{Z}_\varepsilon) \right] + O(\varepsilon^2) \\ &= (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})^\top g_j(\mathbf{Y}) + O(\varepsilon^2) > 0 \\ &= [g_j(\mathbf{Y})]_{k^+} - [g_j(\mathbf{Y})]_{k^-} + O(\varepsilon^2) > 0. \end{aligned}$$

The last inequality comes from Proposition 5.1 because if \mathbf{Y} does not satisfy the KKT conditions then $[g_j(\mathbf{Y})]_{k^+} - [g_j(\mathbf{Y})]_{k^-} > 0$. \square

5.6.2.2 Computation of an Optimal Step Size

Proposition 5.2 provides a direction \mathbf{D} that is both feasible and ascending for \mathbf{Y} with respect to (5.8). We will now address the problem of determining an optimal step ε^* to be taken along the direction \mathbf{D} . This optimal step is given by the following one dimensional optimization problem:

$$\varepsilon^* \in \arg \max_{0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}} f(\mathbf{Z}_\varepsilon), \quad (5.13)$$

where $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon \mathbf{D}$. This problem is concave as stated in the following proposition.

Proposition 5.3 *The optimization problem in (5.13) is concave.*

Proof The direction \mathbf{D} is everywhere null except in the j th column. Since the sum in (5.9) is taken over all pairs (i, j) such that $i \neq j$ we have that the argument of every log function (which is a concave function) is linear in ε . Concavity is preserved by the composition of concave functions with linear ones and by the sum of concave functions [5]. Hence, the maximization problem is concave. \square

Let $\rho(\varepsilon')$ denote the first order derivative of f with respect to ε evaluated at ε' , i.e.,

$$\rho(\varepsilon') = \lim_{\varepsilon \rightarrow \varepsilon'} \frac{d}{d\varepsilon} f(\mathbf{Z}_\varepsilon) = (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})^\top g_j(\mathbf{Z}_{\varepsilon'}).$$

By the convexity of (5.13) and Kachurovskii's theorem [19], we have that ρ is non-increasing in the interval $0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}$. Moreover, $\rho(0) > 0$ since \mathbf{D} is an ascending direction as stated by Proposition 5.2. In order to compute the optimal step ε^* in (5.13), we distinguish 2 cases:

- If $\rho((\mathbf{y}_j)_{k^-}) \geq 0$ then $\varepsilon^* = (\mathbf{y}_j)_{k^-}$ for $f(\mathbf{Z}_\varepsilon)$ is non-decreasing in the feasible set of (5.13);
- If $\rho((\mathbf{y}_j)_{k^-}) < 0$ then ε^* is a zero of ρ that can be found by dichotomic search.

Suppose the second case holds, i.e., assume $\rho((\mathbf{y}_j)_{k^-}) < 0$. Then ε^* can be found by iteratively updating the search interval as follows:

$$\begin{aligned} (\ell^{(0)}, r^{(0)}) &= (0, (\mathbf{y}_j)_{k^-}), \\ (\ell^{(t+1)}, r^{(t+1)}) &= \begin{cases} (\ell^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) < 0, \\ (m^{(t)}, r^{(t)}) & \text{if } \rho(m^{(t)}) > 0, \\ (m^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) = 0, \end{cases} \end{aligned} \quad (5.14)$$

for all $t > 0$, where $m^{(t)}$ denotes the center of segment $[\ell^{(t)}, r^{(t)}]$, i.e., $m^{(t)} = (\ell^{(t)} + r^{(t)})/2$.

We are not in general interested in determining a precise step size ε^* but an approximation is sufficient. Hence, the dichotomic search is carried out until the

interval size is below a given threshold. If δ is this threshold, the number of iterations required is expected to be $\log_2((\mathbf{y}_j)_{k^-}/\delta)$ in the worst case.

5.6.2.3 Algorithm and Computational Complexity

Consider a generic iteration t of our algorithm (shown in Algorithm 1) and assume $A^{(t)} = \mathbf{Y}^\top \mathbf{Y}$ and $g_i^{(t)} = g_i(\mathbf{Y})$ given for all $i \in \mathcal{O}$, where $\mathbf{Y} = \mathbf{Y}^{(t)}$.

The computation of ε^* requires the evaluation of function ρ at different values of ε . Each function evaluation can be carried out in $O(n)$ steps by exploiting $\mathbf{A}^{(t)}$ as follows:

$$\rho(\varepsilon) = \sum_{i \in \mathcal{O} \setminus \{j\}} c_{ji} \frac{\mathbf{d}_j^\top \mathbf{y}_i}{A_{ji}^{(t)} + \varepsilon \mathbf{d}_j^\top \mathbf{y}_i} + (N_{ji} - c_{ji}) \frac{\mathbf{d}_j^\top \mathbf{y}_i}{1 - A_{ji}^{(t)} - \varepsilon \mathbf{d}_j^\top \mathbf{y}_i}, \quad (5.15)$$

where $\mathbf{d}_j = (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})$. The complexity of the computation of the optimal step size is thus $O(n\gamma)$ where γ is the average number of iterations needed by the dichotomic search.

Next, we can efficiently update $\mathbf{A}^{(t)}$ as follows:

$$\mathbf{A}^{(t+1)} = (\mathbf{Y}^{(t+1)})^\top \mathbf{Y}^{(t+1)} = \mathbf{A}^{(t)} + \varepsilon^* (\mathbf{D}^\top \mathbf{Y}^{(t)} + \mathbf{Y}^{(t)\top} \mathbf{D} + \varepsilon^* \mathbf{D}^\top \mathbf{D}). \quad (5.16)$$

Indeed, since \mathbf{D} has only two nonzero entries, namely (k^-, j) and (k^+, j) , the terms within parenthesis can be computed in $O(n)$.

The computation of $\mathbf{Y}^{(t+1)}$ can be performed in constant time by exploiting the sparsity of \mathbf{D} as $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \varepsilon^* \mathbf{D}$.

The computation of $g_i^{(t+1)} = g_i(\mathbf{Y}^{(t+1)})$ for each $i \in \mathcal{O} \setminus \{j\}$ can be efficiently accomplished in constant time (it requires $O(nK)$ to update all of them) as follows:

$$\begin{aligned} g_i^{(t+1)} &= g_i^{(t)} + c_{ij} \left(\frac{\mathbf{y}_j^{(t+1)}}{A_{ij}^{(t+1)}} - \frac{\mathbf{y}_j^{(t)}}{A_{ij}^{(t)}} \right) \\ &\quad + (N_{ij} - c_{ij}) \left(\frac{\mathbf{y}_j^{(t+1)}}{1 - A_{ij}^{(t+1)}} - \frac{\mathbf{y}_j^{(t)}}{1 - A_{ij}^{(t)}} \right). \end{aligned} \quad (5.17)$$

The complexity of the computation of $g_j^{(t+1)}$, on the other hand, requires $O(nK)$ steps:

$$g_j^{(t+1)} = \sum_{i \in \mathcal{O} \setminus \{j\}} c_{ji} \frac{\mathbf{y}_i^{(t+1)}}{A_{ji}^{(t+1)}} - (N_{ji} - c_{ji}) \frac{\mathbf{y}_i^{(t+1)}}{1 - A_{ji}^{(t+1)}}. \quad (5.18)$$

By iteratively updating the quantities $A^{(t)}$, $g_i^{(t)}$ and $Y^{(t)}$ according to the aforementioned procedures, we can keep a per-iteration complexity of $O(nK)$, that is linear in the number of variables in \mathbf{Y} .

Algorithm 1: PEACE

Require: \mathcal{E} : ensemble of clusterings
Require: $\mathbf{Y}^{(0)} \in \Delta_K^n \cap \text{dom}(f)$: starting distribution
 Compute \mathbb{C} and $\{N_{ij}\}$ from \mathcal{E}
 Initialize $\mathbf{A}_i^{(0)} \leftarrow (\mathbf{Y}^{(0)})^\top \mathbf{Y}^{(0)}$
 Initialize $g_i^{(0)} \leftarrow g_i(\mathbf{Y}^{(0)})$ for all $i \in \mathcal{O}$, as per Eq. (5.17)
 $t \leftarrow 0$
while termination-condition **do**
 Compute k^+, k^-, j as in Proposition 5.1
 Compute \mathbf{D} as in Proposition 5.2
 Compute ε^* as described in Sect. 5.6.2.2/5.6.2.3
 Update $\mathbf{A}^{(t+1)}$ as per Eq. (5.16)
 Update $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \varepsilon^* \mathbf{D}$
 Update $g_i^{(t+1)}$ as per Eq. (5.17)
 Update $g_j^{(t+1)}$ as per Eq. (5.18)
 $t \leftarrow t + 1$
end while
return $\mathbf{Y}^{(t)}$

Iterations stop when the KKT conditions of Proposition 5.1 are satisfied under a given tolerance τ , i.e., $([g_j(\mathbf{Y})]_{k^+} - [g_j(\mathbf{Y})]_{k^-}) < \tau$.

5.7 Results and Discussion

We evaluated the previous methods on both real and synthetic datasets, in a comparative study with the EAC method. In the later, we explored three hierarchical algorithms for the computation of the consensus solution from the normalized co-association matrix, namely *single-link* (SL), *average link* (AL), and *Wards link* (WL). In this study, we assume known the true number of clusters, K . In order to assess the quality of consensus results, we compute the *consistency index* (CI) between the consensus partition and the ground-truth labeling of the data. The consistency index, also called H index [31], gives the accuracy of the obtained partitions and is obtained by matching the clusters in the consensus partition \mathcal{P}^i with the ground truth partition \mathcal{P}^{GT} :

$$\text{CI}(\mathcal{P}^i, \mathcal{P}^{\text{GT}}) = \frac{1}{n} \sum_{k'=\text{match}(k)} m_{k,k'}, \quad (5.19)$$

where $m_{k,k'}$ denotes the contingency table, i.e., $m_{k,k'} = |\mathcal{C}_k^{(i)} \cap \mathcal{C}_{k'}^{\text{GT}}|$. It corresponds to the percentage of correct labelings when the number of clusters in \mathcal{P}^i and \mathcal{P}^{GT} is the same.

5.7.1 Experimental Setup

We conducted experiments on synthetic datasets (see Fig. 5.1), and on real-world datasets from the UCI Irvine and UCI KDD Machine Learning Repository: iris, wine, house-votes, ionosphere, std-yeast-cell, breast-cancer, and optdigits. Table 5.1 summarizes the experimental setting, indicating the number of clusters, K , and the size, n , of each data set.

Two different types of clustering ensembles were created, exploring different strategies:

- \mathcal{E} -Split—implementing a split strategy [28] (splitting “natural” clusters into small clusters), the K-means was used as base clustering algorithm, with K randomly chosen in an interval $\{K_{\min}, K_{\max}\} = \{\lceil \sqrt{n}/2 \rceil, \lceil \sqrt{n} \rceil\}$. The size of each CE was $N = 100$.
- \mathcal{E} -Hybrid—a combination of multiple algorithms (agglomerative hierarchical algorithms: single, average, ward, centroid link; k-means; spectral clustering [32]) with different number of clusters K_i , as specified in Table 5.1 (last column). For each clustering approach and each parametrization of the same, we generated $N = 100$ different subsampled versions of the data-set (90 % resampling percentage).

5.7.2 Clustering Results Using Embeddings

We applied the DR-EAC method to the clustering ensembles \mathcal{E} -Split and \mathcal{E} -Hybrid, in the two interpretations of the normalized co-association matrix: as similarity, hereafter denoted as *Similarity Space*; and as features, denoted as *Feature Space*. This leads to four experimental scenarios. For each scenario, we applied each of the dimensionality reduction methods described in Sect. 5.5.1, namely LPP, NPE, LLE, LE, Sammon, CCA, Isomap, and CDA. For extracting the consensus partition, we used the same three hierarchical agglomerative methods used with EAC: single-link, average-link, and Wards-link.

Figure 5.2 summarizes the overall performance of the several variants of the method, in direct comparison with EAC. In this figure, each sub-figure plots the four scenario matrices for a given DR method, as indicated at the top. For each scenario, lines correspond to data sets, and columns to the consensus extraction algorithm, SL, AL, and WL. Within each cell, a color scheme is used to code the comparative performances of the DR-EAC vs. EAC methods, as measured by the consistency index, with white corresponding to equal performance, warm color meaning a superiority of DR-EAC over EAC (in a gradient where red corresponding to high/significantly increased performance values); and cool colors (in a gradient of blue) represent a decrease in performance of DR-EAC in comparison with EAC.

Figures 5.3 and 5.4 present the best consistency index obtained for each data set (indicated on the left of each plot), and each consensus clustering method (indicated at the bottom), for the four combinations of interpretations of the matrix \hat{C}

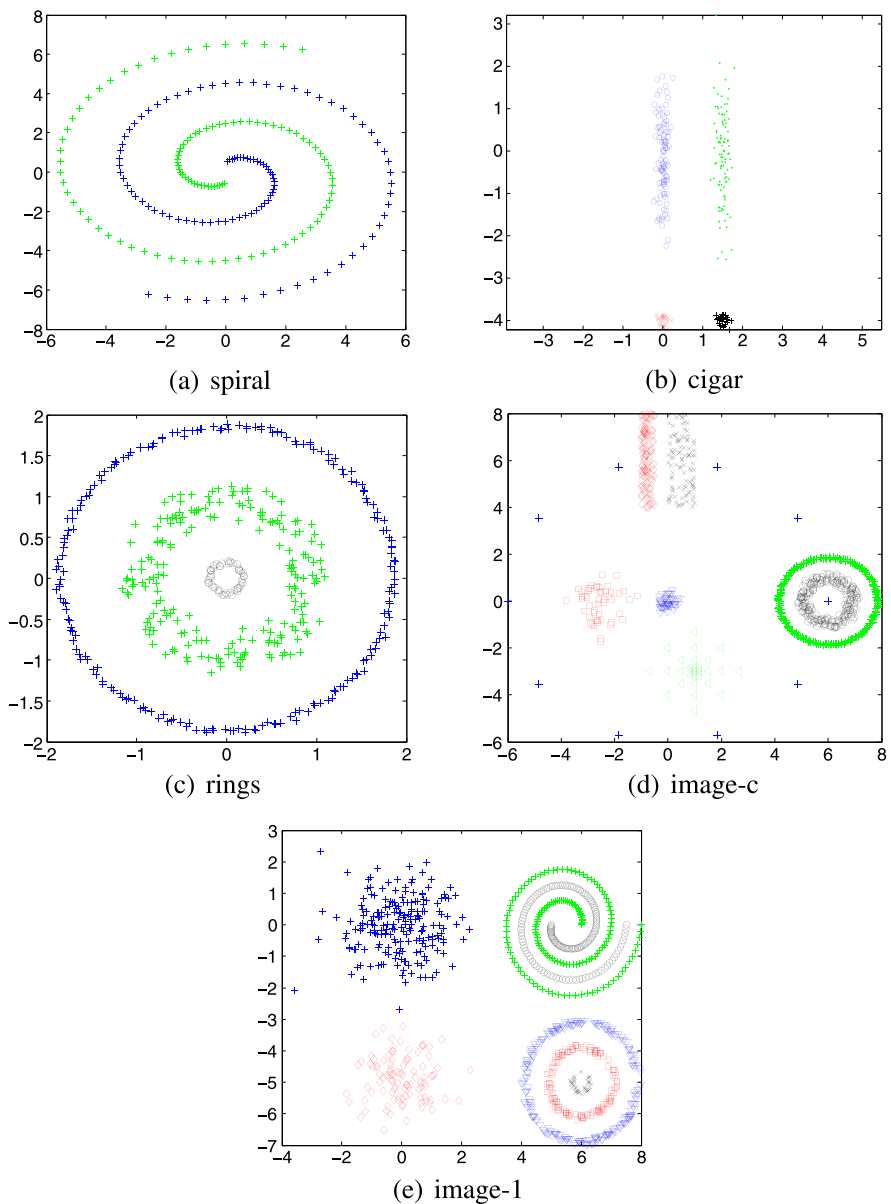


Fig. 5.1 Sketch of the synthetic data sets

and clustering ensemble types. For the DR-EAC method, the variant associated with the DR method is indicated by the corresponding DR designation. On each cell, the best consistency index value obtained by comparing results from the three clustering extraction algorithms is shown over a background color that reveals the winning

Table 5.1 Benchmark datasets and K_i parameter values for the clustering ensembles \mathcal{E} -Hybrid

Data-Sets	K	n	K_i —Ensemble
spiral	2	200	2–9
cigar	4	250	4–9
rings	3	450	2–6
image-c	7	739	8–15,20
image-1	8	1000	7–15,20
iris	3	150	3–10
wine	3	178	4–10,15,20
house-votes	2	232	4–8
ionsphere	2	351	4–10
std-yeast-cell	5	384	5–10
breast-cancer	2	683	2–10
optdigits	10	1000	10, 12, 15, 20

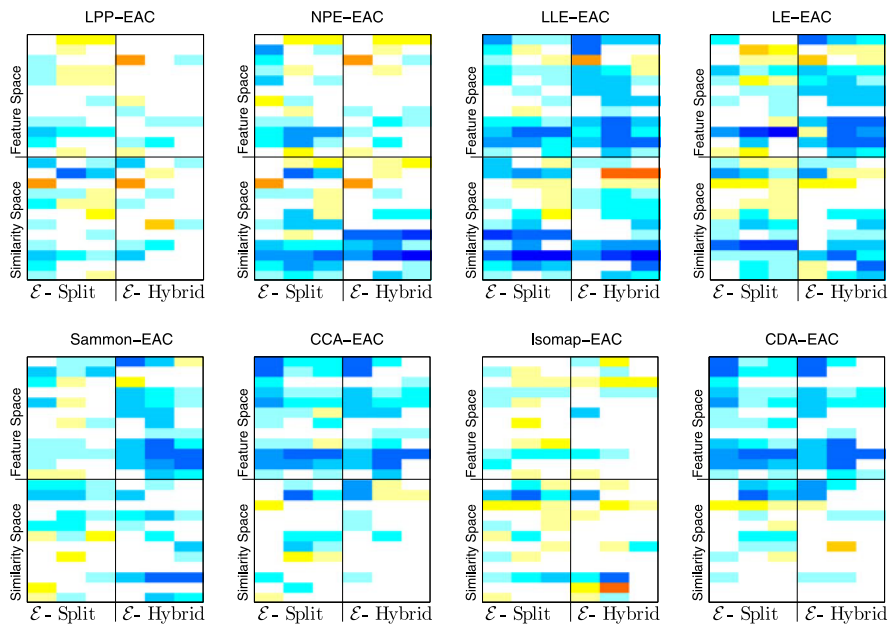
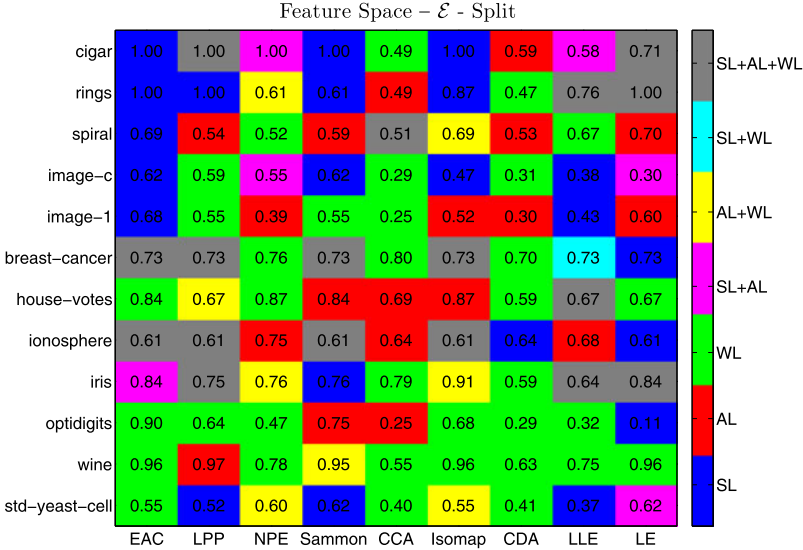
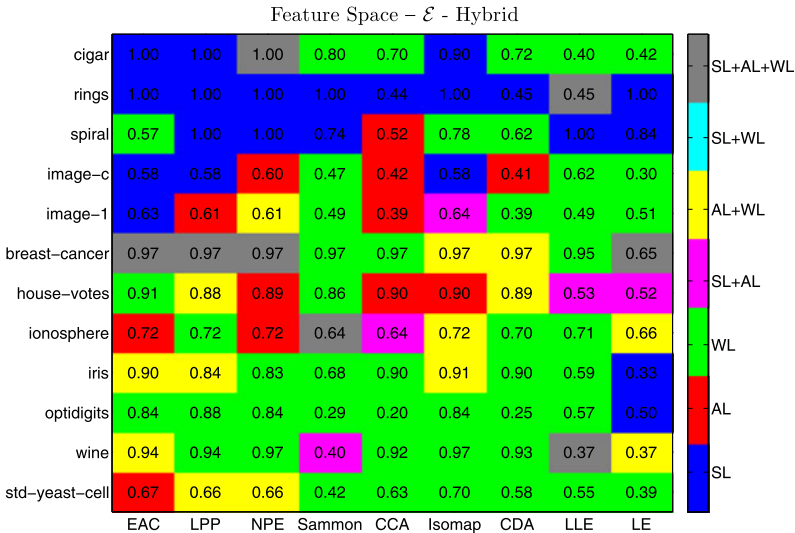


Fig. 5.2 Comparison of various DR methods with EAC using the consistency index. The *top-left sub-figure*, labeled “LPP-EAC”, compares the DR method LPP with the EAC baseline. Four scenarios are depicted in this sub-figure: feature space vs. similarity space and \mathcal{E} -Split vs. \mathcal{E} -Hybrid. Each of the four scenarios presents a 12×3 matrix, corresponding to the 12 datasets and the 3 clustering methods in the following order: SL, AL, and WL. A white cell means that LPP and EAC yielded roughly the same performance. *Warm colors* mean that LPP yielded better performance, whereas *cool colors* mean that it yielded worse performance. Darker tones mean that the difference between the two methods was larger in absolute value. The other *seven sub-figures* show similar information for the seven remaining DR methods



(a) Results for CE \mathcal{E} -Split



(b) Results for CE \mathcal{E} -Hybrid

Fig. 5.3 Results on the feature spaces. (Top) Consistency index for \mathcal{E} -Split for each dataset (vertical axis), DR method (horizontal axis), for the best clustering method (color). Each cell shows the value of the best consistency index obtained for the corresponding dataset and DR method out of the three clustering algorithms tested. A blue cell indicates that the best value came from using single-link, a red cell corresponds to average-link, and a green cell to Ward-link. Color addition is used to present ties: if both single-link and average-link yielded the maximum value, that cell is shown in magenta, etc. (Bottom) Same as before, but for \mathcal{E} -Hybrid

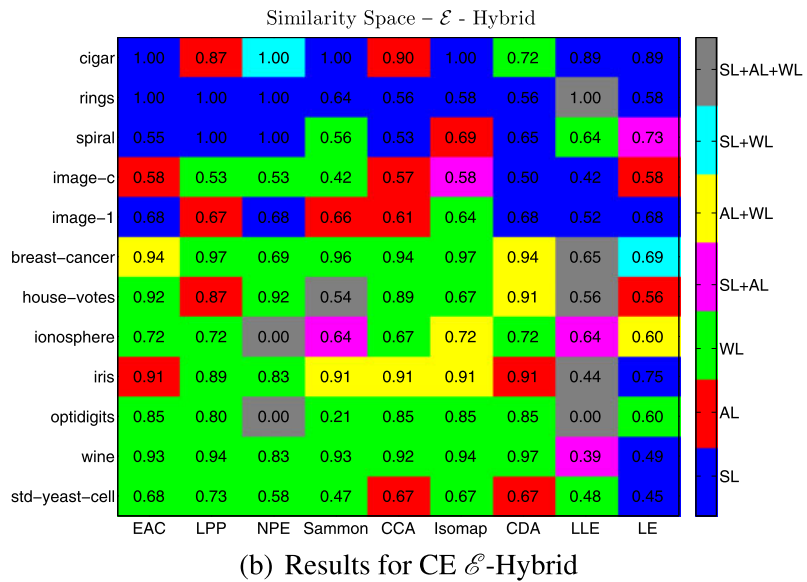
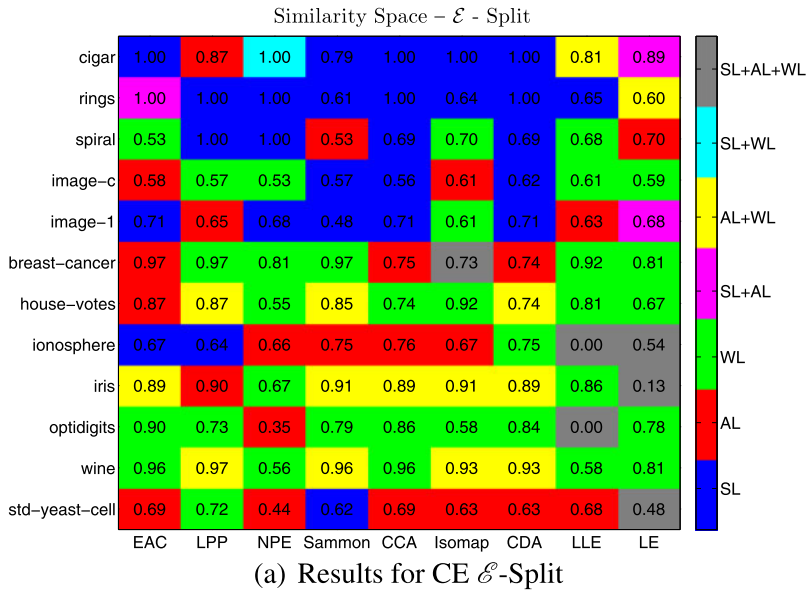


Fig. 5.4 Results on the similarity spaces. The meaning of plots are as in Fig. 5.3

algorithm, according to the color scheme presented on the right of each figure. In addition, for each data set, we circle the best consensus clustering result obtained over the four combinations of spaces interpretations and CEs, as plotted in Figs. 5.3(a), 5.3(b), 5.4(b), and 5.4(a).

Figure 5.2 yields some interesting conclusions. The most immediate one is that blindly performing DR is a bad idea since there are many more blue–cyan cells than orange–yellow ones, randomly choosing a DR method for a certain dataset and clustering method is likely to decrease the performance. However, this should not discourage us from using DR. In fact, for some cases the improvement in the results is considerable, such as for certain cells of LPP in \mathcal{E} -Split.

Overall, Isomap is the method that more consistently produced better results than EAC (notice the high percentage of positive colors), with rare situations of (mild) decreased performance; however, improvements are also in general moderate to low (there are large white areas). LPP is a method that in general leads to good results; improvements are in some instances quite significant, as indicated in reddish tones. This is further corroborated by the analysis of Figs. 5.3 and 5.4, where we can notice the high number of best CI scores obtained for instance in the combination of the Similarity Space with the \mathcal{E} -Split CE (see Fig. 5.4(a)).

LLE, except for point-wise situations, is the method that overall performed worse, immediately followed by LE, with many dark blue areas.

CCA and CDA perform poorly on the feature space, having a more adequate behavior on the similarity space, in particular with the \mathcal{E} -Split CEs. This can be further observed in Fig. 5.4(a).

NPE is better suited for data with complex structure, namely the synthetic data sets; it nevertheless performs reasonably well on real data, in particular on \mathcal{E} -Hybrid CEs. Sammon mapping, on the other hand, performs better with \mathcal{E} -Split CEs, achieving moderate improvements.

Concerning best obtained results per data set and embedding method (Figs. 5.3 and 5.4), it is clear the overall better performance of the single-link algorithm for the extraction of the combined partition over the synthetic data sets (see the large areas of blue, pink and brown on all maps, in particular on the similarity space).

On the other hand, the Wards-link was the best performing method on the real data (green, yellow and brown areas).

5.7.3 Probabilistic Clustering

For each data set, the PEACE algorithm was applied to the clustering ensembles \mathcal{E} -Split and \mathcal{E} -Hybrid, leading to corresponding probabilistic cluster assignments.

Figure 5.5 illustrates the empirical co-association matrices, $\hat{\mathbb{C}}$, and corresponding estimated co-occurrences probabilities, $\mathbf{Y}^T \mathbf{Y}$, on both clustering ensembles, for the iris dataset. In these images, \hat{c}_{ij} values are represented in a gradient of colors from dark blue (corresponding to 0) to red (corresponding to 1). While a block structure of three clusters is apparent in all figures, it is more clear and less noisy in the true co-association $\mathbf{Y}^T \mathbf{Y}$. The corresponding soft cluster assignments, \mathbf{Y} , are plotted in Fig. 5.6, where object indices are on the x -axis, and probabilities for each cluster assignment (on the y -axis) are given in color, in a gradient from dark blue to red.

For the direct comparison with the ground-truth hard-partition, \mathcal{P}^{GT} , the probabilistic consensus clusterings are converted into hard-partitions by assigning each

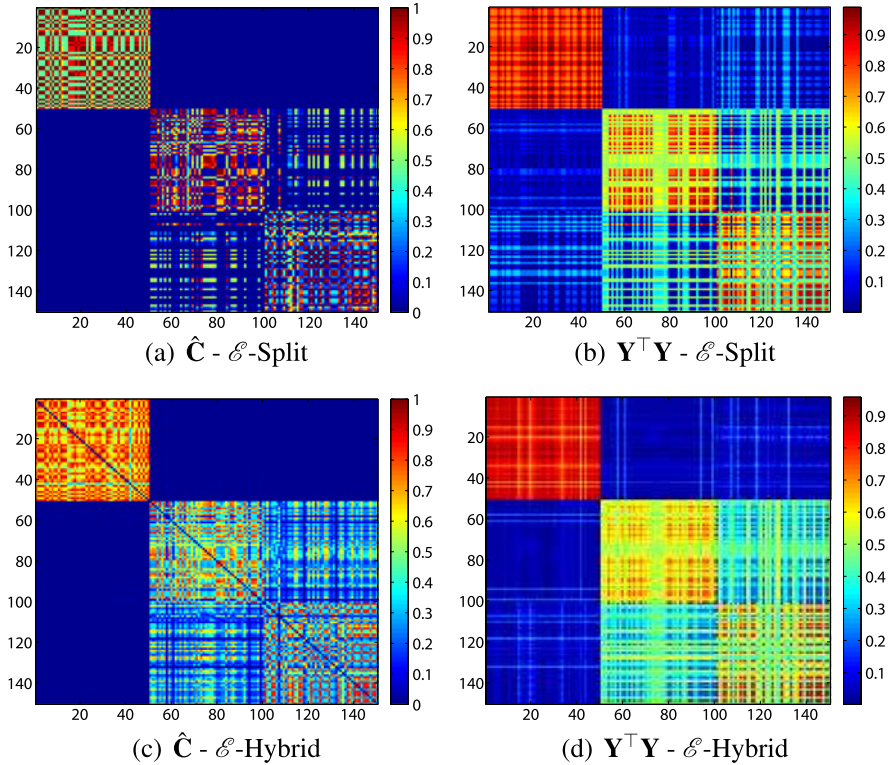


Fig. 5.5 Iris data set. Co-association matrices and corresponding estimated co-occurrences probabilities, as given by the PEACE algorithm. The *top row* corresponds to the clustering ensemble \mathcal{L} -split, while the *bottom row* corresponds to \mathcal{L} -Hybrid

object \mathbf{o}_i to the class with the highest estimated probability in \mathbf{y}_i , i.e., according to the ML rule: $i \in \mathcal{C}_j : j = \arg \max_k y_{ik}$. Given different initializations in the optimization process, it is possible to obtain different consensus solutions with the proposed algorithm. We thus performed several runs of the algorithm, and evaluated the performances in terms of the consistency index, $\text{CI}(\mathcal{P}^i, \mathcal{P}^{\text{GT}})$. Tables 5.2 and 5.3 summarize the obtained results, indicating minimum, maximum, average, and standard deviation of the CIs for each data set. In addition, the first column (“selected”) refers to the CI of the selected consensus solution over the several runs, according to the intrinsic optimization criterion, i.e., highest value of $\text{Pr}(\mathbf{C} | \mathbf{Y})$. The last three columns in these tables register the results with the EAC method with three consensus extraction clustering algorithms: single-, average-, and Wards-link. Highest CI values for each data set are highlighted in bold.

From the analysis of Tables 5.2 and 5.3, it is apparent that the PEACE algorithm performs poorly in data sets exhibiting complex structure, where clusters are defined by connectedness as opposed to compactness properties, such as in most of the synthetic data sets. For these, the EAC method, in combination with the SL algorithm,

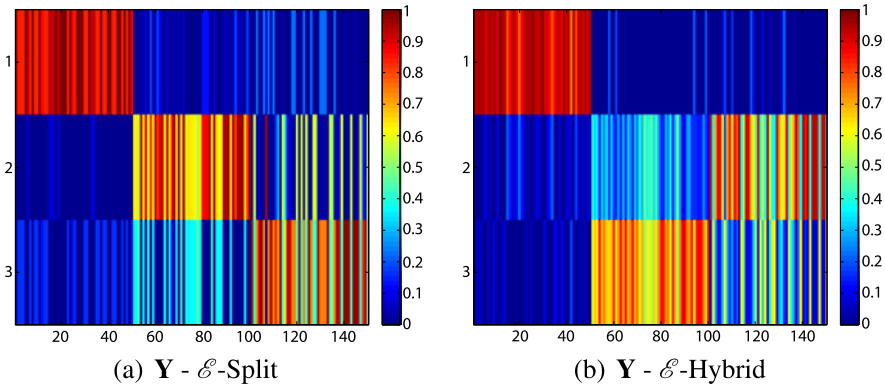


Fig. 5.6 Iris data set—probabilistic cluster assignments given by the PEACE algorithm on the clustering ensembles ℓ -split and ℓ -Hybrid

Table 5.2 Consistency indices of consensus solutions for the clustering ensemble ℓ -Split

Data set	PEACE					EAC		
	selected	av	std	max	min	SL	AL	WL
cigar	0.636	0.628	0.020	0.640	0.592	1.000	0.816	0.708
rings	0.509	0.526	0.018	0.551	0.509	1.000	1.000	0.729
spiral	0.505	0.505	0.000	0.505	0.505	0.505	0.500	0.525
image-c	0.499	0.503	0.002	0.505	0.499	0.582	0.583	0.433
image-l	0.555	0.570	0.025	0.613	0.550	0.666	0.590	0.465
breast-cancer	0.734	0.923	0.106	0.971	0.734	0.657	0.971	0.734
house-votes	0.901	0.892	0.012	0.901	0.879	0.668	0.871	0.853
ionosphere	0.632	0.632	0.000	0.632	0.632	0.667	0.541	0.613
iris	0.907	0.864	0.095	0.907	0.693	0.747	0.893	0.893
optdigits	0.894	0.871	0.042	0.898	0.798	0.618	0.798	0.899
std-yeast-cell	0.544	0.543	0.001	0.544	0.542	0.526	0.688	0.542
wine	0.961	0.961	0.000	0.961	0.961	0.674	0.927	0.961

performs the best, in particular when adopting the split strategy for building the CE, i.e., in ℓ -Split.

On the real data sets, however, the proposed algorithm shows an overall superior performance, positively correlated with the more dense block diagonal structure of the empirical co-association matrices. Corroborating this conclusion, we can notice the increased number of best results (as compared with EAC) in the ℓ -Hybrid CEs (Table 5.3), were this block structure is promoted by the use of lower K values for building the CEs. A notable exception to this conclusion on real data sets is the case of the optdigits, for which much better results are obtained by both PEACE and EAC methods when using the split strategy on the CE. It should be noted, however,

Table 5.3 Consistency indices of consensus solutions for the clustering ensemble \mathcal{E} -Hybrid

Data Set	PEACE					EAC		
	selected	av	std	max	min	SL	AL	WL
cigar	0.688	0.688	0.000	0.688	0.688	1.000	0.820	0.708
rings	0.318	0.320	0.006	0.331	0.318	1.000	0.349	0.351
spiral	0.510	0.510	0.000	0.510	0.510	0.550	0.505	0.515
image-c	0.593	0.533	0.034	0.593	0.517	0.514	0.583	0.559
image-1	0.625	0.625	0.001	0.626	0.625	0.677	0.620	0.606
breast-cancer	0.968	0.968	0.000	0.968	0.968	0.652	0.944	0.944
house-votes	0.901	0.901	0.000	0.901	0.901	0.530	0.530	0.918
ionosphere	0.718	0.718	0.000	0.718	0.718	0.644	0.658	0.715
iris	0.913	0.913	0.000	0.913	0.913	0.747	0.907	0.900
optdigits	0.497	0.419	0.072	0.499	0.366	0.499	0.716	0.855
std-yeast-cell	0.677	0.677	0.000	0.677	0.677	0.359	0.672	0.680
wine	0.944	0.939	0.003	0.944	0.938	0.393	0.371	0.927

that for this dataset the \mathcal{E} -Split CE does not explore a severe splitting strategy: as indicated in Table 5.1, this data set has 10 classes and 1000 samples, leading to an interval $\{K_{\min}, K_{\max}\} = \{15, 31\}$ for \mathcal{E} -Split, while the \mathcal{E} -Hybrid uses the values $\{10, 12, 15, 20\}$ for K . This suggests that the “mild” split strategy favors the revelation of the intrinsic organization structure of the dataset. This is apparent when we compare the empirical and “true” co-associations in the \mathcal{E} -Split with the ones in the \mathcal{E} -Hybrid in Fig. 5.7, where the intrinsic 10-class structure is more clear in \mathcal{E} -Split. This leads to considerably better probabilistic cluster assignments from the \mathcal{E} -Split CE, as seen in Fig. 5.8. If we reorder samples within each “natural” cluster in the co-association matrix, based on pairwise similarities, using for instance the VAT algorithm [4], we obtain the matrix in Fig. 5.9. In this figure, we can observe “microstructure” within each cluster, supposedly associated with writing styles; this can justify the better adequacy of the split strategy for this data set.

5.8 Conclusions

In this chapter, we addressed the Evidence Accumulation Clustering paradigm as a means of learning pairwise similarity between objects, summarized in a co-association matrix. We revised the EAC as a kernel method for extracting relations between objects. We discussed several possible interpretations for the learned co-associations, in particular the duality between similarity/data representation and probabilistic interpretations, and exploited these in two consensus clustering methods: DR-EAC, a hard clustering method exploring embeddings over learned pairwise associations; and PEACE, a unified probabilistic approach leading to soft assignments of objects to clusters.

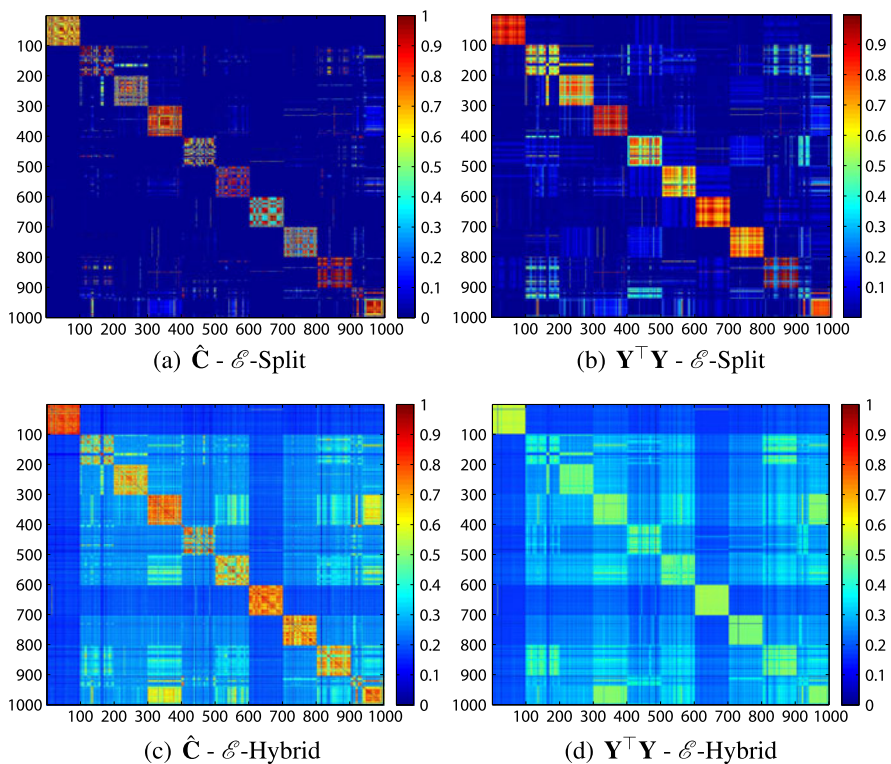


Fig. 5.7 Optidigits data set. Co-association matrices and corresponding estimated co-occurrences probabilities, as given by the PEACE algorithm

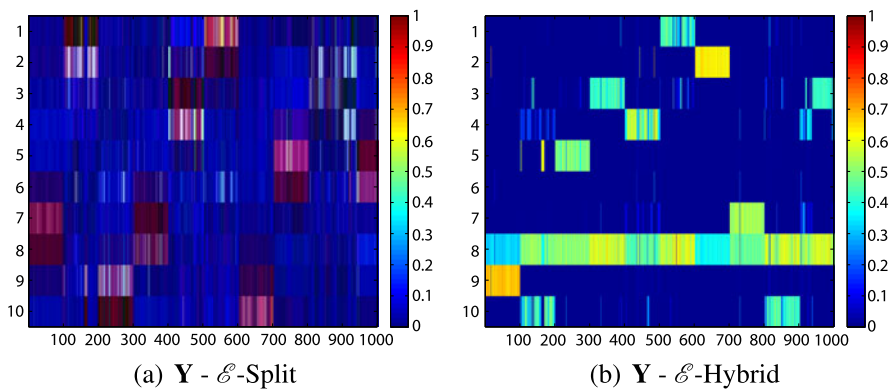
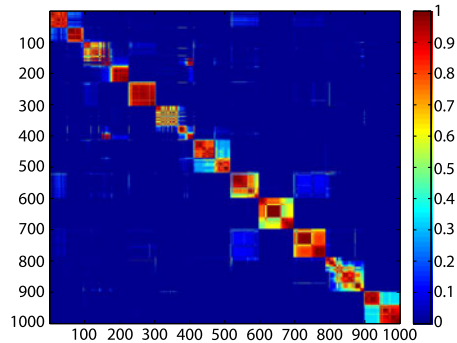


Fig. 5.8 Optidigits data set—probabilistic cluster assignments given by the PEACE algorithm on the clustering ensembles \mathcal{E} -Split and \mathcal{E} -Hybrid

Fig. 5.9 Optidigits data set—reordered empirical co-association matrix \hat{C} for the \mathcal{L} -Split clustering ensemble, evincing micro-structure within each digit class



The DR-EAC method was evaluated in comparison with the EAC, several dimensionality reduction techniques being studied. Although no DR algorithm consistently outperformed all the others, this study showed that the use of dimensionality reduction techniques in clustering ensembles presents interesting advantages in accuracy and robustness. Future work is needed to study the influence of different strategies to construct the clustering ensemble, and criteria for the choice of DR and clustering algorithms.

PEACE obtains probabilistic cluster assignments through an optimization process that maximizes the likelihood of observing the empirical co-associations given the underlying object to cluster assignment model, which was shown to be equivalent to minimizing the Kullback–Leibler divergence between the empirical co-associations and the estimated “real” co-association distribution. When converting soft assignments to hard clusterings, the method performed favorably as compared with the EAC method for handling real data sets, and data with homogeneous clusters. In addition, PEACE, by providing probabilistic cluster assignments to objects, yields a richer level of information about cluster structure. Its poor performance on complex structure data sets is the object of current investigation.

References

1. Aidos, H., Fred, A.: A study of embedding methods under the evidence accumulation framework. In: Pelillo, M., Hancock, E. (eds.) *Similarity-Based Pattern Recognition*. Lecture Notes in Computer Science, vol. 7005, pp. 290–305. Springer, Berlin (2011). http://link.springer.com/chapter/10.1007/978-3-642-24471-1_21
2. Ayad, H., Kamel, M.S.: Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(1), 160–173 (2008)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in Neural Information Processing Systems (NIPS 2001)*, vol. 14, pp. 585–591 (2002)
4. Bezdek, J., Hathaway, R.: Vat: a tool for visual assessment of (cluster) tendency. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02*, vol. 3, pp. 2225–2230 (2002)
5. Boyd, S., Vandenberghe, L.: *Convex Optimization*, 1st edn. Cambridge University Press, Cambridge (2004)

6. Demartines, P., Héroult, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Trans. Neural Netw.* **8**(1), 148–154 (1997)
7. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. In: *AFSS'02*, 332–338 (2002)
8. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: *Proc. ICML'04* (2004)
9. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) *Multiple Classifier Systems*, vol. 2096, pp. 309–318. Springer, Berlin (2001)
10. Fred, A., Jain, A.: Data clustering using evidence accumulation. In: *Proc. of the 16th Int'l Conference on Pattern Recognition*, pp. 276–280 (2002)
11. Fred, A., Jain, A.: Combining multiple clustering using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(6), 835–850 (2005)
12. Fred, A.L., Jain, A.K.: Learning pairwise similarity for data clustering. In: *Proc. of the 18th Int'l Conference on Pattern Recognition (ICPR 2006)*, pp. 925–928. *IEEE Comput. Soc., Washington* (2006). doi:[10.1109/ICPR.2006.754](https://doi.org/10.1109/ICPR.2006.754)
13. Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. *Inf. Fusion* **7**(3), 264–275 (2006)
14. He, X., Niyogi, P.: Locality preserving projections. In: *Advances in Neural Information Processing Systems (NIPS 2003)*, vol. 16 (2004)
15. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: *Proc. of the 10th Int. Conf. on Computer Vision (ICCV 2005)*, vol. 2, pp. 1208–1213 (2005)
16. Hofmann, T., Puzicha, J., Jordan, M.I.: Learning from Dyadic Data. *Advances in Neural Information Processing Systems (NIPS)*, vol. 11. MIT Press, Cambridge (1999)
17. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.* **31**(8), 651–666 (2010)
18. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* **31**, 264–323 (1999)
19. Kachurovskii, I.R.: On monotone operators and convex functionals. *Usp. Mat. Nauk* **15**(4), 213–215 (1960)
20. Karypis, G., Kumar, V.: Multilevel algorithms for multi-constraint graph partitioning. In: *Proceedings of the 10th Supercomputing Conference* (1998)
21. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in vlsi domain. In: *Proc. Design Automation Conf.* (1997)
22. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: *Proc. of the IEEE International Conference on Systems, Man & Cybernetics, Hague, Netherlands*, pp. 1214–1219 (2004)
23. Kuncheva, L., Hadjitodorov, S., Todorova, L.: Experimental comparison of cluster ensemble methods. In: *9th International Conference on Information Fusion*, pp. 1–7 (2006). doi:[10.1109/ICIF.2006.301614](https://doi.org/10.1109/ICIF.2006.301614)
24. Lee, J.A., Verleysen, M.: *Nonlinear Dimensionality Reduction. Information Science and Statistics*. Springer, Berlin (2007)
25. Lee, J.A., Lendasse, A., Verleysen, M.: Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis. *Neurocomputing* **57**, 49–76 (2004)
26. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: *Advances in Neural Information Processing Systems (NIPS 2004)*, vol. 17 (2004)
27. Lourenço, A., Fred, A.: Selectively learning clusters in multi-EAC. In: *International Conference on Knowledge Discovery and Information Retrieval (KDIR 2010)*, Valencia, Spain (2010)
28. Lourenço, A., Fred, A., Jain, A.K.: On the scalability of evidence accumulation clustering. In: *ICPR. Istanbul Turkey* (2010)
29. Lourenço, A., Fred, A., Figueiredo, M.: A generative dyadic aspect model for evidence accumulation clustering. In: Pelillo, M., Hancock, E. (eds.) *Similarity-Based Pattern Recognition. Lecture Notes in Computer Science*, vol. 7005, pp. 104–116. Springer, Berlin (2011). http://link.springer.com/chapter/10.1007/978-3-642-24471-1_8

30. Luenberger, D.G., Ye, Y.: *Linear and Nonlinear Programming*, 3rd edn. Springer, Berlin (2008)
31. Meila, M.: Comparing clusterings by the variation of information. In: *Proc. of the Sixteenth Annual Conf. of Computational Learning Theory (COLT)*. Springer, Berlin (2003)
32. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: *NIPS*, pp. 849–856. MIT Press, Cambridge (2001)
33. Punera, K., Ghosh, J.: *Advances in Fuzzy Clustering and Its Applications*, Chap. *Soft Consensus Clustering*. Wiley, New York (2007)
34. Rota Bulò, S., Lourenço, A., Fred, A., Pelillo, M.: Pairwise probabilistic clustering using evidence accumulation. In: *Proc. 2010 Int. Conf. on Structural, Syntactic, and Statistical Pattern Recognition, SSPR&SPR'10*, pp. 395–404 (2010)
35. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**, 2323–2326 (2000)
36. Sammon, J.W.: A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* **18**(5), 401–409 (1969)
37. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
38. Steyvers, M., Griffiths, T.: *Probabilistic Topic Models*, Chap. *Latent Semantic Analysis: a Road to Meaning*. Laurence Erlbaum, Hillsdale (2007)
39. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**, 583–617 (2002)
40. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**, 2319–2323 (2000)
41. Theodoridis, S., Koutroumbas, K.: *Pattern Recognition*. Elsevier, Amsterdam (2003)
42. Topchy, A., Jain, A., Punch, W.: Combining multiple weak clusterings. In: *IEEE Intl. Conf. on Data Mining*, Melbourne, FL, pp. 331–338 (2003)
43. Topchy, A., Jain, A., Punch, W.: A mixture model of clustering ensembles. In: *Proc. of the SIAM Conf. on Data Mining* (2004)
44. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(12), 1866–1881 (2005)
45. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. In: *9th SIAM Int. Conf. on Data Mining* (2009)
46. Wang, P., Domeniconi, C., Laskey, K.B.: Nonparametric Bayesian clustering ensembles. In: *ECML PKDD'10*, pp. 435–450 (2010)

Part III
Embedding and Beyond

Chapter 6

Geometricity and Embedding

Peng Ren, Furqan Aziz, Lin Han, Eliza Xu, Richard C. Wilson,
and Edwin R. Hancock

Abstract In this chapter, we compare and contrast two approaches to the problem of embedding non-Euclidean data, namely geometric and structure preserving embedding. Under the first heading, we explore how spherical embedding can be used to embed data onto the surface of sphere of optimal radius. Here we explore both elliptic and hyperbolic geometries, i.e., positive and negative curvatures. Our results on synthetic and real data show that the elliptic embedding performs well under noisy conditions and can deliver low-distortion embeddings for a wide variety of datasets. Hyperbolic data seems to be much less common (at least in our datasets) and is more difficult to accurately embed. Under the second heading, we show how the Ihara zeta function can be used to embed hypergraphs in a manner which reflects their underlying relational structure. Specifically, we show how a polynomial characterization derived from the Ihara zeta function leads to an embedding which captures the prime cycle structure of the hypergraphs.

6.1 Introduction

A particularly interesting case in which the use of similarity-based representations is highly relevant is when the data is abstracted in terms of graphs, a data structure that is used extensively throughout computer science to represent relational data. For instance, genomic data, shape/image data and documents can all be abstracted using relational graphs [124]. Each of these applications can potentially involve large data sets of tens of thousands or even millions of graphs. One of the challenges that arise is that of knowledge discovery from large graph data sets. The tools that are required in this endeavor are robust algorithms that can be used to organize, query

P. Ren · F. Aziz · L. Han · E. Xu · R.C. Wilson · E.R. Hancock (✉)
Department of Computer Science, University of York, York, UK
e-mail: erh@cs.york.ac.uk

P. Ren
e-mail: pengren@cs.york.ac.uk

R.C. Wilson
e-mail: wilson@cs.york.ac.uk

and navigate such databases. Unfortunately, the manipulation of relational data has proven more elusive than that of vectorial data. Also, lacking a canonical order or correspondence between nodes, relational structures do not have a natural map to a uniform feature space, and this makes the application of traditional feature-based approaches problematic. On the other hand, it is quite natural to provide suitable notions of distance (or similarity) between them, thereby making similarity-based techniques particularly attractive.

The term “embedding” refers to any procedure that takes a set of (dis)similarities as input and produces a vectorial representation of the data as output, such that the proximities are either locally or globally preserved. This is an (approximate or ideal) isometric mapping which finds a set of vectors in an instances-specific Euclidean space that are capable of describing the data satisfactorily. One of the earliest and best-known attempts towards this goal is multi-dimensional scaling (MDS) [125]. In MDS, the coordinates are assigned in such a way that a given set of dissimilarity, similarity, or ordinal relations is preserved as closely as possible by the embedded points. More recent approaches try to reduce the curse of dimensionality by inferring a low dimensional manifold in which the data resides. The focus is either on the local preservation of Euclidean distances [66] or by an approximation of suitably re-defined (dis)similarities [7, 82]. However, here the emphasis is on finding a low-dimensional representation of the feature space (often for visualization purposes) rather than correcting non-geometric effects. There are conflicting reports of the viability of these embedding procedures as a means to correcting non-(geo)metric behavior [103, 126]. Other approaches to correcting the non-(geo)metricity of the similarities range from changing the signature of the non-Euclidean directions making it Euclidean [105, 109], to transforming all off-diagonal elements of the dissimilarity matrix by a concave function [109, 127, 128], to adding a suitable constant to all off-diagonal elements of the dissimilarity matrix to make it embeddable in a Euclidean space [45, 105, 129]. Finally, we mention the “dissimilarity representation” introduced by Pekalska and Duin [109], described also in Chap. 2, which is not based on corrections or approximations, but attempts instead to arrive at a vectorial representation with a different interpretation of the given similarities. The graph-drawing community [130] has made considerable progress in understanding how to visualize complex graphs. This is a problem of generic importance, since its solution involves embedding a graph in a low-dimensional space. In fact, the topic of how to embed a graph in a low-dimensional manifold, although studied for several decades by mathematicians [131], has recently attracted renewed interest because of the need to visualize complex relational structures such as the internet. Recently, there has been a significant convergence of effort with the machine learning, graphical models, algorithms and pattern recognition communities focusing on the commonality of the task at hand. Despite this effort aimed at developing discrete algorithms for manipulating large sets of graph-data, the problem of learning probabilistic generative models from sets of relational data has received less attention. Early efforts concentrated on how to extend ideas from string matching to graphs, and led to the definition of similarity measures based on graph edit distance [84]. An alternative approach is to use graph-spectral approaches to replace the structural

characterization of graphs with a geometric one. The first steps here were to explore whether the spectrum of the Laplacian matrix could be used to characterize graphs for the purposes of clustering [120]. More recently, ideas from spectral geometry have been exploited in the manifold learning community to understand the rate of convergence of the discrete and continuous Laplacians [132].

Despite the growing interest around embedding, the search for robust embeddings procedures on structured data such as weighted graphs has proven elusive, and their geometric and probabilistic characterizations still remains to be explored in depth. In a prior work [108], we have explored the link between the curvature associated with an edge and the embedding coordinates of nodes. The first approach to this problem involved estimating the curvature of edges, using the difference between geodesic and Euclidean distances for standard embeddings. This has been explored using both the Laplacian embedding and the heat-kernel embedding, and the latter provides more flexibility due to the inclusion of a time parameter. The second approach to the problem has been to embed nodes of a graph onto a manifold of fixed Gaussian curvature using Kruskal coordinates. However, while the former approach is spectral and hence simple, it does not allow the curvature of the manifold to be controlled. The second approach, on the other hand, allows curvature to be specified but does not have a spectral realization. In the SIMBAD project, we explored in depth the geometric nature of the embeddings that result from the spectral analysis of graphs. To this end, we drew on ideas from spectral geometry to extract differential invariants from the graph-spectra. With the geometric characterization to hand, we aimed to construct probabilistic models that can account for the distributions of the invariants. Within this strand, we also investigated approaches that, instead of approximating the original (dis)similarities by Euclidean distances, try to preserve the underlying group structure of the data, thereby bringing us back to the geometric domain and hence allowing us to apply standard methods.

The overall goal of the work described in this chapter is that, given some dissimilarity data describing objects of interest, we wish to develop algorithms for transforming them into instance-specific spatial representations (embeddings) that are suitable for geometric learning algorithms. In particular, we focus on the following classes of method:

- Spectral and geometric manifold embedding
- Structure-preserving embedding

Under the first bullet, within SIMBAD our aim was to develop spectral methods for embedding weighted graphs in a geometrically meaningful way, and using the resulting embeddings to construct generative models for graph structure. In particular, we aimed at developing spectral methods for embedding with guaranteed curvature properties. The route is provided by the spherical embeddings, where we analyze the links between the radius of curvature and the statistics of the dissimilarity data. This idea has its routes in recently developed methods for representing the statistics on manifolds [121]. With the embedding to hand, we use Principal Geodesic Analysis (PGA) [116] to construct classifiers for the embedded data.

The second bullet implies a category of embedding methods with a fundamentally different focus: instead of approximating the original (dis)similarities by Euclidean distances, these approaches try to preserve the underlying group structure of the data. Here we turn to the Ihara zeta function to capture invariances of graphs at the level of prime cycles. We aim at devising model-specific embedding procedures that preserve the underlying structure of the graphs and hypergraphs. Such embeddings then allow us to apply the whole arsenal of data preprocessing methods that have been developed for vectorial data over the last decades. The next chapter describes another approach to structure-preserving embedding in the context of clustering.

6.1.1 Curvature Dependent Embedding

Many pattern recognition problems can be posed in terms of measuring the dissimilarities between a set of objects. This is a very general approach, as it is a superset of the classic feature-based approach. Nearly all approaches to recognition involve measuring a dissimilarity or distance and classifying on this basis. One approach to this problem is to embed objects into a vector-space using techniques such as multidimensional scaling or ISOMAP [102]. Once embedded in such a space then the objects can be characterized by their embedding co-ordinate vectors, and analyzed in a conventional manner using Euclidean distance.

There are, however, some limits to this paradigm; Euclidean distances are always *definite* and are intrinsically unable to represent dissimilarities which are *indefinite*. We discuss the issue of indefinite dissimilarities in more detail in the next section. In practice, many dissimilarity measures are indefinite; examples include shape-similarities, and distance measures used in gesture interpretation and graph comparison, but there are many more. Any method of comparison which relies on local alignment or variable local control parameters has the potential to produce indefinite (non-Euclidean) dissimilarities.

One alternative is to ‘correct’ the data to remove the indefinite part. However, as the analysis of Chap. 2 has shown (see also [103]), there is potentially useful information in the non-Euclidean part of the dissimilarities, and removing this can result in worse performance. Another alternative is to embed the data in a pseudo-Euclidean space, i.e., one where certain dimensions are characterized by negative eigenvalues and the squared-distance between objects has positive and negative components which sum together to give the total distance. A pseudo-Euclidean space is, however, non-metric, which makes it difficult to correctly compute the geometric quantities required by many classifiers. This is because locality is not preserved in this space; two points which are far apart can both be close to a third point.

A third alternative, which we explore here, is to use a non-Euclidean, but metric, embedding space. A Riemannian manifold is curved, and the geodesic distances are metric. However, they can also be indefinite and so can represent indefinite dissimilarities. In this chapter, we explore the embedding of objects onto the hypersphere

with its associated spherical geometry. Non-Euclidean embeddings have been reported elsewhere in the literature. For example, Lindman and Caelli have studied both spherical and hyperbolic embeddings in the context of interpreting psychological data [104]. Cox and Cox [105] describe multidimensional scaling constrained to a spherical space and optimize the stress to find a good embedding. Shavitt and Tankel have used the hyperbolic embedding as a model of internet connectivity [106]. Hubert et al. have investigated the use of unidimensional embeddings on circles [107]. Robles-Kelly and Hancock [108] preprocess the available similarity data so that it conforms either to elliptic or hyperbolic geometry. In practice, the former corresponds to a scaling of the distance using a sine function, and the latter scaling the data using a hyperbolic sine function.

6.1.2 Graph Characteristics and Zeta Functions

Various statistical methods are available for learning patterns represented by vectors. However, these statistical methods are not suitable for structured data such as trees, graphs and hypergraphs. This is because structural patterns cannot be easily converted into vectors, and the difficulties arise in several aspects. First, there is no natural ordering for the vertices in an unlabeled structure, and this is in contrast to vector components that have a natural order. Second, the variation within a particular graph class may result in subtle changes in structures of individual graphs. This may involve different vertex set and edge set cardinalities for graphs drawn from the same class. Moreover, subspaces (e.g., eigenspaces) spanned by the matrix representations of graphs with different vertex set cardinalities are of different dimensions, and thus pattern vectors residing in the resulting subspaces would be of different lengths. All these difficulties need to be addressed if we want to apply the existing statistical methods to learning with structural patterns.

The task of structural characterization is to characterize classes of structural patterns into a feature space where statistical learning methods can be readily applied. To this end, the key issue is to extract from structural patterns a set of characteristics which not only exactly describe the individual structures but also capture the variations between/within the structure classes. In this regard, the most straightforward characteristics for graphs are the topological properties, such as vertex set cardinality, edge density, graph perimeter and volume [48]. Furthermore, by measuring the topological difference between graphs, graph edit distance can be neatly defined [67]. Bunke et al. [28, 57, 58] embed graphs into a feature space by using kernel strategies which adopt edit distance as a similarity measure. Within such graph characterization frameworks, graphs can be easily classified by using statistical learning approaches such as SVM. Although the topological features have a straightforward meaning concerning the structures, they are hard to enumerate for objects with a considerable size. The computational complexity for edit distance is exponential to the cardinality of the vertex set and is usually computationally prohibitive in practice, unless approximations are made subject to certain

constraints [57]. These shortcomings limit the direct use of topological properties for the purpose of structural characterization.

Another approach to graph characterization is to extract alternative vertex permutation invariant characteristics straightforwardly from the matrix representations of graphs. Here the initial matrix representation \mathbf{M} can be based either on the adjacency matrix, the Laplacian matrix or the signless Laplacian [22]. The definition of the adjacency matrix \mathbf{A} for a graph $G(V, E)$ is as follows

$$A_{uv} = \begin{cases} w(u, v) & \text{if } \{u, v\} \in E; \\ 0 & \text{otherwise;} \end{cases} \quad (6.1)$$

where $w(u, v)$ is the weight attached to the edge $\{u, v\}$. For an unweighted graph, $w(u, v)$ is 1 if there is an edge between vertices u and v . The degree of a vertex $u \in V$, denoted by $d(u)$, is defined as

$$d(u) = \sum_{v:\{v,u\} \in E} w(u, v). \quad (6.2)$$

For an unweighted graph, the degree of a vertex is simply the number of vertices adjacent to it. For a graph $G(V, E)$ with $|V| = N$, the matrix

$$\mathbf{D} = \text{diag}(d(v_1), d(v_2), \dots, d(v_N)),$$

with the vertex degrees on the diagonal and zeros elsewhere is referred to as the degree matrix.

The Laplacian matrix \mathbf{L} of a graph $G(V, E)$ is defined as $\mathbf{L} = \mathbf{A} - \mathbf{D}$, with entries

$$L_{uv} = \begin{cases} W(u, v) & \text{if } \{u, v\} \in E; \\ -d(u) & \text{if } u = v; \\ 0 & \text{otherwise.} \end{cases} \quad (6.3)$$

The matrix representation can be characterized using its eigenvalues $\text{sp}(\mathbf{M})$ and eigenvectors (i.e., using spectral graph theory). For instance, Luo, Wilson and Hancock [48] have made use of graph spectra to construct a set of handcrafted permutation invariant spectral features for the purpose of clustering graphs. Kondor et al. [42] have presented an approach to extracting the skew spectrum from the adjacency matrix of a graph up to a combinatorial transformation, and incorporated it into SVM kernels for the classification of chemical molecules. Furthermore, the same authors have refined their spectral method by considering the number as well as the position of labeled subgraphs in a given graph [43]. Though the spectral features appear to be less related to graph topology than the straightforward topological characteristics, the Laplacian spectra give a competitive performance in clustering graphs over various alternative methods [92].

For the graph matrix representation \mathbf{M} , the coefficients of its characteristic polynomial $\det(\lambda \mathbf{I} - \mathbf{M})$ can also be taken as graph characteristics. These coefficients are closely related to the eigenvalues of \mathbf{M} , i.e., the graph spectrum. Brooks [11]

has generalized the computation of the coefficients of the characteristic polynomial using three different methods. His first method is to express the coefficients in terms of the eigenvalues of the matrix representation, his second method uses the relationship between the coefficients and the k th derivative of the associated determinant, and the third method is a brute force method using matrix elements. Thus, it is clear that the eigenvalue-based and polynomial-based approaches are closely related to each other and can lead to a number of practical graph characterizations. In this regard, pioneering research can be found in Wilson, Hancock and Luo's work [91] which shows how to extract a rich family of permutation invariants from a graph by applying elementary symmetric polynomials to the elements of the spectral matrix derived from the Laplacian matrix.

An alternative possible characterization method that has received relatively little attention in the computer vision and pattern recognition community is provided by the zeta functions. In number theory, the Riemann zeta function is determined by the locations of the prime numbers. Bai, Wilson and Hancock [3] have explored the use of a modified version of the Riemann zeta function as a means of characterizing the shape of the heat kernel trace of a graph. They have also shown that the derivative of the zeta function at the origin is related to the determinant of the Laplacian matrix. Another natural extension of the Riemann zeta function from prime numbers to graphs is the Ihara zeta function. The Ihara zeta function is determined by the set of prime cycles on a graph, and is detailed in [39] and [40]. Hashimoto [35] subsequently deduced explicit factorizations for bi-regular bipartite graphs. Bass [6] has generalized Hashimoto's factorization to all finite graphs. Stark and Terras [78–80] have published a series of articles on the topic. They commence by presenting a survey of the Ihara zeta function and its properties. Their novel contribution is to generalize the Ihara zeta function to develop edge and path based variants. Recently, Storm has further developed and refined the Ihara zeta function for hypergraphs [81].

The Ihara zeta function draws on the reciprocal of a polynomial associated with a graph and is hence akin to methods from algebraic graph theory. However, it also relies upon a graph transformation. This is an interesting observation since the quest for improved alternatives to the adjacency and Laplacian matrices has been a long-standing quest in spectral graph theory. Recently, the signless Laplacian (i.e., the degree matrix plus the adjacency matrix) has been suggested. However, Emms et al. [25–27] have recently drawn on ideas from quantum computing [32, 72] and have shown that a unitary matrix characterization of the oriented line graph can be used to reduce or even completely lift the cospectrality of certain classes of graph, including trees and strongly regular graphs [15]. This points to the fact that one potentially profitable route to improving methods from spectral graph theory may reside in graph transformation.

Although the Ihara zeta function has been widely investigated in the mathematics literature [5, 44, 51, 68, 70, 97], it has received little attention as a means of characterizing graphs in machine learning. Furthermore, to be rendered tractable for real world problems in pattern recognition, the issue of how to generate stable pattern vectors from the Ihara zeta function must be addressed. Zhao et al. [100] have recently used Savchenko's formulation of the zeta function [69], expressed in terms

of cycles, to generate merge weights for clustering over a graph-based representation of pairwise similarity data. Their formulation is based on a representation of oriented line graphs, which is an intermediate step in the development of the Ihara zeta function studied in this work. Watanabe et al. [89] have presented an approach to the analysis of Loopy Belief Propagation (LBP) by establishing a formula that connects the Hessian of the Bethe free energy with the edge Ihara zeta function.

In this chapter, we turn to the Ihara zeta function as a tool for structure-preserving embedding. The motivation here is twofold: first, the Ihara zeta function is determined by cycle frequencies, and thus capable of reflecting graph topologies; second, the Ihara zeta function can be expressed in a polynomial form of a transformed graph such that certain polynomial and spectral analysis can be done based on it. These properties of the Ihara zeta function allow it to naturally incorporate topologies, spectra and polynomials into a unifying representation, and thus enable it to have the potential to result in a rich family of structural characteristics.

6.1.3 Graph Representations for Pattern Recognition

This section reviews the various graph representations used in pattern recognition, not restricted to graph characterization. Graph-based methods are widely used in solving problems in computer vision and pattern recognition at different levels of feature abstraction. Early work related to graph-based representations focuses on identifying subgraph isomorphism [85] or measuring edit distance [67] for the purpose of structural pattern recognition. These methods enumerate the node attributes to obtain an optimal solution to certain cost functions. Therefore, graphs are not characterized in a mathematically consistent way by using these methods. However, this shortcoming can be overcome by adopting graph spectral methods [20] for graph characterization. In addition to representing graphs in terms of vertex set and edge set, another graph representation used in spectral graph theory is adjacency matrix or Laplacian matrix. Each entry of the matrix is associated with the pairwise relationship between two vertices, and the indices of the entry represent labels for the two vertices. By using the matrix representations, graphs can be processed in a computationally efficient and consistent way, because existing computing algorithms for matrices can be straightforwardly applied to graphs. Therefore, many statistical pattern recognition algorithms can directly work on graph-based data once the matrix representations are established. One good example is to formulate the problem of clustering as that of computing the principal eigenvector of the normalized affinity matrix for a graph [90]. Furthermore, Zass et al. [98] have shown how to provide a probabilistic interpretation for this formulation by developing a completely positive factorization scheme. On the other hand, Shi et al. have [77] presented a method based on the normalized Laplacian matrix rather than the normalized affinity matrix. Their method is referred to as normalized cut because it is capable of balancing the cut and the association. Robles-Kelly et al. [59] have introduced a probabilistic framework based on a Bernoulli model which adopts EM

algorithm for extracting the leading eigenvector as the cluster membership indicators. Pavan and Pelillo [53] have formulated the problem of pairwise clustering as that of extracting the dominant set of vertices from a graph (see also Chap. 8). Based on this notion, Rota Bulò et al. have developed game-theoretic approaches [4, 12] to partial clique enumeration [62] and hypergraph clustering [63, 64]. Qiu et al. [54] have characterized the random walk on a graph using the commute time between vertices and proved that the commute time matrix is a more robust measure of the proximity of data than the raw proximity matrix. Behmo et al. [8] have exploited the formulation based on commute times as a manner of image representation. Furthermore, some researchers have investigated the problem of graph based learning by incorporating the path-based information between vertices as a replacement of pairwise similarity. Representative work includes Path-Based Clustering [29] and the sum-over-paths covariance kernel [49].

Different from clustering graph vertices, the research on graph embedding aims to seek a low dimensional coordinates for the vertices. This is often conducted in a manifold learning scenario, where certain local features of the manifold underlying the original data are preserved. Based on a similar notion to normalized cut, Belkin et al. [7] have presented a graph embedding framework called Laplacian eigenmaps for dimensionality reduction. Other notable manifold learning methods include ISOMAP [82] and LLE [66]. These manifold learning methods adopt different cost functions and thus result in different local structure preservations. Recently, Yan et al. [94] have generalized traditional embedding methods such as PCA by using a graph embedding framework and extended it into non-negative versions [47, 88, 96]. Shaw et al. [75] have introduced an embedding strategy which preserves the global topological properties of an input graph.

A preliminary step for all these graph-based methods (both for clustering and embedding) is to establish a graph over the training data. Data samples are represented as vertices of the graph and the edges represent the pairwise relationships between them. The methods for establishing a graph and measuring vertex similarities (i.e., edge weights) have a great influence on the subsequent graph-based learning algorithms. Therefore, the process of graph construction has recently attracted much research interest [23, 41, 50] as it remains only partially solved.

In addition to representing the pairwise relationship within a training data set (i.e., normalized cut, ISOMAP and LLE), graph-based methods also play an important role in learning with structured data. Problems of this kind arise when training data are not represented in vectors but in terms of relational structures such as trees and graphs. In this case, learning algorithms which admit structured data are needed. For example, the problem of discovering shape and object categories is frequently posed as one of clustering a set of graphs. This is an important process since it can be used to organize large databases of graphs in a manner that renders retrieval efficiency [71].

The strategies for learning with structured data can be roughly classified into two categories. The first is the graph characterization methods reviewed in Sect. 6.1.2. The second is to develop specific learning algorithms which admit individual graphs or trees as input. For the second category, a similarity between structured data samples is defined and traditional learning schemes are applied based on the pairwise

similarities between structured data samples. For example, graph similarities can be computed by using tree or graph edit distance and structured objects are assigned to classes using pairwise clustering [14, 83]. However, the use of pairwise similarities alone is a rather crude way to capture the modes of variation present in graphs of a particular class. Moreover, it requires the computation of vertex correspondences which is sometimes an unreliable process. The graph kernels [55, 87] overcome this problem to a certain degree by naturally incorporating vertex correspondences into the process of learning. This is effected by the learning process in which every pair of vertices drawn separately from two graphs are compared to obtain an entry of the kernel matrix. However, the process of vertex enumeration gives rise to computational inefficiency. Although fast computational scheme [76] has recently been proposed, these methods still undergo heavy computational overheads. In contrast to the graph kernel strategies, graph characterization methods would be more efficient if pattern vectors are suitably established, because it avoids enumerating the comparisons between every pair of vertices.

6.1.4 Hypergraph Representations for Pattern Recognition

There has recently been an increasing interest in hypergraph-based methods for representing and processing structures where the relations present are not simply pairwise [19, 122, 123]. The main reason for this trend is that hypergraph representations allow vertices to be multiply connected by hyperedges and can hence capture multiple relationships between features. Due to their effectiveness in representing multiple relationships, hypergraph-based methods have been applied to various practical problems such as partitioning netlists [33] and clustering categorical data [30]. For visual processing, to the best of our knowledge, the first attempt at representing visual objects using hypergraphs dates back to Wong et al.'s [93] framework for 3D object recognition. In this work, a 3D object model based on a hypergraph representation is constructed, and this encodes the geometric and shape information with polyhedrons as vertices and hyperedges. Object synthesis and recognition tasks are performed by merging and partitioning the vertex and hyperedge set. The method is realized using set operations and the hypergraphs are not characterized in a mathematically consistent way. Later, Bretto et al. [10] introduced a hypergraph model for image representation, where they successfully and simultaneously solved the problems of image segmentation, noise reduction and edge detection. However, their method also relies on a crude form of set manipulation. Agarwal et al. [1] have performed visual clustering by partitioning a weighted graph transformed from the original hypergraph by a weighted sum of its hyperedges into the graph edge. Recently, Rota Bulò et al. [61] have established a hypergraph model for estimating affine parameters in vision problems. Bunke et al. [13] have developed a hypergraph matching algorithm for object recognition, where consistency checks are conducted on hyperedges. The computational paradigm underlying their method is based on tree search operations. Zass et al. [99] and Duchenne et al. [24] have separately applied high-degree affinity arrays (i.e., tensors) to formulating hypergraph matching

problems up to different cost functions. Both methods address the matching process in an algebraic manner but must undergo intractable computational overheads if hyperedges are not suitably sampled. Shashua et al. [73, 74] have performed visual clustering by adopting tensors for representing uniform hypergraphs (i.e., those for which the hyperedges have identical cardinality) extracted from images and videos. Their work have been complemented by He et al.'s algorithm for detecting number of clusters in tensor-based framework [38]. Similar methods include those described in [16–18, 31, 63, 64], in which tensors (uniform hypergraphs) have been used to represent the multiple relationships between objects. Additionally, tensors have recently been used to generalize dimensionality reduction methods based on linear subspace analysis into higher orders [36, 37, 86, 95]. However, the tensor representation considers all possible permutations of a subset of vertices and establishes hyperedges with cardinality consistent with the relational order. Therefore, tensors can only represent uniform hypergraphs, and are not suited for nonuniform hypergraphs (i.e., hypergraphs with varying hyperedge cardinalities).

One common feature of these existing hypergraph representations is that they exploit domain specific and goal directed representations. Specifically, most of them are confined to uniform hypergraphs and do not lend themselves to generalization. The reason for this lies in the difficulty in formulating a nonuniform hypergraph in a mathematically neat way for computation. There has yet to be a widely accepted and consistent way for representing and characterizing nonuniform hypergraphs, and this remains an open problem when exploiting hypergraphs for machine learning. Moreover, to be easily manipulated, hypergraphs must be represented in a mathematically consistent form, using structures such as matrices or vectors.

Since Chung's [21] definition of the Laplacian matrix for K -uniform hypergraphs, there have been several attempts to develop matrix representations of hypergraphs. To establish the adjacency matrix and Laplacian matrix for a hypergraph, an equivalent graph representation is often required. Once the graph approximation is at hand, its graph representation matrices (e.g., the adjacency matrix (6.1) and the Laplacian matrix (6.3)) are referred to as the corresponding hypergraph representation matrices. It is based on these approximate matrix representations that the subsequential processes of hypergraphs (e.g., high order clustering and matching) take place. Agarwal et al. [2] have compared a number of alternative graph representations [9, 30, 46, 60, 101] for hypergraphs and explained their relationships with each other in machine learning. One common feature for these methods, as well as the method in [1], is that a weight is assumed to be associated with each hyperedge. Additionally, the graph representations for a hypergraph can be classified into two categories: (a) the clique expansion [1, 9, 30, 60] and (b) the star expansion [46, 101]. The clique expansion represents a hypergraph by constructing a graph with all pairs of vertices within a hyperedge connecting to each other. The star expansion represents a hypergraph by introducing a new vertex to every hyperedge, and constructing a graph with all vertices within a hyperedge connecting to the newly introduced vertex. In both strategies, each edge in each individual graph representation is weighted in a manner determined by the corresponding hyperedge weight in a task-specific way that is different from others. Moreover, these

graph-based representations for hypergraphs are just approximations and give rise to information loss, as reported in [1]. This deficiency may result in ambiguities when the approximation methods are used to distinguish structures with different relational orders.

To address these shortcomings, an effective matrix representation for hypergraphs is needed, such that the ambiguities of relational order can be overcome. To this end, trivial graph approximations should be avoided for hypergraph representation. In the mathematics literature, the definitions of the Ihara zeta function has recently been extended from graphs to hypergraphs [81]. In the determinant form of the Ihara zeta function, a graph representation is also used for describing the hypergraph. However, this graph representation uses color edges to capture the hyperedge connectivity and does not result in information loss regarding relational order. We will make a polynomial analysis of the hypergraph Ihara zeta function and develop a family of features that readily characterize hypergraphs into a feature space suitable for hypergraph clustering.

6.2 Spherical Embedding

In this section, we detail work undertaken in SIMBAD aimed at developing a curvature-dependent embedding of dissimilarity data. The idea is to embed the data on an hypersphere of optimal radius, and then perform pattern analysis tasks such as variance analysis and classification in the tangent space to the sphere.

6.2.1 Indefinite Spaces

We begin with the assumption that we have a set of objects of interest and have measured a set of dissimilarities or distances between all pairs of objects in our problem. This is denoted by the matrix \mathbf{D} , where D_{ij} is the distance between objects i and j . We can define an equivalent set of similarities by using the matrix of squared distances \mathbf{D}' , where $D'_{ij} = D_{ij}^2$. This is achieved by identifying the similarities as $-\frac{1}{2}\mathbf{D}'$ and centering the resulting matrix:

$$\mathbf{S} = -\frac{1}{2}\left(\mathbf{I} - \frac{1}{n}\mathbf{J}\right)\mathbf{D}'\left(\mathbf{I} - \frac{1}{n}\mathbf{J}\right). \quad (6.4)$$

Here \mathbf{J} is the matrix of all-ones, and n is the number of objects. In a Euclidean space, this procedure gives exactly the inner-product or kernel matrix for the points.

If \mathbf{S} is positive semi-definite, then the original distances are Euclidean and we can use the kernel embedding to locate positions \mathbf{x}_i for the points in a Euclidean space as follows:

$$\mathbf{S} = \mathbf{U}_S \mathbf{A}_S \mathbf{U}_S^T = \mathbf{X}\mathbf{X}^T, \quad (6.5)$$

$$\mathbf{X} = \mathbf{U}_S \mathbf{A}_S^{\frac{1}{2}}, \quad (6.6)$$

where \mathbf{U}_S and \mathbf{A}_S are the eigenvector and eigenvalue matrices of \mathbf{S} , respectively. The position-vector \mathbf{x}_i of the i th point corresponds to the i th row of \mathbf{X} .

If \mathbf{S} is indefinite, which is often the case, then the objects cannot exist in a Euclidean space with the given distances. This does not necessarily mean the distances are non-metric; metricity is a separate issue. One measure of the deviation from definiteness which has proved useful is the negative eigenfraction (NEF) which measures the fractional weight of eigenvalues which are negative [109]:

$$\text{NEF} = \frac{\sum_{\lambda_i < 0} |\lambda_i|}{\sum_i |\lambda_i|}. \quad (6.7)$$

If $\text{NEF} = 0$, then the data is definite and can be represented by points in a Euclidean space. We can measure the *non-metricity* of the data by counting the number of violations of metric properties. It is very rare to have an initial distance measure which gives negative distance, so we will assume that the distances are all positive. The two measures of interest are then the fraction of triples which violate the triangle inequality (TV) and the degree of asymmetry of the distances (γ) [103]:

$$\gamma = \sum_{i \neq j} \frac{|\tilde{d}(i, j) - \tilde{d}(j, i)|}{|\tilde{d}(i, j) + \tilde{d}(j, i)|}, \quad (6.8)$$

where $\tilde{d}(\cdot, \cdot)$ is the dissimilarity scaled so that the average dissimilarity is one.

If the data is metric (or, in practice, close to metric) but indefinite then we must use a curved space to embed the points.

6.2.2 Spherical Space

A spherical space is an example of a Riemannian manifold. On the manifold, distances are measured by geodesics (the shortest curve between points), and geodesic distances are metric. Spherical space is curved, however, and so the distances are fundamentally non-Euclidean and in general the similarity matrix of points in spherical space will be indefinite. This makes it a potential choice for representing non-Euclidean datasets.

A manifold embedding is important because it allows the use of geometric and statistical tools on the embedded points. On a Riemannian manifold, distances are defined between any pair of points in the manifold in a consistent way (not just between the sample data-points). Geodesic distance is defined as the length of the shortest curve which joins two points (the curve is known as a geodesic), and is a metric. Geodesics are the equivalent of straight lines in Euclidean space, and allow us to construct a geometry in curved space. We can also compute statistics such as the mean in a way consistent with the normal Euclidean definition. This means that

all the standard classifiers can be applied (at least in theory) to the data, but the exact formulation will differ from vector-space classifiers.

The spherical manifold in 2D is isomorphic to the 2D surface of a sphere embedded in 3D space, which has a well-known parametric form. Here r is the radius of the sphere, u is the azimuth angle and v is the zenith angle;

$$\mathbf{x} = (r \sin u \sin v, r \cos u \sin v, r \cos v)^T. \quad (6.9)$$

This geometry generalizes to an $(n - 1)$ -dimensional hypersphere embedded in an n -dimensional Euclidean space. The surface can be defined implicitly using the constraint

$$\sum_i x_i^2 = r^2 \quad (6.10)$$

where r is the radius of the hypersphere. This surface is curved and has a constant sectional curvature of $K = 1/r^2$ everywhere.

The geodesic distance between two points in curved space is the length of the shortest curve lying in the space and joining the two points. On the hypersphere, the geodesic is a great circle. The distance is the length of the arc of the great circle which joins the two points. If the angle subtended by two points at the center of the hypersphere is θ_{ij} , then the distance between them is

$$d_{ij} = r\theta_{ij}. \quad (6.11)$$

With the coordinate origin at the center of the hypersphere, we can represent a point by a position vector \mathbf{x}_i of length r . Since the inner product is $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = r^2 \cos \theta_{ij}$, we can also write

$$d_{ij} = r \cos^{-1} \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{r^2}. \quad (6.12)$$

6.2.3 The Exponential Map

Our procedure for embedding points on a sphere requires one important tool of Riemannian geometry, which is the exponential map. The exponential map is a map from points on the manifold to points on a tangent space of the manifold. As the tangent space is flat (i.e., Euclidean), we can calculate quantities in a straightforward way. The map has an origin, which defines the point at which we construct the tangent space of the manifold. The formal definition of the exponential map is the map which connects the Lie algebra on the tangent space to the Lie group which defines the manifold. We will not concern ourselves with the technical details here, but the map has an important property which simplifies geometric computations; the geodesic distance between the origin of the map and a point on the manifold is the same as the Euclidean distance between the images of the two points on the tangent space. Formally, the definition of these properties as follows: Let T_M be the tangent

space at some point M on the manifold, P be a point on the manifold and X a point on the tangent space. We have

$$X = \text{Log}_M P, \quad (6.13)$$

$$P = \text{Exp}_M X, \quad (6.14)$$

$$d_g(P, M) = d_e(X, M). \quad (6.15)$$

The Log and Exp notation defines a log-map from the manifold to the tangent space and an exp-map from the tangent space to the manifold. This is a formal notation and does not imply the normal log and exp functions—although they do coincide for some types of data, they are not the same for the spherical space. M is the origin of the map and is mapped onto the origin of the tangent space. The distance $d_g(\cdot, \cdot)$ is the geodesic distance on the manifold and $d_e(\cdot, \cdot)$ the Euclidean distance on the tangent space.

For the spherical manifold, the exponential map is as follows. We define a point P on our manifold as a position vector \mathbf{p} with length r (the origin is at the center of the hypersphere). Similarly, the point M is represented by the vector \mathbf{m} , and M is the origin of the map. The maps are then

$$\mathbf{x} = \frac{\theta}{\sin \theta} (\mathbf{p} - \mathbf{m} \cos \theta), \quad (6.16)$$

$$\mathbf{p} = \mathbf{m} \cos \theta + \frac{\sin \theta}{\theta} \mathbf{x}, \quad (6.17)$$

$$d_g(P, M) = d_e(X, M) = |\mathbf{x}| = r\theta, \quad (6.18)$$

where $\theta = \cos^{-1} \langle \mathbf{p}, \mathbf{m} \rangle / r^2$. The vector \mathbf{x} is the image of P in the tangent space, and the image of M is at the origin of the tangent space.

6.2.4 Spherical Embedding

Given a dissimilarity matrix \mathbf{D} , we want to find the embedding of a set of points on the surface of a hypersphere of radius r , such that the geodesic distances are as similar as possible to \mathbf{D} . Unfortunately, this appears to be a hard problem, and therefore we use an approximate optimization-based approach. We simplify the problem by considering just the distances to a single point at a time. Let the point of interest be \mathbf{p}_i ; we then want to find a new position for this point on the hypersphere such that the geodesic distance to point j is d_{ij}^* where $*$ denotes that this is the target distance. We formulate the estimation of position as a least-squares problem which minimizes

$$E = \sum_{j \neq i} (d_{ij}^2 - d_{ij}^{*2})^2, \quad (6.19)$$

where d_{ij} is the actual distance between the points. This is a similar formulation to [105] and other approaches to non-Euclidean multidimensional scaling, who seek to minimize the ‘stress’. Direct optimization on the sphere is complicated by the need to restrict points to the manifold. However, as we are considering a single point at a time, we can construct a linear embedding using the log-map and optimize in the Euclidean space. This is a different approach to that of [105]. If the current point-positions on the hypersphere are $\mathbf{p}_j, \forall j$, we can use the log-map to obtain point-positions for each object in the tangent space of $\mathbf{x}_j, \forall j$ as follows:

$$\mathbf{x}_j = \text{Log}_{\mathbf{p}_i} \mathbf{p}_j = \frac{\theta_{ij}}{\sin \theta_{ij}} (\mathbf{p}_j - \mathbf{p}_i \cos \theta_{ij}) \quad (6.20)$$

with $\mathbf{x}_i = 0$.

We have found standard optimization schemes to be infeasible on larger datasets, so here we propose a gradient descent scheme with optimal step-size. In this iterative scheme, we update the position of the point \mathbf{x}_i in the tangent space to obtain a better fit to the given distances. At iteration k , the point is at position $\mathbf{x}_i^{(k)}$. Initially, the point is at the origin, so $\mathbf{x}_i^{(0)} = 0$. Since the points lie in tangent space, which is Euclidean, we then have $d_{ij}^2 = (\mathbf{x}_j - \mathbf{x}_i)^T (\mathbf{x}_j - \mathbf{x}_i)$ and gradient of the error is

$$\nabla E = 4 \sum_{j \neq i} (d_{ij}^2 - d_{ij}^{*2}) (\mathbf{x}_i - \mathbf{x}_j), \quad (6.21)$$

and our iterative update procedure is

$$\mathbf{x}_i^{(k+1)} = \mathbf{x}_i^{(k)} + \eta \nabla E. \quad (6.22)$$

Finally, we can determine the optimal step size as follows: let $\Delta_j = d_{ij}^2 - d_{ij}^{*2}$ and $\alpha_j = \nabla E^T (\mathbf{x}_i - \mathbf{x}_j)$, then the optimal step size is the smallest root of the cubic

$$n |\nabla E|^2 \eta^3 + 3 |\nabla E|^2 \left(\sum_j \alpha_j \right) \eta^2 + \left(2 \sum_j \alpha_j^2 + |\nabla E|^2 \sum_j \Delta_j \right) \eta + \sum_j \alpha_j \Delta_j. \quad (6.23)$$

This step-size is optimal in the sense that it minimizes the error in the direction of the gradient.

After finding a new point position \mathbf{x}_i , we apply the exp-map to locate the new point position on the spherical manifold

$$\mathbf{p}'_i = \mathbf{p}_i \cos \theta + \frac{\sin \theta}{\theta} \mathbf{x}_i. \quad (6.24)$$

6.2.4.1 Classifiers in the Manifold

As well as embedding distances on the spherical manifold, it is important to be able to perform operations such as classification in the manifold. Some classifiers are

trivially implemented on a spherical manifold, for example, the nearest-neighbors (NN). Others which utilize geometry must be modified to incorporate the non-Euclidean geometry of curved space. Here we discuss the nearest mean classifier (NMC) in a non-flat manifold.

The *intrinsic mean* of a set of points on the manifold may be computed via the generalized mean [116]

$$\bar{P} = \arg \min_P \sum_i d_g(P, P_i). \quad (6.25)$$

We can solve for the mean of a set of points in a manifold using the following iterative procedure involving the exponential map [116]:

$$\mathbf{m}^{(k+1)} = \text{Exp}_{\mathbf{m}^{(k)}} \frac{1}{n} \sum_i \text{Log}_{\mathbf{m}^{(k)}} \mathbf{P}_i. \quad (6.26)$$

While the convergence of this process is not guaranteed in a general manifold, it is well behaved on the hypersphere [116]. As a result, we can compute the means of each class $\mathbf{m}_1, \dots, \mathbf{m}_C$ and implement the NMC:

$$c^* = \arg \min_c \left[r \cos^{-1} \frac{\langle \mathbf{x}, \mathbf{m}_c \rangle}{r^2} \right]. \quad (6.27)$$

6.2.5 Experimental Results

We have applied our embedding method to a number of indefinite datasets. These are summarized in Table 6.1, along with their degree of indefiniteness, as measured by the negative eigenfraction (Eq. (6.7)). These datasets are produced by dissimilarity measures applied to a variety of real world problems. The Coil datasets are produced by graph-matching algorithms applied to corner-graphs of some of the objects in the COIL database [52, 119], using graduated assignment [118] (CoilYork) and the JoEig approach [110] (CoilDelftDiff and CoilDelftSame). The CatCortex data gives the similarity between different cortical regions in terms of connectivity [111]. The DelftGestures dataset consists of the dissimilarities computed from a set of gestures in a sign-language using a dynamic time warping procedure [112]. The FlowCyto series of datasets is based on the L_1 -norm dissimilarities between flow-cytometer histograms of breast cancer tissues. The data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000–2004. Newsgroups is a small subset of the 20 Newsgroups data of Roweis. The ProDom dataset is a set of dissimilarities derived from the structural matching of protein domain sequences [113]. WoodyPlants50 is a dataset of shape dissimilarities between plant leaves [114]. The Zongker dissimilarities are based on deformable template matching between 2000 handwritten digits in 10 classes [115]. Finally, the Chickenpiece dataset is another set of shape dissimilarities derived from

Table 6.1 Properties of datasets used

Dataset	Size	NEF	Triangle violations	Asymmetry
CoilYork	288	0.258	1/23639616	0.009
DelftGestures	1500	0.308	14798/3368253000	0
FlowCyto-1	612	0.275	272052/228098520	0
FlowCyto-2	612	0.268	161517/228098520	0
FlowCyto-3	612	0.275	272879/228098520	0
FlowCyto-4	612	0.272	268991/228098520	0
NewsGroups	600	0.202	4643/214921200	0
Chickenpieces-5	446	0.216	0/88120680	0.044
Chickenpieces-10	446	0.257	1/88120680	0.046
Chickenpieces-15	446	0.286	74/88120680	0.051
Chickenpieces-20	446	0.307	695/88120680	0.057
Chickenpieces-25	446	0.320	1375/88120680	0.063
Chickenpieces-30	446	0.331	3188/88120680	0.067
Chickenpieces-35	446	0.339	4834/88120680	0.073
Chickenpieces-40	446	0.345	7549/88120680	0.076
CatCortex	65	0.272	286/262080	0
CoilDelftDiff	288	0.128	1/23639616	0
CoilDelftSame	288	0.027	0/23639616	0
WoodyPlants50	791	0.229	115253/493038210	0
ProDom	2604	0.043	136/17636907624	0
Zongker	2000	0.419	6583656/7988004000	0.051

string-edit distance on the contours of chicken piece silhouettes [103]. This data has a number of controllable parameters which influence the indefinite nature of the dissimilarities. Here we use an edit cost of 45 and a variety of contour lengths (5, 10, 15, 20, 25, 30, 35, 40).

We characterize the accuracy of our embeddings in two different ways. Firstly, we measure the RMS fractional error of the embedded distances:

$$\text{RMS Error} = \left(\frac{1}{n} \sum_{ij} \frac{D_{ij} - D_{ij}^*}{\bar{D}} \right), \quad (6.28)$$

where \bar{D} is the average dissimilarity between objects in the original data. Secondly, we measure the 1NN classifier error, both before and after embedding. This demonstrates whether the embedding preserves the local structure of the classes adequately. In the final column, we show the performance of the NMC classifier on the hypersphere.

Table 6.2 Embedding results for the datasets (in order of increasing error)

Dataset	Size	1NN (orig)	Error	Radius	1NN (emb)	NMC
Newsgrroups	600	0.269 ± 0.015	0.022	0.6298	0.279 ± 0.012	0.208 ± 0.015
CoilDelftDiff	288	0.487 ± 0.033	0.030	0.0277	0.479 ± 0.022	0.467 ± 0.034
Chickenpieces-5	446	0.350 ± 0.022	0.030	66.9	0.417 ± 0.022	0.407 ± 0.02
WoodyPlants50	791	0.101 ± 0.008	0.034	0.4362	0.147 ± 0.015	0.197 ± 0.016
Chickenpieces-10	446	0.170 ± 0.016	0.039	33.4	0.249 ± 0.018	0.338 ± 0.022
DelftGestures	1500	0.042 ± 0.0048	0.039	3.9826	0.135 ± 0.009	0.104 ± 0.004
Chickenpieces-15	446	0.079 ± 0.011	0.049	20.73	0.116 ± 0.018	0.249 ± 0.028
Chickenpieces-20	446	0.069 ± 0.012	0.052	17	0.109 ± 0.011	0.202 ± 0.022
Chickenpieces-25	446	0.048 ± 0.01	0.057	13.1	0.086 ± 0.013	0.21 ± 0.025
FlowCyto-2	612	0.366 ± 0.019	0.059	12132	0.378 ± 0.017	0.389 ± 0.028
Chickenpieces-30	446	0.048 ± 0.009	0.062	11.01	0.091 ± 0.013	0.197 ± 0.015
CoilYork	288	0.278 ± 0.025	0.063	177.8	0.307 ± 0.024	0.471 ± 0.029
FlowCyto-3	612	0.413 ± 0.013	0.072	13078	0.421 ± 0.021	0.4 ± 0.015
Chickenpieces-35	446	0.065 ± 0.011	0.073	10.12	0.069 ± 0.007	0.178 ± 0.023
Chickenpieces-40	446	0.087 ± 0.014	0.078	8.14	0.099 ± 0.012	0.2 ± 0.015
FlowCyto-1	612	0.369 ± 0.013	0.078	12794	0.425 ± 0.008	0.385 ± 0.02
CatCortex	65	0.095 ± 0.034	0.084	2.33	0.111 ± 0.04	0.047 ± 0.025
FlowCyto-4	612	0.425 ± 0.023	0.090	11761	0.413 ± 0.018	0.436 ± 0.026
ProDom	2604	0.002 ± 0.001	0.122	471.1	0.038 ± 0.003	0.21 ± 0.011
CoilDelftSame	288	0.636 ± 0.031	0.134	0.0577	0.674 ± 0.040	0.433 ± 0.038
Zongker	2000	0.372 ± 0.016	0.233	0.2887	0.043 ± 0.005	0.109 ± 0.009

The result in Table 6.2 show that we obtain an accuracy spherical embedding for nearly all the data. Of the 21 datasets, only three have more than 10 % RMS error on the embedding. This demonstrates the effectiveness of our embedding technique at locating optimal embeddings. For ten of the datasets, we see virtually identical 1NN performance both before and after embedding, and for one a large improvement (Zongker). We do not know the cause of this unexpected behavior, but it seems to be a feature of this particular dataset. For the other ten sets, we see deterioration in the 1NN classification, indicating that the local structure has been changed somewhat. This is particularly evident in the Chickenpieces data, for which six of the eight examples give worse 1NN scores. It seems that this data series is unsuitable for spherical embedding.

The NMC classifier shows a far wider range of performance. The Chickenpieces data series, CoilYork, WoodyPlants50 and ProDom show a substantially worse performance with the NMC than with the original 1NN classifier, whereas Newsgrroups, CatCortex, CoilDelftSame and Zongker show a substantial improvement.

6.2.6 Section Summary

In this section, we have shown how spherical embedding can be used as a solution to the problem of indefinite, non-Euclidean dissimilarities. This embedding preserves some of the non-Euclidean nature of the dissimilarities which may be important in other tasks such as classification. We developed an optimization-based procedure for embedding objects on hyperspherical manifolds which uses the Lie group representation of the hypersphere and its associated Lie algebra to define the exponential map between the manifold and its local tangent space. The optimization is then solved locally in Euclidean space. This process is efficient enough to allow us to embed datasets of several thousand objects. We also defined the nearest mean classifier on the manifold.

Experiments on a variety of non-Euclidean datasets show that we can obtain accurate embeddings representing the dissimilarities on the hypersphere. The classification results show that the embedding of some datasets is very useful (for example, the Newsgroups data), and for others not effective (the Chickens data).

6.3 Embeddings from the Ihara Zeta Function

In this section, we will illustrate an alternative use of embedding methods and will show how to characterize and embed irregular unweighted hypergraphs using Ihara coefficients. The proposed hypergraph representation proves to be a flexible tool in learning the structure of irregular unweighted hypergraphs with different relational orders. Our contributions are two-fold. First, we propose a vectorial representation, which naturally avoids the ambiguity induced by the matrix representations such as the hypergraph Laplacian, for irregular unweighted hypergraphs. We construct pattern vectors using the Ihara coefficients, i.e., the characteristic polynomial coefficients extracted from Ihara zeta function for hypergraphs. Second and more importantly, we propose an efficient method for computing the Ihara coefficient set, which renders the computation of the coefficients tractable. We use the pattern vectors consisting of Ihara coefficients for clustering hypergraphs extracted from images of different object views and demonstrate their effectiveness in hypergraph characterization.

6.3.1 Hypergraph Laplacian

A hypergraph is a generalization of a graph. Unlike the edge of a graph, which can connect only two vertices, the hyperedge in a hypergraph can connect any number of vertices. A hypergraph is normally defined as a pair $H(V, E_H)$ where V is a set of elements, called nodes or vertices, and E_H is a set of non-empty subsets of V called hyperedges. The representation of a hypergraph in the form of sets, concretely captures the relationship between vertices and hyperedges. However, it is

difficult to manipulate this form in a computationally uniform way. Thus one alternative representation of a hypergraph is in the form of a matrix. For a hypergraph $H(V, E_H)$ with I vertices and J hyperedges, we establish an $I \times J$ matrix \mathbf{H} which is referred to as the incidence matrix of the hypergraph. \mathbf{H} has element $h_{i,j}$ equal to 1 if $v_i \in e_j$ and 0 otherwise.

The incidence matrix can be more easily manipulated than its equivalent set representation. To obtain a vertex-to-vertex representation, we need to establish the adjacency matrix and Laplacian matrix for a hypergraph. To this end, a graph representation for the hypergraph is required. Agarwal et al. [2] have classified the graph representations for a hypergraph into two categories, namely (a) the clique expansion and (b) the star expansion. The clique expansion represents a hypergraph by constructing a graph with all the pairs of vertices within a hyperedge connecting each other. The star expansion represents a hypergraph by introducing a new vertex to every hyperedge and constructing a graph with all vertices within a hyperedge connecting the newly introduced vertex. The common feature of these methods is that each edge in a graph representation is weighted in terms of the corresponding hyperedge weight subject to certain conditions. For example, the normalized Laplacian matrix $\hat{\mathbf{L}}_H = \mathbf{I} - \mathbf{D}_v^{-1/2} \mathbf{H} \mathbf{D}_e \mathbf{H}^T \mathbf{D}_v^{-1/2}$ introduced in [117] is obtained from the star expansion of a hypergraph, and its individual edges are weighted by the quotient of the corresponding hyperedge weight and cardinality. Here \mathbf{D}_v is the diagonal vertex degree matrix whose diagonal element $d(v_i)$ is the summation of the i th row of \mathbf{H} , \mathbf{D}_e is the diagonal vertex degree matrix whose diagonal element $d(e_j)$ is the summation of the j th column of \mathbf{H} , and \mathbf{I} is a $|V| \times |V|$ identity matrix. In this case, even edges derived from an unweighted hyperedge are assigned a nonunit weight. On the other hand, rather than attaching a weight to each edge in the graph representation, the adjacency matrix and the associated Laplacian matrix for an irregular unweighted hypergraph can be defined as $\mathbf{A}_H = \mathbf{H} \mathbf{H}^T - \mathbf{D}_v$ and $\mathbf{L}_H = \mathbf{D}_v - \mathbf{A}_H = 2\mathbf{D}_v - \mathbf{H} \mathbf{H}^T$, respectively [56]. In practice, these two definitions are obtained in terms of the clique expansion without attaching a weight to a graph edge. The eigenvalues of \mathbf{L}_H are referred to as the hypergraph Laplacian spectrum and can be used in a straightforward way as hypergraph characteristics.

Although the vertex-to-vertex matrix representations for hypergraphs described above naturally reduce to those for graphs when the relational order is two, there are deficiencies for these representations in distinguishing relational structures. When relational structures have the same vertex cardinality but different relational orders, these vertex-to-vertex matrix representations become ambiguous. For example, for the graph in Fig. 6.1(a) and the hypergraph in Fig. 6.1(b), the adjacency matrices of the two hypergraphs are identical, and so are the associated Laplacian matrices. The adjacency matrix and Laplacian matrix are as follows:

$$\mathbf{A}_H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad \mathbf{L}_H = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

It is clear that the unweighted adjacency matrix and Laplacian matrix cannot distinguish these two hypergraphs. The reason for this deficiency is that the adjacency

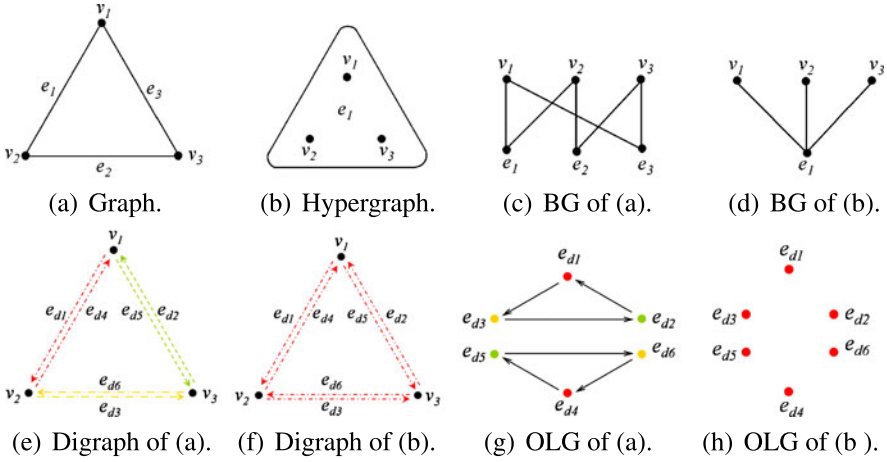


Fig. 6.1 Hypergraph examples and their graph representations

matrix and the Laplacian matrix only record the adjacency relationships between pairs of nodes and neglect the cardinalities of the hyperedges. In this regard, they induce certain information loss in representing relational structures and cannot always distinguish between pairwise relationships and high order relationships for the same set of vertices. The normalized Laplacian matrices for Figs. 6.1(a) and 6.1(b) are \hat{L}_{H1} and \hat{L}_{H2} , respectively:

$$\hat{L}_{H1} = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix}, \quad \hat{L}_{H2} = \begin{pmatrix} 1/2 & -1/4 & -1/4 \\ -1/4 & 1/2 & -1/4 \\ -1/4 & -1/4 & 1/2 \end{pmatrix}.$$

Since $\hat{L}_{H2} = \frac{3}{4}\hat{L}_{H1}$, the eigenvalues of \hat{L}_{H2} are found by scaling those of \hat{L}_{H1} by a factor $3/4$, and both matrices have the same eigenvectors. Thus the normalized Laplacian matrices for different hypergraphs may yield spectra that are just scaled relative to each other. This hinders the hypergraph characterization when the eigenvectors are used. One important reason for the limited usefulness of the above hypergraph matrix representations is that they result in information loss when relational orders of varying degree are present. To overcome this deficiency, we use characteristic polynomials extracted from the Ihara zeta function as a means of representing hypergraphs. In the next section, we commence by showing that the Ihara zeta function can be used to represent this type of relational structure in hypergraphs. We use the Ihara coefficients, i.e., the characteristic polynomial coefficients extracted from the Ihara zeta function, as hypergraph characteristics. We show that the Ihara coefficients not only encode the relational structural in a consistent way but also overcome the deficiencies listed above.

6.3.2 Ihara Zeta Function from Graphs to Hypergraphs

The rational expression of the Ihara zeta function for a graph is as follows [6]:

$$Z_G(u) = (1 - u^2)^{\chi(G)} \det(\mathbf{I}_{|V(G)|} - u\mathbf{A} + u^2\mathbf{Q})^{-1}, \quad (6.29)$$

where $\chi(G) = |V| - |E|$, \mathbf{A} is the adjacency matrix of the graph, and $\mathbf{Q} = \mathbf{D} - \mathbf{I}_{|V(G)|}$ where $\mathbf{I}_{|V(G)|}$ is the identity matrix and \mathbf{D} is the degree matrix, which can be generated by placing the column sums as the diagonal elements while setting the off-diagonal elements to zero.

To formulate the Ihara zeta function for a hypergraph in a similar form with (6.29), the bipartite graph representation of the hypergraph is needed. To this end, we use a dual representation in which each hyperedge is represented by a new vertex. The new vertex is incident to each of the original vertices in the corresponding hyperedge. The union of the new vertex set and the original vertex set constitute the vertex set of the associated bipartite graph. The new vertices corresponding to hyperedges are on one side and the original hypergraph vertices on the other side. Thus the bipartite graph and star expansion for a hypergraph share the same form, although they are defined for different purposes. For instance, the bipartite graphs associated with the example hypergraphs in Figs. 6.1(a) and 6.1(b) are shown in Figs. 6.1(c) and 6.1(d), respectively (BG stands for bipartite graph).

The Ihara zeta function of the hypergraph $H(V, E_H)$ can be expressed in a rational form as follows:

$$\zeta_H(u) = (1 - u)^{\chi(\text{BG})} \det(\mathbf{I}_{|V(H)|+|E_H(H)|} - \sqrt{u}\mathbf{A}_{\text{BG}} + u\mathbf{Q}_{\text{BG}})^{-1}, \quad (6.30)$$

where $\chi(\text{BG})$ is the Euler number of the associated bipartite graph, \mathbf{A}_{BG} is the adjacency matrix of the associated bipartite graph, and $\mathbf{Q}_{\text{BG}} = \mathbf{D}_{\text{BG}} - \mathbf{I}_{|V(H)|+|E_H(H)|}$. Further details on the arguments leading from (6.29) to (6.30) can be found in [81].

The adjacency matrix of the associated bipartite graph can be formulated using the incidence matrix \mathbf{H} of $H(V, E_H)$:

$$\mathbf{A}_{\text{BG}} = \begin{bmatrix} \mathbf{0}_{|E_H(H)| \times |E_H(H)|} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0}_{|V(H)| \times |V(H)|} \end{bmatrix}. \quad (6.31)$$

The hypergraph Ihara zeta function in the form of (6.30) provides an alternative method for the function value computation, as well as an efficient method of computing the Ihara coefficients, which will be discussed later on in Sect. 6.3.4.

6.3.3 Determinant Expression for Hypergraph Zeta Function

Although the Ihara zeta function can be evaluated efficiently using (6.30), the task of enumerating the coefficients of the polynomial appearing in the denominator of

the Ihara zeta function is difficult, except by resorting to software for symbolic calculation. To efficiently compute these coefficients, a different strategy is adopted. The hypergraph is first transformed into an oriented line graph. The Ihara zeta function is then the reciprocal of the characteristic polynomial for the adjacency matrix of the oriented line graph. Our novel contribution here is to use the existing ideas from hypergraph theory to develop a new hypergraph representation, which can be used in machine learning to distinguishing hypergraphs with the same vertex set but different relational orders.

6.3.3.1 Oriented Line Graph

To establish the oriented line graph associated with the hypergraph $H(V, E_H)$, we commence by constructing a $|e_i|$ -clique, i.e., clique expansion, by connecting each pair of vertices in the hyperedge $e_i \in E_H$ through an edge. The resulting clique expansion graph is denoted by $\text{GH}(V, E_G)$. For $\text{GH}(V, E_G)$, the associated symmetric digraph $\text{DGH}(V, E_d)$ can be obtained by replacing each edge of $\text{GH}(V, E_G)$ by an arc (oriented edge) pair in which the two arcs are inverse to each other. For the example hypergraphs in Figs. 6.1(a) and 6.1(b), their $\text{DGH}(V, E_d)$ are shown in Figs. 6.1(e) and 6.1(f), respectively, where the oriented edges derived from the same hyperedge are colored the same while from different hyperedges are colored differently. Finally, the oriented line graph of the hypergraph can be established based on the symmetric digraph. The vertex set V_{ol} and edge set E_{ol} of the oriented line graph are defined as follows [81]:

$$\begin{aligned} V_{\text{ol}} &= E_d(\text{DGH}), \\ E_{\text{ol}} &= \{(e_d(u, v), e_d(v, w)) \in E_d \times E_d; u, w \notin E_H\}. \end{aligned} \quad (6.32)$$

One observation that needs to be made here is that the adjacency matrix A_H and Laplacian matrix L_H for a hypergraph introduced in Sect. 6.3.1 are actually those of the graph established on the clique expansion, but without an edge-weight attachment. These matrix representations can induce ambiguity when representing relational structures with different relational orders. This point is illustrated by the two example hypergraphs in Figs. 6.1(a) and 6.1(b) which have the same clique graph and thus the same adjacency matrix and Laplacian matrix. The reason for this is that the clique expansion only records adjacency relationships between pairs of nodes and cannot distinguish whether or not two edges in the clique are derived from the same hyperedge. Thus the clique graph representations for hypergraph result in loss of information concerning relational order. However, the Ihara zeta function overcomes this deficiency by avoiding the interaction between two edges derived from the same hyperedge. This is due to the constraint in (6.32) that the connecting oriented edge pair in the same clique of DGH cannot establish an oriented edge in the oriented line graph. According to these properties, the example hypergraphs with the same adjacency matrix and Laplacian matrix in Figs. 6.1(a) and 6.1(b) produce oriented line graphs with totally different structures as shown in

Figs. 6.1(g) and 6.1(h), respectively (OLG stands for oriented line graph), where the constraint in (6.32) prevents connections between any nodes with the same color in Figs. 6.1(g) and 6.1(h). The adjacency matrix \mathbf{T}_H of the oriented line graph is the Perron–Frobenius operator of the original hypergraph. For the (i, j) th entry of \mathbf{T}_H , $\mathbf{T}_H(i, j)$ is 1 if there is one edge directed from the vertex with label i to the vertex with label j in the oriented line graph, otherwise it is 0. Unlike the adjacency matrix of an undirected graph, the Perron–Frobenius operator for a hypergraph is not a symmetric matrix. This is because of the constraint described above that arises in the construction of oriented edges. Specifically, it is the fact that the arc pair with two arcs that are derived from the same hyperedge in the original hypergraph is not allowed to establish an oriented edge in the oriented line graph that causes the asymmetry of \mathbf{T}_H .

6.3.3.2 Characteristic Polynomial

With the oriented line graph to hand, the Ihara zeta function for a hypergraph can be written in the form of a determinant using the Perron–Frobenius operator [81]:

$$\zeta_H(u) = \det(\mathbf{I}_H - u\mathbf{T}_H)^{-1} = (c_0 + c_1u + \dots + c_{M-1}u^{M-1} + c_Mu^M)^{-1}, \quad (6.33)$$

where M is the highest order of the polynomial. The polynomial coefficients c_0, c_2, \dots, c_M are referred to as the Ihara coefficients. From (6.33), we can see that M is the dimensionality of the square matrix \mathbf{T}_H . To establish pattern vectors from the hypergraph Ihara zeta function for the purposes of characterizing hypergraphs in machine learning, it is natural to consider taking function samples as the elements. Although the function values at most of the sampling points will perform well in distinguishing hypergraphs, there is the possibility of sampling at poles giving rise to meaningless infinities. Hence, the pattern vectors consisting of function samples are potentially unstable representations of hypergraphs, since the distribution of poles is unknown beforehand. The characteristic polynomial coefficients, i.e., the Ihara coefficients, do not give rise to infinities. From (6.33), it is clear that each coefficient can be derived from the elementary symmetric polynomials of the eigenvalue set $\{\lambda_1, \lambda_2, \lambda_3, \dots\}$ of \mathbf{T}_H as $c_r = (-1)^r \sum_{k_1 < k_2 < \dots < k_r} \lambda_{k_1} \lambda_{k_2} \dots \lambda_{k_r}$.

Furthermore, the Ihara coefficients relate strongly to the hypergraph-structure since the Ihara zeta function records information about prime cycles in the hypergraphs. We can construct pattern vectors using a dominant subset of the Ihara coefficients $\mathbf{v} = [c_{r1}c_{r2} \dots c_{rN}]^T$ for a hypergraph and then apply them to clustering hypergraphs.

6.3.4 Numerical Computation

The formation of \mathbf{T}_H and its eigen-decomposition tend to be computationally expensive for practical problems, because the matrix \mathbf{T}_H are usually of big size. To

overcome the deficiency of computing the Ihara coefficients using (6.33), we develop a straightforward, yet efficient method which starts from the associated bipartite graph. Instead of constructing the oriented line graph for a hypergraph, we establish the oriented line graph for the bipartite graph. Considering the rational expression (6.30) based on the associated bipartite graph, we have

$$\zeta_H^{-1}(u) = Z_{BG}^{-1}(\sqrt{u}) = \det(\mathbf{I}_{BG} - \sqrt{u}\mathbf{T}_{BG}), \tag{6.34}$$

where \mathbf{T}_{BG} is the Perron–Frobenius operator of the associated bipartite graph, of which the Ihara zeta function (according to its original definition [6]) is represented as

$$Z_{BG}^{-1}(u) = \prod_{p \in P_{BG}} (1 - u^{|p|}) = (1 - u^{|p_1|})(1 - u^{|p_2|})(1 - u^{|p_3|}) \dots, \tag{6.35}$$

where p_i is the i th prime cycle in the set P_{BG} of prime cycle equivalence classes of the bipartite graph. Note that every cycle in a bipartite graph has an even length, i.e., $|p_i|$ is always an even number for a bipartite graph. Let $\{\tilde{c}_0, \tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4, \tilde{c}_5, \tilde{c}_6, \dots\}$ denote the Ihara coefficient set of the bipartite graph. It is clear that $Z_{BG}^{-1}(u)$ is a polynomial with the odd coefficients equal to zero:

$$\begin{aligned} Z_{BG}^{-1}(u) &= \det(\mathbf{I}_{BG} - u\mathbf{T}_{BG}) = \tilde{c}_0 + \tilde{c}_1u + \tilde{c}_2u^2 + \tilde{c}_3u^3 + \tilde{c}_4u^4 \\ &\quad + \tilde{c}_5u^5 + \tilde{c}_6u^6 + \dots \\ &= \tilde{c}_0 + \tilde{c}_2u^2 + \tilde{c}_4u^4 + \tilde{c}_6u^6 + \dots \end{aligned} \tag{6.36}$$

Taking \sqrt{u} as the argument of the bipartite graph Ihara zeta function instead of u ,

$$\begin{aligned} \zeta_H^{-1}(u) &= Z_{BG}^{-1}(\sqrt{u}) = \det(\mathbf{I}_{BG} - \sqrt{u}\mathbf{T}_{BG}) \\ &= (1 - (\sqrt{u})^{|p_1|})(1 - (\sqrt{u})^{|p_2|}) \dots \\ &= \tilde{c}_0 + 0\sqrt{u} + \tilde{c}_2(\sqrt{u})^2 + 0(\sqrt{u})^3 + \tilde{c}_4(\sqrt{u})^4 \\ &\quad + 0(\sqrt{u})^5 + \tilde{c}_6(\sqrt{u})^6 + \dots \\ &= \tilde{c}_0 + \tilde{c}_2u + \tilde{c}_4u^2 + \tilde{c}_6u^3 + \dots \\ &= c_0 + c_1u + c_2u^2 + c_3u^3 + \dots \end{aligned} \tag{6.37}$$

As we can see in (6.37), the Ihara coefficients of a hypergraph can be efficiently obtained by selecting just the even-indexed Ihara coefficients of the associated bipartite graph. This is much more efficient than the computation based on the oriented line graph of the hypergraph, because \mathbf{T}_{BG} is much smaller in size than \mathbf{T}_H , especially for large hypergraphs. The size of the Perron–Frobenius operator of an irregular hypergraph tends to be difficult to enumerate. Here we thus use the K -regular hypergraph, i.e., hypergraph with every hyperedge containing K vertices, for analyzing the computational complexity of the Perron–Frobenius operators \mathbf{T}_H and \mathbf{T}_{BG} . Suppose there are in total N hyperedges in the K -regular hypergraph. To obtain \mathbf{T}_H , the

clique expansion and its digraph of the K -regular hypergraph need to be established according to the transform introduced in Sect. 6.3.3.1. This procedure produces an oriented line graph with $K(K-1)N$ vertices and a Perron–Frobenius operator of size $(K-1)KN \times (K-1)KN$. To obtain T_{BG} , the bipartite graph and its digraph of the K -regular hypergraph need to be established. This procedure produces an oriented line graph with $2KN$ vertices and a Perron–Frobenius operator of size $2KN \times 2KN$. For regular hypergraphs K is greater than 2, and the relation always holds for $2KN < (K-1)KN$. As a result, the size of T_{BG} is smaller than that of T_H when $K > 3$. The computational complexity of obtaining the Ihara coefficients is governed by the eigen-decomposition of the Perron–Frobenius operator. This requires $O(n^3)$ operations where n is the size of the Perron–Frobenius operator. Therefore, the computational overheads of eigen-decomposition on T_{BG} are lower than those of T_H . We refer to [65] for an efficient way of computing the Ihara coefficients given the eigenvalues of T_{BG} .

6.3.5 Experimental Evaluation

To establish hypergraphs on the visual objects, we first extract feature points using the Harris detector [34] as the vertices of hypergraphs. Let $\mathbf{c}(v_i)$ denote the spatial coordinate of the feature point v_i in an image, and $I(v_i)$ denote the intensity of v_i . For each image, we construct the hypergraph using the method introduced in [56], where the element $H(i, j)$ of incidence matrix is 1 if $\|\mathbf{c}(v_i) - \mathbf{c}(v_j)\| \leq \text{Th}_{j1}$ and $|I(v_i) - I(v_j)| \leq \text{Th}_{j2}$, and 0 otherwise. Here Th_{j1} is the neighborhood threshold set to 1/4 the size of the image and Th_{j2} is the similarity threshold determined by the standard deviation of the intensities of neighboring feature points.

We first test the Ihara coefficient pattern vector in the form of $v_H = [c_3, c_4, \ln(|c_{M-3}|), \ln(|c_{M-2}|), \ln(|c_{M-1}|), \ln(|c_M|)]^T$ in characterizing within-class hypergraphs. We establish hypergraphs on ten images of a model house in the Chalet data set [56]. The images are taken consecutively as the camera pans around the model house in regular angular increments. Figure 6.2 shows the PCA projections of the hypergraphs based on the truncated Laplacian spectrum, i.e., the leading six nonzero Laplacian eigenvalues, and the Ihara coefficients. The Laplacian spectra produce an erratic trajectory. The Ihara coefficients produce a much smoother trajectory and the neighboring images in the sequence are generally Euclidean neighbors in the eigenspace.

Figure 6.3 compares the performance of the largest Laplacian eigenvalue and the final Ihara coefficient for hypergraphs extracted from four objects in the COIL dataset [56]. The Ihara coefficients give clearer class separability than the Laplacian eigenvalues.

Finally, we test the Ihara coefficients for clustering both unweighted graphs and unweighted hypergraphs. The graphs and hypergraphs are extracted from the images in the COIL dataset. We establish a Delaunay graph on the feature points of each image, and construct the pattern vectors in the form of $v_{Gs} = [c_3, c_4, \ln(|c_{2M}|)]^T$ for

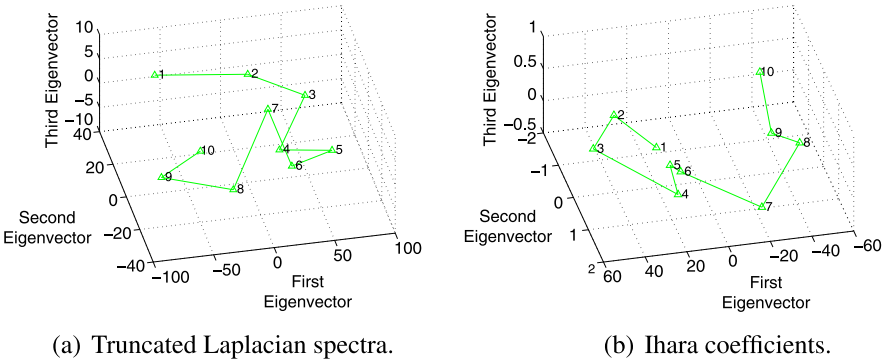


Fig. 6.2 Within-class trajectory

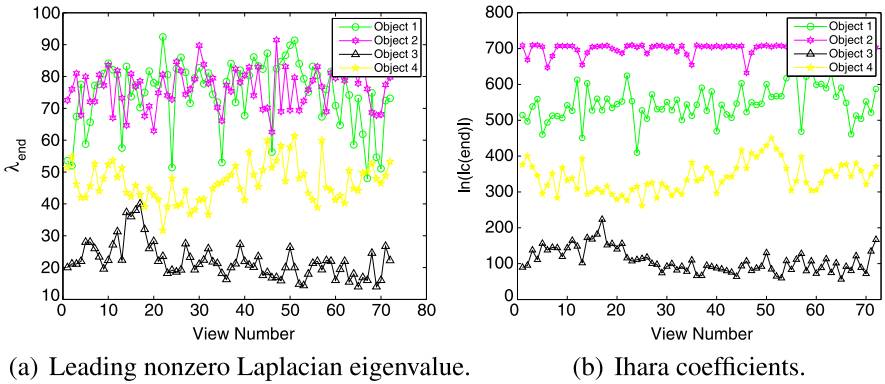


Fig. 6.3 Ihara coefficient plot

graphs. We evaluate the clustering performance obtained with different numbers of object classes. After performing PCA on the pattern vectors both for graphs and hypergraphs, we locate the clusters using the K -means method and calculate the Rand index, which is plotted as a function of class number in Fig. 6.4. We use Laplacian spectra for graphs and hypergraphs for comparison. From this set of experiments, it is clarified that for both graphs and hypergraphs, the Ihara coefficients outperform the Laplacian spectra.

6.3.6 Section Summary

We have pointed out the deficiency of the vertex-to-vertex matrix representations for learning hypergraph-structure and applied the Ihara coefficients to hypergraph characterization to overcome these problems. The Ihara coefficients are a flexible tool which can be computed in a consistent manner for both graphs and hypergraphs.

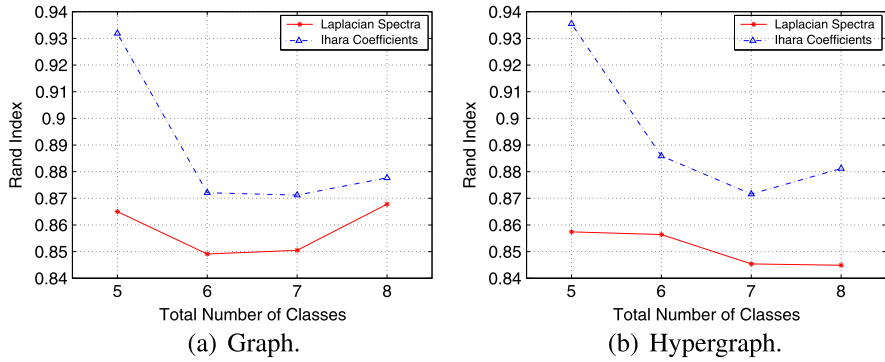


Fig. 6.4 Rand index

They can effectively overcome the ambiguity in distinguishing high order relational structures when matrix representations fail to work. Furthermore, we have proposed an efficient method for computing the Ihara coefficient set. Experimental results show that the Ihara coefficients are superior to spectral methods, both for graphs and hypergraphs.

6.4 Chapter Summary

In this chapter, we have illustrated two contrasting approaches to the problem of embedding non-Euclidean data. The first was based on the idea of spherical embedding, where data is embedded onto the surface of sphere of optimal radius. The second is a method designed to preserve elements of the structure of hypergraphs.

Turning first on the spherical embedding, our results on synthetic and real data show that the elliptic embedding performs well under noisy conditions and can deliver low-distortion embeddings for a wide variety of datasets. Hyperbolic data seems to be much less common (at least in our datasets) and is more difficult to accurately embed. Nevertheless, in low-noise cases and for some datasets, the hyperbolic space can also be used to accurately embed non-Euclidean dissimilarity data. While accurate embedding is our goal here, it is natural to want to apply pattern recognition techniques to the embedded data. Unfortunately, many methods rely, either explicitly or implicitly, on an underlying kernel space which is Euclidean. We believe that much more work needs to be done on applying such techniques in non-flat spaces.

In the case of the Ihara coefficients, we have performed a characteristic polynomial analysis on hypergraphs and characterized (irregular) unweighted hypergraphs based on the Ihara zeta function. We have used the Ihara coefficients as the elements of pattern vectors for a hypergraph. Experimental results show the effectiveness of the proposed method. Further research will focus on investigating the possibility of using Ihara zeta function for the characterization of weighted hypergraphs.

K -regular hypergraphs with weighted hyperedges have recently been found to be a powerful tool in representing data with high-order relations. In the light of its potential in revealing high-order structure, we will investigate developing methods for improving the accuracy of clustering and matching data with high-order affinities by involving the Ihara zeta function.

References

1. Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., Belongie, S.: Beyond pairwise clustering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 838–845 (2005)
2. Agarwal, S., Branson, K., Belongie, S.: Higher-order learning with graphs. In: Proceedings of the International Conference on Machine Learning, pp. 17–24 (2006)
3. Bai, X., Hancock, E.R., Wilson, R.C.: Graph characteristics from the heat kernel trace. *Pattern Recognit.* **42**(11), 2589–2606 (2009)
4. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Am. Math. Soc.* **73**, 360–363 (1967)
5. Bartholdi, L.: Counting paths in graphs. *Enseign. Math.* **45**, 83–131 (1999)
6. Bass, H.: The Ihara–Selberg zeta function of a tree lattice. *Int. J. Math.* **6**, 717–797 (1992)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**(6), 1373–1396 (2003)
8. Behmo, R., Paragios, N., Prinet, V.: Graph commute times for image representation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
9. Bolla, Spectra, M.: Euclidean representations and clusterings of hypergraphs. *Discrete Math.* **117** (1993)
10. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview. *Pattern Recognit.* **35**(3), 651–658 (2001)
11. Brook, B.P.: The coefficients of the characteristic polynomial in terms of the eigenvalues and the elements of an $n \times n$ matrix. *Appl. Math. Lett.* **19**(6), 511–515 (2006)
12. Broom, M., Cannings, C., Vickers, G.T.: Multi-player matrix games. *Bull. Math. Biol.* **59**(5), 931–952 (1997)
13. Bunke, H., Dickinson, P., Neuhaus, M., Stettler, M.: Matching of hypergraphs—algorithms, applications, and experiments. *Stud. Comput. Intell.* **91**, 131–154 (2008)
14. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. *Pattern Recognit. Lett.* **19**(3), 255–259 (1998)
15. Cameron, P.J.: Strongly regular graphs. In: Topics in Algebraic Graph Theory, pp. 203–221. Cambridge University Press, Cambridge (2004)
16. Chen, G., Lerman, G.: Spectral curvature clustering (SCC). *Int. J. Comput. Vis.* **81**(3), 317–330 (2009)
17. Chen, G., Lerman, G.: Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Found. Comput. Math.* **9**, 517–558 (2009)
18. Chen, G., Atev, S., Lerman, G.: Kernel spectral curvature clustering (KSCC). In: Proceedings of International Workshop on Dynamical Vision, pp. 765–772 (2009)
19. Chertok, M., Keller, Y.: Efficient high order matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2205–2215 (2010)
20. Chung, F.: Spectral Graph Theory. Am. Math. Soc., Providence (1992)
21. Chung, F.: The Laplacian of a hypergraph. In: AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 10, pp. 21–36 (1993)
22. Cvetković, D., Rowlinson, P., Simić, S.K.: Eigenvalue bounds for the signless Laplacian. *Publ. Inst. Math. (Belgr.)* **81**(95), 11–27 (2007)

23. Daitch, S.I., Kelner, J.A., Spielman, D.A.: Fitting a graph to vector data. In: Proceedings of International Conference on Machine Learning, pp. 201–208 (2009)
24. Duchenne, O., Bach, F.R., Kweon, I.S., Ponce, J.: A tensor-based algorithm for high-order graph matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1980–1987 (2009)
25. Emms, D.: Analysis of graph structure using quantum walks. Ph.D. Thesis, University of York (2008)
26. Emms, D., Hancock, E.R., Severini, S., Wilson, R.C.: A matrix representation of graphs and its spectrum as a graph invariant. *Electron. J. Comb.* **13**(R34) (2006)
27. Emms, D., Severini, S., Wilson, R.C., Hancock, E.R.: Coined quantum walks lift the cospectrality of graphs and trees. *Pattern Recognit.* **42**(9), 1988–2002 (2009)
28. Ferrer, M., Valveny, E., Serratos, F., Riesen, K., Bunke, H.: Generalized median graph computation by means of graph embedding in vector spaces. *Pattern Recognit.* **43**(4), 1642–1655 (2010)
29. Fischer, B., Buhmann, J.M.: Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(4), 513–518 (2003)
30. Gibson, D., Kleinberg, J., Raghavan, P.: Clustering categorical data: an approach based on dynamical systems. *VLDB J.* **8**(4–3), 222–236 (2000)
31. Govindu, V.M.: A tensor decomposition for geometric grouping and segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1150–1157 (2005)
32. Grover, L.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on the Theory of Computation, pp. 212–219 (1996)
33. Hagen, L., Kahng, A.B.: New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **11**(9), 1074–1085 (1992)
34. Harris, C.G., Stephens, M.J.: A combined corner and edge detector. In: Proceedings of Fourth Alvey Vision Conference, pp. 147–151 (1994)
35. Hashimoto, K.: Artin-type L-functions and the density theorem for prime cycles on finite graphs. *Adv. Stud. Pure Math.* **15**, 211–280 (1989)
36. He, X., Cai, D., Niyogi, P.: Tensor subspace analysis. In: Proceedings of Advances in Neural Information Processing Systems, pp. 507–514 (2005)
37. He, X., Cai, D., Liu, H., Han, J.: Image clustering with tensor representation. In: Proceedings of ACM Multimedia, pp. 132–140 (2005)
38. He, Z., Cichocki, A., Xie, S., Choi, K.: Detecting the number of clusters in n -way probabilistic clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(11), 2006–2021 (2010)
39. Ihara, Y.: Discrete subgroups of $PL(2, k_\varphi)$. In: Proceedings of Symposium on Pure Mathematics, pp. 272–278 (1965)
40. Ihara, Y.: On discrete subgroups of the two by two projective linear group over p -adic fields. *J. Math. Soc. Jpn.* **18**, 219–235 (1966)
41. Jebara, T., Wang, J., Chang, S.F.: Graph construction and b -matching for semi-supervised learning. In: Proceedings of International Conference on Machine Learning, pp. 441–448 (2009)
42. Kondor, R., Borgwardt, K.M.: The skew spectrum of graphs. In: Proceedings of International Conference on Machine Learning, pp. 496–503 (2008)
43. Kondor, R., Shervashidze, N., Borgwardt, K.M.: The graphlet spectrum. In: Proceedings of International Conference on Machine Learning, pp. 529–536 (2009)
44. Kotani, M., Sunada, T.: Zeta functions of finite graphs. *J. Math. Sci. Univ. Tokyo* **7**(1), 7–25 (2000)
45. Lenman, G., Whitehouse, J.T.: On d -dimensional d -semimetrics and simplex-type inequalities for high-dimensional sine functions. *J. Approx. Theory* **156**(1), 52–81 (2009)
46. Li, W., Sole, P.: Spectra of regular graphs and hypergraphs and orthogonal polynomials. *Eur. J. Comb.* **17**, 461–477 (1996)
47. Liu, X., Yan, S., Jin, H.: Projective nonnegative graph embedding. *IEEE Trans. Image Process.* **19**(5), 1126–1137 (2010)

48. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. *Pattern Recognit.* **36**(10), 2213–2223 (2003)
49. Mantrach, A., Yen, L., Callut, J., Francoise, K., Shimbo, M., Sauerens, M.: The sum-over-paths covariance kernel: a novel covariance measure between nodes of a directed graph. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(6), 1112–1126 (2010)
50. Maier, M., von Luxburg, U., Hein, M.: Influence of graph construction on graph-based clustering measures. In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 1025–1032 (2008)
51. Mizuno, H., Sato, I.: Bartholdi zeta function of graph coverings. *J. Comb. Theory, Ser. B* **89**(1), 27–41 (2003)
52. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96 (1996)
53. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1), 167–172 (2007)
54. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(11), 1873–1890 (2007)
55. Ramon, J., Gartner, T.: Expressivity versus efficiency of graph kernels. In: *Proceedings of First International Workshop on Mining Graphs, Trees and Sequences*, pp. 65–74 (2003)
56. Ren, P., Wilson, R.C., Hancock, E.R.: Spectral embedding of feature hypergraphs. In: *Proceedings of Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 308–317 (2008)
57. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. *Image Vis. Comput.* **27**(7), 950–959 (2009)
58. Riesen, K., Bunke, H.: Graph classification by means of Lipschitz embedding. *IEEE Trans. Syst. Man Cybern., Part B, Cybern.* **39**(6), 1472–1483 (2009)
59. Robles-Kelly, A., Hancock, E.R.: A probabilistic spectral framework for grouping and segmentation. *Pattern Recognit.* **37**(7), 1387–1405 (2004)
60. Rodriguez, J.A.: On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear Multilinear Algebra* **51**, 285–297 (2003)
61. Rota Bulò, S., Albarelli, A., Pelillo, M., Torsello, A.: A hypergraph-based approach to affine parameters estimation. In: *Proceedings of the International Conference on Pattern Recognition*, pp. 1–4 (2008)
62. Rota Bulò, S., Torsello, A., Pelillo, M.: A game-theoretic approach to partial clique enumeration. *Image Vis. Comput.* **27**(7), 911–922 (2009)
63. Rota Bulò, S., Pelillo, M.: A game-theoretic approach to hypergraph clustering. In: *Proceedings of Neural Information Processing Conference*, vol. 22, pp. 1571–1579 (2009)
64. Rota Bulò, S., Pelillo, M.: A game-theoretic approach to hypergraph clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1312–1327 (2013)
65. Rota Bulò, S., Hancock, E.R., Aziz, F., Pelillo, M.: Efficient computation of Ihara coefficients using the Bell polynomial recursion. *Linear Algebra Appl.* **436**(5), 1436–1441 (2012)
66. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
67. Sanfeliu, A., Fu, K.S.: A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. Syst. Man Cybern.* **13**(3), 353–362 (1983)
68. Sato, I.: A new Bartholdi zeta function of a graph. *Int. J. Algebra Comput.* **1**(6), 269–281 (2007)
69. Savchenko, S.V.: The zeta function and Gibbs measures. *Russ. Math. Surv.* **48**(1), 189–190 (1993)
70. Scott, G., Storm, C.K.: The coefficients of the Ihara zeta function. *Involve—J. Math.* **1**(2), 217–233 (2008)
71. Sengupta, K., Boyer, K.L.: Organizing large structural modelbases. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**(4), 321–332 (1995)
72. Shankar, R.: *Principles of Quantum Mechanics*, 2nd edn. Plenum, New York (1994)

73. Shashua, A., Levin, A.: Linear image coding for regression and classification using the tensor-rank principle. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 623–630 (2001)
74. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symmetric non-negative tensor factorization. In: Proceedings of the European Conference on Computer Vision, pp. 595–608 (2006)
75. Shaw, B., Jebara, T.: Structure preserving embedding. In: Proceedings of International Conference on Machine Learning, pp. 937–944 (2009)
76. Shervashidze, N., Borgwardt, K.M.: Fast subtree kernels on graphs. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1660–1668 (2009)
77. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
78. Stark, H.M., Terras, A.A.: Zeta functions of finite graphs and coverings. *Adv. Math.* **121**, 124–165 (1996)
79. Stark, H.M., Terras, A.A.: Zeta functions of finite graphs and coverings, II. *Adv. Math.* **154**, 132–195 (2000)
80. Stark, H.M., Terras, A.A.: Zeta functions of finite graphs and coverings, III. *Adv. Math.* **208**(2), 467–489 (2007)
81. Storm, C.K.: The zeta function of a hypergraph. *Electron. J. Comb.* **13** (2006)
82. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
83. Torsello, A., Robles-Kelly, A., Hancock, E.R.: Discovering shape classes using tree edit distance and pairwise clustering. *Int. J. Comput. Vis.* **72**(3), 259–285 (2007)
84. Torsello, A., Hancock, E.R.: Learning Shape-Classs Using a Mixture of Tree-Unions. *IEEE Trans. Pattern Anal. Mach. Intell.* 954–967 (2006)
85. Tsai, W.H., Fu, K.S.: Subgraph error-correcting isomorphism for syntactic pattern recognition. *IEEE Trans. Syst. Man Cybern.* **13**(1), 48–62 (1983)
86. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: tensorFaces. In: Proceedings of the European Conference on Computer Vision, pp. 447–460 (2002)
87. Vishwanathan, S.V.N., Borgwardt, K.M., Kondor, I.R., Schraudolph, N.N.: Graph kernels. *J. Mach. Learn. Res.* **11**, 1201–1242 (2010)
88. Wang, C., Song, Z., Yan, S., Zhang, L., Zhang, H.J.: Multiplicative nonnegative graph embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 389–396 (2009)
89. Watanabe, Y., Fukumizu, K.: Graph zeta function in the Bethe free energy and loopy belief propagation. In: Proceedings Neural Information Processing Systems, pp. 2017–2025 (2009)
90. Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: Proceedings of International Conference on Computer Vision, pp. 975–982 (1999)
91. Wilson, R.C., Hancock, E.R., Luo, B.: Pattern vectors from algebraic graph theory. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(7), 1112–1124 (2005)
92. Wilson, R.C., Zhu, P.: A study of graph spectra for comparing graphs and trees. *Pattern Recognit.* **41**(9), 2833–2841 (2008)
93. Wong, A.K.C., Lu, S.W., Rioux, M.: Recognition and shape synthesis of 3D objects based on attributed hypergraphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(3), 279–290 (1989)
94. Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S.: Graph embedding and extension: a general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1), 40–51 (2007)
95. Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., Zhang, H.J.: Discriminant analysis with tensor representation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 526–532 (2005)
96. Yang, J., Yan, S., Fu, Y., Li, X., Huang, T.S.: Non-negative graph embedding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
97. Zaslavskiy, M., Bach, F., Vert, J.-P.: A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(12), 2227–2242 (2009)

98. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: Proceedings of International Conference on Computer Vision, pp. 294–301 (2005)
99. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
100. Zhao, D., Tang, X.: Cyclizing clusters via zeta function of a graph. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1953–1960 (2008)
101. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: clustering, classification, and embedding. In: Proceedings of Advances in Neural Information Processing Systems, pp. 1601–1608 (2007)
102. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 2319–2323 (2000)
103. Pekalska, E., Harol, A., Duin, R.P.W., Spillmann, B., Bunke, H.: Non-Euclidean or non-metric measures can be informative. In: Proceedings of SSPR/SPR, pp. 871–880 (2006)
104. Lindman, H., Caelli, T.: Constant curvature Riemannian scaling. *J. Math. Psychol.* 89–109 (1978)
105. Cox, T.F., Cox, M.A.A.: In: Multidimensional Scaling on a Sphere, pp. 2943–2953 (1991)
106. Shavitt, Y., Tankel, T.: Hyperbolic embedding of Internet graph for distance estimation and overlay construction. In: *IEEE/ACM Transactions on Networking*, pp. 25–36 (2008)
107. Hubert, L., Arabie, P., Meulman, J.: Linear and circular unidimensional scaling for symmetric proximity matrices. *Br. J. Math. Stat. Psychol.* 253–284 (1997)
108. Robles-Kelly, A., Hancock, E.R.: A Riemannian approach to graph embedding. *Pattern Recognit.* 1042–1056 (2007)
109. Pekalska, E., Duin, R.P.W.: *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, Singapore (2005)
110. Lee, W.J., Duin, R.P.W.: An inexact graph comparison approach in joint eigenspace. In: Proceedings of SS+SPR2008 (2008)
111. Scannell, J., Blakemore, C., Young, M.: Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.* 1463–1483 (1995)
112. Lichtenauer, J., Hendriks, E.A., Reinders, M.J.T.: Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal. Mach. Intell.* 2040–2046 (2008)
113. Roth, V., Laub, J., Buhmann, J.M., Mueller, K.-R.: Going metric: denoising pairwise data. In: *Advances in Neural Information Processing Systems*, pp. 841–856 (2003)
114. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 286–299 (2007)
115. Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Anal. Mach. Intell.* 1386–1391 (1997)
116. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Trans. Med. Imaging* 995–1005 (2004)
117. Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: clustering, classification, and embedding. In: *NIPS* (2007)
118. Gold, S., Rangarajan, A.: A Graduated Assignment Algorithm for Graph Matching. *IEEE Trans. Pattern Anal. Mach. Intell.* 377–388 (1996)
119. Nene, S.A., Nayar, S.K., Murase, H.: *Columbia Object Image Library (COIL-100)*, Technical Report CUCS-006-96 (1996)
120. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. In: *Pattern Recognition*, pp. 2213–2230 (2003)
121. Pennec, X., Fillard, P., Ayache, N.: A Riemannian framework for tensor computing. In: *International Journal of Computer Vision*, pp. 41–66 (2006)
122. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview. In: *Pattern Recognition*, pp. 651–658 (2002)
123. Ren, P., Aleksić, T., Wilson, R.C., Hypergraphs, E.R.H.: Characteristic polynomials and the Ihara zeta function. In: *Proceedings of CAIP* (2009)

124. Friedman, N., Koller, D.: Being Bayesian about Bayesian Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *Mach. Learn.* 95–125 (2003)
125. Torgerson, W.S.: *Theory and Methods of Scaling*. Wiley, New York (1958)
126. Laub, J., Roth, V., Buhmann, J.M., Müller, K.R.: On the information and representation of non-Euclidean pairwise data. *Pattern Recognit.* 1815–1826 (2006)
127. Kondor, R., Lafferty, J.: Diffusion kernels on graphs and other discrete input spaces. In: *Proceedings of ICML* (2002)
128. Wu, G., Chang, E.Y., Zhang, Z.: Learning with non-metric proximity matrices. In: *ACM International Conference on Multimedia*, pp. 411–414 (2005)
129. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.M.: Optimal cluster preserving embedding of non-metric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.* 1540–1551 (2003)
130. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New York (1999)
131. Krauthgamer, R., Linial, N., Magen, A.: Metric embeddings: Beyond one-dimensional distortion. *Discrete Comput. Geom.* 339–356 (2004)
132. Hein, M., Audibert, J.Y., Luxburg, U.V.: From graphs to manifolds—weak and strong point-wise consistency of graph Laplacians. In: *Annual Conference on Learning Theory* (2005)

Chapter 7

Structure Preserving Embedding of Dissimilarity Data

Volker Roth, Thomas J. Fuchs, Julia E. Vogt, Sandhya Prabhakaran,
and Joachim M. Buhmann

Abstract Partitioning methods for observations represented by pairwise dissimilarities are studied. Particular emphasis is put on their properties when applied to dissimilarity matrices that do not admit a loss-free embedding into a vector space. Specifically, the *Pairwise Clustering* cost function is shown to exhibit a *shift invariance* property which basically means that any symmetric dissimilarity matrix can be modified to allow a vector-space representation without distorting the optimal group structure. In an approximate sense, the same holds true for a probabilistic generalization of Pairwise Clustering, the so-called *Wishart–Dirichlet Cluster Process*. This shift-invariance property essentially means that these clustering methods are “blind” against Euclidean or metric violations. From the application side, such blindness against metric violations might be seen as a highly desired feature, since it broadens the applicability of certain algorithms. From the viewpoint of theory building, however, the same property might be viewed as a “negative” result, since studying these algorithms will not lead to any new insights on the role of metricity in clustering problems.

V. Roth (✉) · J.E. Vogt · S. Prabhakaran
Computer Science Department, University of Basel, Basel, Switzerland
e-mail: volker.roth@unibas.ch

J.E. Vogt
e-mail: julia.vogt@unibas.ch

S. Prabhakaran
e-mail: sandhya.prabhakaran@unibas.ch

T.J. Fuchs
California Institute of Technology, Pasadena, CA 91125, USA
e-mail: fuchs@caltech.edu

J.M. Buhmann
Swiss Federal Institute of Technology Zurich, Zurich, Switzerland
e-mail: jbuhmann@inf.ethz.ch

7.1 Introduction

For several major applications in data mining, data is often not available as feature vectors in a vector space. For instance, genomics typically produce data represented as strings from some alphabet, psychology yields sets of similarity judgments, yet other fields like social sciences measure so-called preference data. The missing vector space representation precludes the use of well established machine learning techniques such as Principal Component Analysis [1] or Support Vector Machines [2].

A common approach to handling non-vectorial datasets is to replace the initial data by a collection of real numbers representing some “comparison” among the elements of the dataset (see Chaps. 2 and 6). This procedure yields a matrix gathering the pairwise relations between the original objects, which may be the starting point of further data analysis.

The clustering approaches discussed in this chapter aim at identifying subsets or clusters of objects represented as “blocks” in a permuted dissimilarity matrix. The underlying idea is that objects grouped together in such a cluster can be reasonably well described as a homogeneous sub-population. Our focus on dissimilarity matrices implies that we do not have access to a vectorial representation of the objects, and in general, no such representation will exist, since we do not assume that the dissimilarity matrix fulfills the axioms of a valid metric.

In this chapter, we summarize our studies on embedding strategies in the context of clustering. In the first part, we will mainly summarize our results for the pairwise k -means clustering cost function as outlined in [3]: we begin with a short overview of proximity-based data grouping, and then we focus on reformulating such problems with vectorial data representations. For the class of pairwise clustering methods that are related to minimizing a shift-invariant cost function, the *constant shift embedding* procedure is presented. A surprising property of this embedding is the complete preservation of group structure. The original non-metric pairwise clustering problem can be restated as a grouping problem over points in a vector space, yielding identical assignments of objects to clusters. Using the constant-shift embedding principle, we then demonstrate the equivalence between the *pairwise clustering* cost function and the classical k -means grouping criterion in the embedding space. The conclusion is that the k -means cost function (or its dissimilarity-based counterpart) is essentially “blind” against metric violations.

In the second part, we will analyze a more general setting where the hard-clustering scenario with fixed number of clusters is replaced by a probabilistic approach which is capable of selecting the number of clusters in a data-adaptive way. We show that this probabilistic model is shift invariant only in an approximate sense, and in particular we show that exact shift invariance and data-adaptive selection of the number of clusters define two conflicting goals.

We conclude this chapter with a (sober) discussion about the role of structure preserving embeddings for the overall goal in the SIMBAD project, namely for building a novel theory for similarity-based pattern recognition.

7.2 Constant Shift Embedding for Pairwise Clustering

7.2.1 Proximity-Based Clustering

Unsupervised grouping or *clustering* aims at extracting hidden structure from data [4]. The term data refers to both a set of objects and a set of corresponding object representations resulting from some physical measurement process. Different types of object representations are possible, the two most common of which are *vectorial data* and *pairwise proximity data*. In the first case, a set of n objects is represented as n points in a d -dimensional vector space, whereas in the second case we are given an $n \times n$ pairwise proximity matrix.

The problem of grouping vectorial data has been widely studied in the literature, and many clustering algorithms have been proposed [4, 5]. One of the most popular methods is k -means clustering. It derives a set of k prototype vectors which quantize the data set with minimal quantization error.

Partitioning proximity data is considered a much harder problem, since the inherent structure is hidden in n^2 pairwise relations. This datatype, however, is abundant in many applications, such as molecular biology, psychology, linguistics, etc. In general, the proximities can violate the requirements of a distance measure, i.e., they may be non-symmetric and negative, and the triangle inequality does not necessarily hold. Thus, a loss-free embedding into a vector space is not possible, so that grouping problems of this kind cannot directly be transformed into vectorial problems by means of classical embedding strategies.

Among several methods for clustering proximity-based data, in this first part of this chapter we will focus on those techniques that explicitly minimize a certain cost function. This subset of clustering methods includes, e.g., graph-theoretic approaches like several variations of *Cut* criteria [6], and several methods derived from an axiomatization of pairwise cost functions in [7]. From a theoretical viewpoint, cost-based clustering methods are interesting insofar, as many properties of the grouping solutions can be derived by analyzing invariance properties of the cost function.

Among the class of cost-based criteria, the main focus of this work concerns those cost functions which are invariant under constant additive shifts of the pairwise dissimilarities. For this subset of clustering criteria, we show that there always exists a set of vectorial data representations such that the grouping problem can be equivalently restated in terms of Euclidean distances between these vectors. A special cost function of this kind is the *pairwise clustering cost function*. It is of particular interest, since it combines the properties of additivity, scale- and shift-invariance, and statistical robustness, see [7]. In [8], this grouping problem is stated as a combinatorial optimization problem, which is optimized in a *deterministic annealing* framework after applying a mean-field approximation.

According to Theorem 7.2, we can always find a vectorial data representation such that the optimal partitioning w.r.t. the pairwise cost function is *identical* to k -means partitioning in the embedding space. This property demonstrates that the

embedding method is optimal w.r.t. to distortions of the *data partition*. This distortion preserving embedding has to be contrasted with alternative, in our view not consistent, approaches that are optimal w.r.t. some *a priori* chosen MDS distortion measure.

Formulating pairwise clustering as a k -means problem yields several advantages, both of theoretical and technical nature: (i) the availability of prototype vectors defines a generic rule for using the learned partitioning in a predictive sense, (ii) we can apply standard noise- and dimensionality-reduction methods in order to separate the “signal” part of the data from underlying “noise”, (iii) fast and efficient local search heuristics for optimizing the clustering cost functional often work much better in low dimensional embedding spaces.

7.2.2 The Pairwise Clustering Cost Function

The modeling idea behind the Pairwise Clustering cost function is to minimize the sum of *pairwise* intra-cluster distances, emphasizing *compact* clusters. Optimizing a compactness criterion is certainly a very intuitive meta-principle for exploratory data analysis. It should be noticed, however, that other such meta-principles have been proposed, such as *separation* measures, mixed *compactness/separation* measures or *connectivity* measures. In order to formalize Pairwise Clustering, we define for each object a binary assignment variable that indicates its cluster membership. Let these variables be summarized in the $n \times k$ binary stochastic assignment matrix $M \in \{0, 1\}^{n \times k} : \sum_{v=1}^k M_{iv} = 1$. Given an $n \times n$ dissimilarity matrix D , the Pairwise Clustering cost function reads:

$$H^{\text{pc}} = \frac{1}{2} \sum_{v=1}^k \frac{\sum_{i=1}^n \sum_{j=1}^n M_{iv} M_{jv} D_{ij}}{\sum_{l=1}^n M_{lv}}. \quad (7.1)$$

The optimal assignments \hat{M} are obtained by minimizing H^{pc} . The minimization itself is an \mathcal{NP} hard problem [9], and some approximation heuristics have been proposed: in [8], a *mean field annealing* framework has been presented. In [7], it has been shown that the time-honored *Ward's method* can be viewed as a hierarchical approximation of H^{pc} .

7.2.3 A Special Case: k -Means Clustering

For the special case of squared Euclidean distances between vectors $\{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^d$, it is well known that H^{pc} is identical to the classical k -means cost function, see [4]. We now briefly review this relationship. The k -means cost function is

defined as

$$H^{\text{km}} = \sum_{v=1}^k \sum_{i=1}^n M_{iv} \|x_i - y_v\|^2. \quad (7.2)$$

It measures the sum of squared intra-cluster distances to the prototype vectors

$$y_v := \frac{\sum_{i=1}^n M_{iv} x_i}{n_v}, \quad (7.3)$$

where $n_v := \sum_{i=1}^n M_{iv}$ denotes the number of objects in cluster v . H^{km} can be written in a pairwise fashion by exploiting a simple algebraic identity for squared Euclidean distances:

$$\begin{aligned} \|x_i - y_v\|^2 &= \frac{1}{n_v} \sum_{j=1}^n M_{jv} \|x_i - x_j\|^2 - \frac{1}{2n_v^2} \sum_{j=1}^n \sum_{l=1}^n M_{jv} M_{lv} \|x_j - x_l\|^2, \\ \sum_{i=1}^n M_{iv} \|x_i - y_v\|^2 &= \frac{1}{2n_v} \sum_{j=1}^n \sum_{l=1}^n M_{jv} M_{lv} \|x_j - x_l\|^2. \end{aligned} \quad (7.4)$$

Substituting the latter identity into (7.2), we obtain

$$H^{\text{km}} = \frac{1}{2} \sum_{v=1}^k \frac{\sum_{i=1}^n \sum_{j=1}^n M_{iv} M_{jv} \|x_i - x_j\|^2}{\sum_{l=1}^n M_{lv}} = H^{\text{pc}}. \quad (7.5)$$

From this viewpoint, k -means clustering can be interpreted as a method for minimizing the sum of squared *pairwise* intra-cluster distances $D_{ij} = \|x_i - x_j\|^2$. The reader should notice, however, that in the general case of arbitrary dissimilarities D_{ij} a direct algebraic re-transformation of H^{pc} into H^{km} is *not* possible. Despite this fact, we will show in the remainder of this paper that it is still possible to obtain the optimal assignment variables \hat{M} with respect to $H^{\text{pc}}(M)$ by minimizing a suitably transformed k -means problem. The key ingredient will be the *shift invariance property* of the Pairwise Clustering cost function: H^{pc} is invariant (up to a constant) under additive shifts of the *off-diagonal* elements of the dissimilarity matrix:

$$\tilde{D}_{ij} = D_{ij} + d_0(1 - \delta_{ij}) \quad \Rightarrow \quad \tilde{H} = H + (1/2) \cdot (n - k)d_0 = H + \text{const}. \quad (7.6)$$

Note that the optimal assignments of objects to clusters are not influenced by adding a constant to the cost function, i.e., $\hat{M}(\tilde{D}) = \hat{M}(D)$.

7.2.4 Constant Shift Embedding

We have introduced the cost function H^{pc} as a special instance of pairwise clustering problems. Due to the shift-invariance property (7.6), the partitioning of the

dataset (i.e., the assignments of a set of n objects to k clusters) is not affected by a constant additive shift on the off-diagonal elements of the pairwise dissimilarity matrix $D = (D_{ij}) \in \mathbb{R}^{n \times n}$. In the remainder of this paper, we will consider general symmetric dissimilarity matrices D , restricted only by the constraint that all self-dissimilarities are zero, i.e., that D has zero diagonal elements. We show that by exploiting the above shift invariance we can always embed such data into a Euclidean space without influencing the cluster structure. An off-diagonal shifted dissimilarity matrix reads

$$\tilde{D} = D + d_o(e_n e_n^t - I_n), \quad (7.7)$$

where $e_n = (1, 1, \dots, 1)^t$ is an n -vector of ones and I_n the $n \times n$ identity matrix. In other words, (7.7) describes a constant additive shift $\tilde{D}_{ij} = D_{ij} + d_o$ for all $i \neq j$.

Before developing the main theory, we have to introduce the notion of a *centralized matrix*. Let P be an $n \times n$ matrix and let $Q = I_n - \frac{1}{n}e_n e_n^t$. Q is the projection matrix on the orthogonal complement of e_n . Define the *centralized* P by

$$P^c = Q P Q. \quad (7.8)$$

A centralized matrix has row- and column-sum equal to zero, which can easily be seen by looking at the components of P^c

$$P_{ij}^c = P_{ij} - \frac{1}{n} \sum_{k=1}^n P_{ik} - \frac{1}{n} \sum_{k=1}^n P_{kj} + \frac{1}{n^2} \sum_{k,l=1}^n P_{kl}. \quad (7.9)$$

Let us now consider symmetric dissimilarity matrices. Given such a symmetric and zero-diagonal matrix D , we decompose it the following way by introducing a new matrix S :

$$D_{ij} = S_{ii} + S_{jj} - 2S_{ij}. \quad (7.10)$$

It is clear that this decomposition is not unique unless we specify the diagonal elements of S . Let \mathbb{S}_D denote the equivalence class of all S yielding the same D . The following lemma states that for all members $S \in \mathbb{S}_D$ the centralized version S^c is identical and uniquely defined by the given matrix D :

Lemma 7.1 *For any symmetric and zero-diagonal matrix D , the following holds:*

$$S^c = -\frac{1}{2}D^c, \quad \text{with } D^c = Q D Q.$$

The matrix S^c is a particularly interesting member of \mathbb{S}_D , since the following theorem holds:

Theorem 7.1 *D derives from a squared Euclidean distance, i.e., $D_{ij} = \|x_i - x_j\|^2$, if and only if S^c is positive semi-definite.*

Proof See [10] referring to [11]. \square

For general dissimilarities, S^c will be indefinite. By shifting its diagonal elements, however, we can transform it into a positive semi-definite matrix: the following lemma states that for any matrix A , a positive semi-definite matrix \tilde{A} can be derived by subtracting the smallest eigenvalue from all of its diagonal elements:

Lemma 7.2 *Let $\tilde{A} = A - \lambda_n(A)I_n$, where $\lambda_n(\cdot)$ is the minimal eigenvalue of its argument. Then \tilde{A} is positive semi-definite.*

Proof Due to the diagonal shift, the smallest eigenvalue becomes zero. \square

We can now summarize the above results: given a matrix D , there exists a unique matrix S^c by Lemma 7.1. If S^c is not positive semi-definite, Lemma 7.2 states that by subtracting $\lambda_n(S^c)$ from its diagonal elements, we obtain a positive semi-definite \tilde{S} . Returning to (7.10) with our fixed matrix S^c , such a diagonal shift of S^c corresponds to an *off-diagonal* shift of the dissimilarities

$$\tilde{D}_{ij} = \tilde{S}_{ii} + \tilde{S}_{jj} - 2\tilde{S}_{ij} \quad \Leftrightarrow \quad \tilde{D} = D - 2\lambda_n(S^c)(e_n e_n^t - I_n). \quad (7.11)$$

In other words, if we were given \tilde{D} instead of our original D , then \tilde{S} would be a positive semi-definite member of the equivalence class $\mathbb{S}_{\tilde{D}}$ of matrices fulfilling the decomposition $\tilde{D}_{ij} = \tilde{S}_{ii} + \tilde{S}_{jj} - 2\tilde{S}_{ij}$. Theorem 7.1 then tells us that this off-diagonally shifted matrix \tilde{D} derives from a squared Euclidean distance. Since every positive semi-definite matrix is a dot product (or *Gram*) matrix in some vector space, there exists a matrix X of vectors such that $\tilde{S} = XX^t$. The matrix \tilde{D} then contains squared Euclidean distances between these vectors. We can now insert \tilde{D} into our clustering procedure (which is assumed shift-invariant), and we will obtain the same partition of the objects as if we had clustered the original matrix D . Contrary to directly using D , however, the matrix \tilde{D} now contains squared Euclidean distances between a set of vectors $\{x_i\}_{i=1}^n$. The vectors themselves can be reconstructed by way of kernel PCA, see [12].

A k -Means Formulation for Pairwise Clustering It is well-known that for the special case of squared Euclidean distances, the Pairwise cost function and the k -means cost function can be transformed into each other by using a simple algebraic identity, see above. The invariance property in Eq. (7.6), however, implies that a similar relationship between both cost functions holds in the general setting:

Theorem 7.2 *Given an arbitrary $n \times n$ dissimilarity matrix D with zero self-dissimilarities, there exists a transformed matrix \tilde{D} such that*

- (i) *The matrix \tilde{D} can be interpreted as a matrix of squared Euclidian distances between a set of vectors $\{x_i\}_{i=1}^n$ with dimensionality $\dim(x_i) \leq n - 1$;*

- (ii) *The original pairwise clustering problem defined by the cost function $H^{\text{pc}}(D)$ is equivalent to the k -means problem with cost function H^{km} in this vector space, i.e., the optimal cluster assignment variables \hat{M}_{i_V} are identical in both problems: $\hat{M}^{\text{pc}}(D) = \hat{M}^{\text{km}}(\tilde{D})$.*

7.3 A Probabilistic Generalization: the Wishart–Dirichlet Cluster Process

Despite its elegance, the approach described above is particularly tailored to certain hard-clustering cost functions like the pairwise k -means function. Here we go one step further and reformulate the matrix partitioning problem in a fully probabilistic framework. Clustering with such models can be viewed as a low-rank matrix approximation, and approximate shift invariance can be explained as a natural consequence of assuming a white noise term capturing the deviations from the low-rank model. In the hard-clustering limit, the k -means model with its known invariance properties appears as a special case of this class of models.

This section is structured as follows: we first review the partitioning model for Gaussian mixtures introduced in [13], which is then extended to a partitioning process on matrices. Connections to multi-dimensional scaling are shown which help to explain the clustering process as a low-rank matrix approximation. Finally, shift invariance properties are analyzed, and the model is tested both on synthetic and real-world data. For further technical details the reader is referred to [14, 15].

7.3.1 Gauss–Dirichlet Cluster Process

Let $[n] := \{1, \dots, n\}$ denote an index set, and \mathbb{B}_n the set of partitions of $[n]$. A partition $B \in \mathbb{B}_n$ is an equivalence relation $B : [n] \times [n] \rightarrow \{0, 1\}$ that may be represented in matrix form as $B(i, j) = 1$ if $x(i) = x(j)$ and $B(i, j) = 0$ otherwise, with x being a function that maps $[n]$ to some label set \mathbb{L} . Alternatively, B may be represented as a set of disjoint non-empty subsets called “blocks” b . A *partition process* is a series of distributions P_n on the set \mathbb{B}_n in which P_n is the marginal distribution of P_{n+1} . Such a process is called *exchangeable* if each P_n is invariant under permutations of object indices, see [16] for details.

A *Gauss–Dirichlet cluster process* consists of an infinite sequence of points in \mathbb{R}^d , together with a random partition of integers into k blocks. A sequence of length n can be sampled as follows, cf. [13, 17, 18]: fix the number of mixture modes k , generate mixing proportions $\pi = (\pi_1, \dots, \pi_k)$ from an exchangeable Dirichlet distribution $\text{Dir}(\lambda/k, \dots, \lambda/k)$, generate a label sequence (x_1, \dots, x_n) from a multinomial distribution, and forget the labels introducing the random

partition B of $[n]$ induced by x . Integrating out π , one arrives at a Dirichlet–Multinomial-type prior over partitions:

$$P_n(B|\lambda, k) = \frac{k!}{(k - k_B)!} \frac{\Gamma(\lambda) \prod_{b \in B} \Gamma(n_b + \lambda/k)}{\Gamma(n + \lambda) [\Gamma(\lambda/k)]^{k_B}}, \quad (7.12)$$

where $k_B \leq k$ denotes the number of blocks present in the partition B and n_b is the size of block b . The limit as $k \rightarrow \infty$ is well defined and known as the Ewens process (a.k.a. Chinese Restaurant process); see, for instance, [19–21]. Given such a partition B , d -dimensional observations $Y = (Y_1, \dots, Y_n)$ are generated from a zero-mean Gaussian distribution with covariance matrix

$$\Sigma_B = I_n \otimes \Sigma_0 + B \otimes \Sigma_1, \quad \text{with } \text{cov}(\mathcal{Y}_{ir}, \mathcal{Y}_{js}|B) = \delta_{ij} \Sigma_{0rs} + B_{ij} \Sigma_{1rs}, \quad (7.13)$$

where Σ_0 is the usual within-class covariance matrix and Σ_1 the between-class matrix, respectively. Since the partition process is invariant under permutations, we can always think of B being block-diagonal. For spherical covariance matrices, $\Sigma_0 = \alpha I_d$, $\Sigma_1 = \beta I_d$, the columns of \mathcal{Y} contain independent copies distributed according to a normal distribution with covariance matrix $\Sigma_B = \alpha I + \beta B$. Further, the distribution also factorizes over the blocks $b \in B$. Introducing for each block an $n_b \times n_b$ -matrix of ones E_{n_b} , the joint distribution of data and partitions reads

$$p(Y, B|\alpha, \beta, \lambda, k) = \left[\prod_{b \in B} \prod_{j=1}^d N(Y_{i_b j} | \alpha I_{n_b} + \beta E_{n_b}) \right] \cdot P(B|\lambda, k), \quad (7.14)$$

where the symbol i_b defines an index-vector for all objects assigned to block b .

7.3.2 Wishart–Dirichlet Cluster Process

We now extend the Gauss–Dirichlet cluster process to a sequence of inner-product and distance matrices. Assume that the random matrix $\mathcal{Y}_{n \times d}$ follows the zero-mean Gaussian distribution specified in (7.13), with $\Sigma_0 = \alpha I_d$, $\Sigma_1 = \beta I_d$. Then, conditioned on the partition B , the inner product matrix $\mathcal{S} = \mathcal{Y} \mathcal{Y}^t / d$ follows a (possibly singular) Wishart distribution with d degrees of freedom, $\mathcal{S} \sim \mathcal{W}_d(\Sigma_B)$ [22]. If we directly observe $\mathcal{S} = S$ (i.e., if we measure similarities expressed as a Mercer kernel matrix), it suffices to consider the conditional probability of partitions, $P_n(B|S)$, which has the same functional form for ordinary and singular Wishart distributions. Due to the block structure in B , $P_n(B|S)$ factorizes over the blocks $b \in B$:

$$P_n(B|S, \alpha, \beta, \lambda, k) \propto \left[\prod_{b \in B} |\Sigma_b|^{-\frac{d}{2}} \exp\left(-\frac{d}{2} \text{tr}(\Sigma_b^{-1} S_b)\right) \right] \cdot P_n(B|\lambda, k), \quad (7.15)$$

where Σ_b, S_b denote the submatrices corresponding to the b th block.

Often, however, we do not directly observe S , but only a matrix D of squared distances with components $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$. Note that S determines D , but not vice versa, since D is constant on equivalence classes of S resulting from the arbitrariness of the mean vector. A squared distance matrix D is characterized by the property of being *negative definite on contrasts*, which means that $\mathbf{x}^t D \mathbf{x} = -\frac{1}{2} \mathbf{x}^t S \mathbf{x} < 0$ for any $\mathbf{x} : \mathbf{x}^t \mathbf{1} = 0$. The distribution of D has been formally studied in [23], where it was shown that if $\mathcal{S} \sim \mathcal{W}_d(\Sigma_B)$, $-D$ follows a generalized Wishart distribution, $-\mathcal{D} \sim \mathcal{W}_d(\mathbf{1}, \Delta)$ defined with respect to the transformation kernel $\mathbf{1}$, where $\Delta_{ij} = \Sigma_{Bii} + \Sigma_{Bjj} - 2\Sigma_{Bij}$. As before, the transformation kernel has the effect that the distribution of D is constant on equivalence classes. Since we are interested in studying the partition B given an observed matrix D , it is convenient to forego the equivalence classes by explicitly choosing a representation S which fulfills $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$. We can again use the projection Q with $Q_{ij} = \delta_{ij} - \frac{1}{n}$ to transform D into centered inner product form via $S = -\frac{1}{2} Q D Q$, which eliminates contributions of the mean vector while preserving the distances. Formally, this choice is justified by the observation that D is a matrix of squared distances if and only if $S = -\frac{1}{2} Q D Q$ is positive semi-definite [10].

Relation to Multi-Dimensional Scaling Classical multi-dimensional scaling [24] can be interpreted as using a distance model

$$-D \sim \mathcal{W}(\mathbf{1}, \Delta) \quad \text{with } \Delta = \Delta_0 - M - \sigma^2 I, \quad (7.16)$$

where Δ_0 stems from the transformation kernel $\mathbf{1}$, M is a low-rank matrix used to approximate the observed matrix D , and $\sigma^2 I$ is a white noise term accounting for deviations from the low-rank model, see [23]. As before, the transformation $S = -Q D Q$ eliminates the contribution of the kernel and transforms the data into inner product form. The matrix M is then computed as the best low-rank approximation to S , which is the rank-constrained maximum likelihood solution in the Wishart model, see [23]. The above expression $\Sigma = \sigma^2 I + M$ is essentially the same as our covariance model $\Sigma_B = \alpha I + \beta B$. The only difference is that B is not an arbitrary low-rank matrix, but additionally constrained to be a binary partition matrix. Thus, our partitioning model can be understood as a binarized version of multi-dimensional scaling. The white noise term αI corresponding to the within-class covariance has the role of absorbing the deviations from the low-rank model.

Shift Invariance The expected value of $\mathcal{S} \sim \mathcal{W}_d(\Sigma_B)$ is $E[\mathcal{S}] = \Sigma_B$. Adding an additional noise term δI shifts the expected value to $\Sigma_B + \delta I$. Reversing this argument for inference problems in which we observe the inner product matrix S , additive shifts of the diagonal elements of S might be absorbed by the white noise term. Note that such additive diagonal terms appear when shifting the *off*-diagonal elements of D . Using sufficiently large shifts ensures that there exists an embedding space in which the transformed dissimilarities D' can be represented as squared Euclidean distances. The idea behind additive shifts is the following: if we observe a matrix D which gives rise to an indefinite matrix $S = -\frac{1}{2} Q D Q$, there are basically

two options: either we can directly use S , irrespective of negative eigenvalues, or we can try to “heal” the negative eigenvalues. Concerning the first option, it is unclear what bias is introduced due to the model mismatch. “Healing” the negative eigenvalues, on the other hand, introduces another sort of bias. In the ideal case, we can find a transformation which exploits some invariance of the analysis model. If the model is invariant under additive shifts, we can safely transform *any* (symmetric) matrix D in such a way that it will be inside the model space. Note that for our clustering model, even symmetry is not required, since all conditionals are invariant under $S \leftarrow 1/2(S + S')$. We first show that exact shift invariance is possible, but only under assumptions that eliminate the probabilistic nature of the model.

The inverse matrix $\Sigma_b^{-1} = (\alpha I_{n_b} + \beta E_{n_b})^{-1}$ can be analytically computed as

$$\frac{1}{\alpha} \left(I - \frac{\beta}{\alpha + n_b \beta} E_{n_b} \right) = \frac{1}{\alpha} \left(I - \frac{\theta}{1 + n_b \theta} E_{n_b} \right) \quad \text{with } \theta := \beta/\alpha. \quad (7.17)$$

Denoting by $\frac{d}{2}A$ the argument of the exponential function in (7.15), a shift $S' = S + \delta I_n$ implies

$$\begin{aligned} \alpha A' &:= -\alpha \operatorname{tr}((\alpha I_{n_b} + \beta E_{n_b})^{-1} (S_b + \delta I)) \\ &= \frac{n_b \theta}{1 + n_b \theta} (n_b \bar{S}_b + \delta) - \operatorname{tr}(S_b) - n_b \delta, \end{aligned} \quad (7.18)$$

where \bar{S}_b denotes the mean value of the b th block of S . For $\alpha \rightarrow 0$, it follows that

$$\begin{aligned} \alpha A' &\approx n_b \bar{S}_b - \operatorname{tr}(S_b) - (n_b - 1)\delta \\ \Rightarrow \sum_{b \in B} \alpha A' &\approx -\operatorname{tr}(S) - (n - k_B)\delta + \sum_{b \in B} n_b \bar{S}_b \end{aligned} \quad (7.19)$$

with k_B being the number of blocks in the partition B . This result implies that for fixed α, θ, k_B , the conditional posterior of partitions is approximately shift invariant. In the hard-clustering limit as $\alpha \rightarrow 0$, this statement becomes exact. The price for exact shift invariance is the problem of estimating k_B . The restriction to hard assignments precludes an intrinsic measure of “clusterability”: the model degenerates to a combinatorial optimization problem in which we need to fix k . The optimal solution will then automatically include all $k_B = k$ blocks. Note that the limit $\alpha \rightarrow 0$ defines the *pairwise clustering* cost function [8] whose invariance properties have been studied in [3].

Here, we consider more realistic situations in which both the covariance parameters and k_B are estimated. Intuitively, we assume that shifts are “absorbed” in the within-class term, i.e., $\alpha' = \alpha + \delta$. Analytically studying the effects on the partition when both α and θ are varying is complicated, in particular due to the influence of the normalization term $|\Sigma_b|^{-(d/2)}$ in (7.15). Thus, we only consider an idealized scenario in which the matrix S has a distinct cluster structure which is consistent with our model. In such a case, there will be a matrix $\Sigma' = \alpha' I + \beta B'$ that

is reasonably close to the observed S , and the ML-estimate of the covariance matrix in the Wishart model is $\hat{\Sigma}_B \approx \Sigma' = \alpha' I + \beta B'$. If there is an additional shift $S^{\text{shifted}} = S + \delta I$, the ML-estimate will be $\hat{\Sigma}_B^{\text{shifted}} \approx (\alpha' + \delta) I + \beta B'$. The normalization term, however, decreases, indicating that the distribution is smeared out due to the increased noise term. Note that we have neglected the influence of the prior $P_n(B)$ defined in Eq. (7.12). For moderate shifts, however, the deviations from “local” uniformity might be reasonably small. Despite the approximate nature of this plausibility argument, our simulation experiments nicely corroborate the intuition that moderate shifts can be absorbed in the white-noise term—at least if the data exhibits a clear cluster structure. In practice, however, observed matrices only rarely show a distinct block structure, and the additional noise component introduced by large shifts severely hampers the estimation of a stable partition, both for our probabilistic model and for the hard-clustering counterpart. Thus, the real benefit of any form of shift invariance might be a justification for first transforming the data into inner product form and then applying (kernel-)PCA-denoising to eliminate the additional noise, which is exactly the approach suggested in [25].

Inference via Gibbs Sampling The main idea in Gibbs sampling is to iteratively sample parameter values from the full conditionals. For the sake of simplicity, we only consider the update equations for the partition B . Assume that n objects in S have already been partitioned according to B . Conditioning on S and B , we want to compute the assignment probabilities for a *new* object o_* , characterized by an additional row and column in the augmented matrix S_* . Due to permutation invariance, we can always assume that S_* is ordered according to blocks in B and that the additional row/column is the last one in some block. Either the new object is assigned to an existing block b , i.e., $o_* \rightarrow b \in B$, or it is assigned to a new block which will be denoted by $o_* \rightarrow \emptyset$.

Consider first the case $o_* \rightarrow b \in B$. Assume that the new row/column is the last one in this block. The number of objects in block b is increased by one, i.e., $n_b^* = n_b + 1$, and the new block mean is denoted by \bar{S}_b^* . With a slight abuse of notation, we write S_{*j} for $S_{n_b+1,j}^*$ and S_{**} for S_{n_b+1,n_b+1}^* . All symbols without (*) refer to the old state with n objects. Denote by $\frac{d}{2} A^*(b)$ the new argument in the exponential function in (7.15). Then,

$$A^*(b) = A + \frac{1}{\alpha} \left(\frac{(n_b + 1)\theta}{1 + (n_b + 1)\theta} (n_b + 1)\bar{S}_b^* - \frac{n_b\theta}{1 + n_b\theta} n_b\bar{S}_b + S_{**} \right). \quad (7.20)$$

Consider now the case of assigning o_* to a new cluster, i.e., $o_* \rightarrow \emptyset$. A new singleton cluster is added, i.e., $k_B^* = k_B + 1$. The associated argument in the exponential function becomes $A^*(\emptyset) = A + \frac{1}{\alpha} \left(\frac{\theta}{1+\theta} S_{**} + S_{**} \right)$. For the conditionals, we need to multiply the exponentiated terms above with the contributions of both the normalization term in (7.15) and the prior. Denoting these terms by $N^*(b)$ and $N^*(\emptyset)$, and

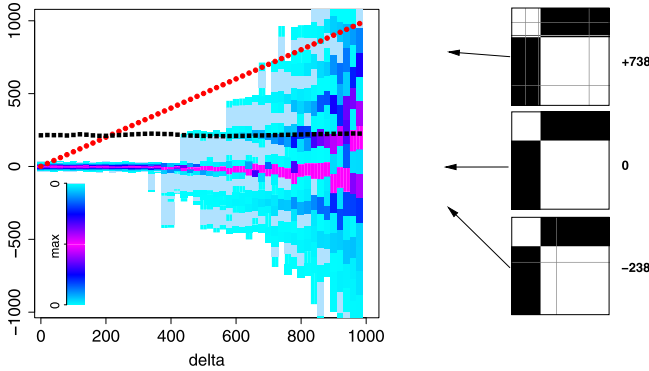


Fig. 7.1 Toy example for analyzing shift invariance. Linearly increasing *red circles* denote the mean values of estimated α -parameter under variation of the shift δ . Almost *horizontal black squares* show the mean values of estimated β -parameter (scaled by a factor of 10 for better visualization). *Color-coded histogram* provides differences between true and sampled partition. *Right panel* presents three sampled partitions

using $\Gamma(x + 1) = x\Gamma(x)$ in (7.12), we find

$$N^*(b) \propto \left[\frac{1 + \theta n_b}{1 + \theta(1 + n_b)} \right]^{(d/2)} \cdot (n_b + \lambda/k), \tag{7.21}$$

$$N^*(\emptyset) \propto (1 + \theta)^{-\frac{d}{2}} \cdot \lambda(1 - k_B/k).$$

7.3.3 Experiments

In a first experiment, we analyze the shift-invariance based on a matrix sampled from $\mathcal{W}(\Sigma_B)$ with a two-block partition (30 %/70 %) and $\alpha = 1, \theta = 20$. Using relatively uninformative priors on α and θ , we add increasing shifts δI to S . To compensate for δ , we adjust the priors over α and θ by shifting their expected value accordingly. Figure 7.1 shows that over a large range of δ -values, the shift is indeed absorbed in α , and the estimate for $\beta = \alpha \cdot \theta$ is roughly constant. Deviations from the “true” partition are summarized in the expression $\sum_{ij} (B_{ij}^{\text{true}} - B_{ij}^{\text{sampled}})$. Note that even for large shifts ($\delta = 1000$ is roughly 25 % of the largest eigenvalue of S), the partition remains rather stable. It is clear that we consider an idealized scenario, but nevertheless we conclude that our intuition about absorbing shifts seems to be correct. In this experiment, the influence of λ is extremely small: λ can be changed over at least 10 orders of magnitude without affecting k_B .

In the two following experiments, we quantitatively investigate the clustering performance in terms of the size-normalized within-sum-of-squared errors (distances), $\text{SSE} = \sum_{b \in B} n_b \bar{D}_b$, and compare the outcome with the Affinity Propagation (AP) method based on two datasets described in [26]. The first dataset contains similarities between 900 face images from the Olivetti database, available at

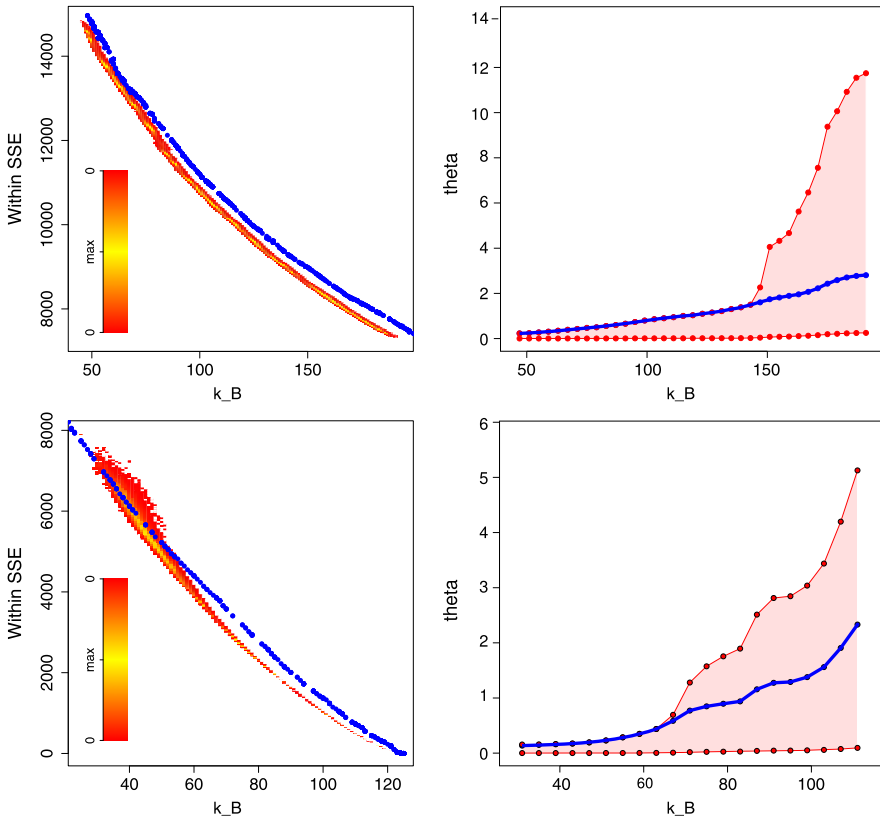


Fig. 7.2 Clustering the face dataset (*top row*) and text dataset (*bottom row*) from [26]. (*Left*) Within SSE obtained from Affinity Propagation with different affinity-parameter values (*blue circles*) and from our algorithm under variation of the prior on θ (*color-coded histogram*). (*Right*) Sampled θ values (*blue*) and range of possible θ values in the discretized prior (*reddish-shaded area*)

<http://www.psi.toronto.edu/index.php?q=affinity%20propagation>. AP has been reported to exhibit some advantages over other centroid-based approaches on this dataset. The results in Fig. 7.2 (top row) show that our model consistently outperforms AP, which means that better centroids have been found (note, however, that in this comparison our model has the advantage of not being restricted to choosing exemplars as centroids). Nevertheless, we conclude that in terms of optimization quality, the Wishart–Dirichlet model is a strong competitor to AP. Even more important, however, is the observation that all partitions with $k_B < 100$ are very implausible, because such a low number of clusters can only be obtained by “forcing” the model to use a very low θ -value via the prior, see the right panel: sampled values “hitting” the upper boundary of admissible values indicate that the model is entirely forced into a certain direction by the prior. Note that θ is the quotient of between-class to

within-class variance, and $\theta < 1$ means that there is hardly any cluster structure in the data.

Using the AP model, on the other hand, we can simply “slide” through all k_B -values by changing the “affinity”-parameter from -74 to -15 . From the AP model alone, we find it difficult to see why one of these results should be preferred over any other one (in [26], the model with $k_B = 62$ has been chosen for further analysis). The computational workload is not really an issue in this example, since even several millions of Gibbs sweeps can be computed reasonably fast (i.e., over night). A similar situation occurs for another dataset containing KL-divergences between sentences in a manuscript, which was used in [26] to demonstrate the performance of AP in situation where metric axioms are violated. Figure 7.2 (bottom row) clearly shows that (after symmetrizing an shifting) our model is a strong competitor in terms of optimization quality. The right panel again indicates that models with a low number of clusters (say < 70) are not very plausible due very small θ -values.

7.3.4 Wishart–Dirichlet Partitioning for Quality Control in Computational Pathology

Motivation A main challenge in computational pathology is the automated analysis of tissue microarrays (TMA). TMAs consist of tissue samples from hundreds of patients which can be stained with various antibodies for protein expression analysis. An automated analysis pipeline consists of three major steps: (i) cell nuclei detection, (ii) nuclei classification into malignant and benign, and (iii) staining estimation (see Chap. 9 for details). The resulting estimation per patient can then be used to correlate marker expression with the survival times or other clinical variables.

The most crucial step in the analysis pipeline is the classification of nuclei into malignant and benign, because the subsequent staining estimation has to be performed only on the subgroup of cancerous nuclei. Proliferation markers like MIB-1 (Ki-67) stain cell nuclei shortly before and after mitosis. The percentage of stained cancerous nuclei is one of the best prognostic factors for the survival of cancer patients [27], due to the fact that it directly relates to aggressiveness of the disease. As a consequence the differentiation between malignant and benign nuclei directly affects the final survival model for cancer patients. Stained benign nuclei, which were falsely classified, can considerably worsen the survival prediction in a domain, where small differences in the low percentage regime have a large impact on the progression of the disease.

Previous approaches [28] to automatic TMA analysis demonstrated that reasonable nuclei detection and staining estimation is possible and approaches the performance of trained pathologists. The main drawback of these models is the requirement for (almost) perfectly processed TMA spots. The predominant problem in clinical practice is the high variability between and within single spots, respectively

patients. Noise and variations are not only imposed by biology but also by technical preprocessing which comprises error prone steps like micro-cutting, punching of TMA spots and staining, which involves applying antibodies and microwaving of the tissue. The final step comprises scanning of the microscope slides and tiling of the TMA into single spots. All these steps lead to biological, technical and digital artifacts in the images resulting in distorted, blurred or obfuscated regions. Thus, trained pathologists do not take into account the whole spot when manually analyzing TMAs, but restrict themselves to regions of high quality only. This preference could also be observed during extensive labeling experiments for generating a “gold standard”. Forcing pathologist to classify randomly drawn nuclei led not only to high inter-pathologist variability but also to high intra-pathologist variability ($\sim 25\%$). To this end, the main goal in this application scenario is to create an algorithm which is robust to tissue variations by mimicking the work-flow of trained domain experts. The technical tool used for developing such an algorithm is a probabilistic model for partitioning dissimilarly matrices. The next section contains a detailed description of this model.

Quality Control in Computational Pathology The dataset consists of 500 cancerous nuclei and 500 normal nuclei sampled from TMA spots of 9 clear cell renal cell carcinoma patients (ccRCC). The spots were exhaustively labeled by a trained pathologist to generate a gold standard.

To differentiate between malignant and benign cell nuclei a Random Forest (RF) classifier [12] is trained. Each sample consists of a patch of size 65×65 pixels, centered at the nucleus. Local Binary Patterns (LBP) [30] are extracted from the gray scale images to form a feature vector of size 256 for each sample. LBPs have the advantage of illumination invariance, i.e., they are invariant with respect to monotonic gray-scale changes. A random forest classifier consists of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, \dots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input \mathbf{x} . One beneficial property of RFs is the internal out-of-bag (OOB) error which provides an unbiased estimate of the generalization error and which is reported in the following evaluation.

Learning an RF on the whole data set leads to an OOB error of 36 % (Fig. 7.3). Hence every third subsequent staining estimation is performed on a falsely classified nucleus. To enhance these results, we follow the analysis strategy of pathologists by excluding detection regions with poor quality. To this end, we use our matrix partitioning model to find subgroups of cell nuclei. A 1000×1000 similarity matrix is generated by measuring the proximity of nuclei in the tree ensemble [12]. All training samples and OOB samples are put down each tree of the ensemble. If two samples end up in the same terminal node, their proximity is increased by one. Finally, the similarities are normalized by the number of trees. The resulting matrix shows some negative eigenvalues, which are dealt with by using the shift-embedding trick, followed by PCA-denoising (Fig. 7.4).

Our clustering model reveals five stable clusters shown in Fig. 7.5. Most of the clusters can be interpreted semantically. For instance, cluster 3 contains large nuclei

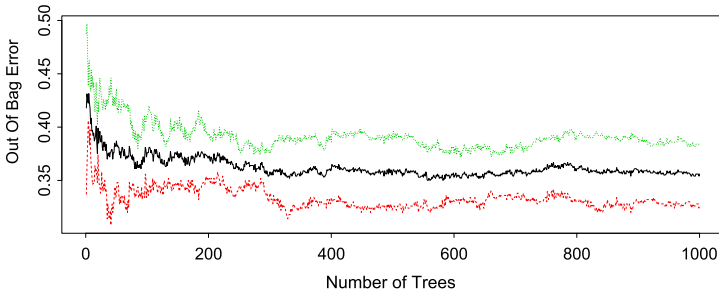
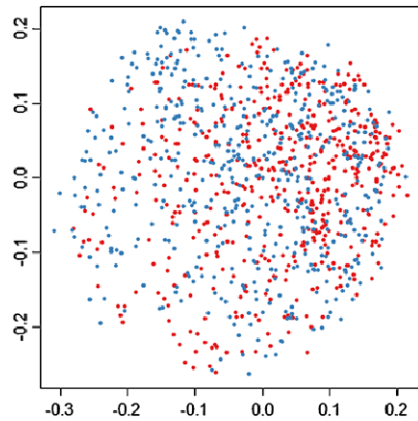


Fig. 7.3 Out-of-bag (OOB) error of the random forest classifier for the whole dataset. The error converges after approximately 300 trees to an average classification error (*black*) of approximately 36 %: (*red*) OOB error of the malignant class; (*green*) OOB error of the benign class

Fig. 7.4 Multidimensional scaling of the random forest proximity matrix. Visually there are no discernible clusters of cancerous (*red*) or normal (*blue*) nuclei



which are clearly distinguishable from background. Such morphology is articulated in cancerous nuclei which are no longer embedded in cohesive tissue. Cluster 2, on the other hand, comprises small and elongated nuclei on cluttered background. This is characteristic for benign nuclei in healthy tissue and for endothelial cells in connective tissue. These observations are consistent with the observed distribution of labels (Fig. 7.6). In contrast, the patches in cluster 5 are either blurred, distorted or show no clear structure. This lack of semantic meaning is mainly caused by technical processing flaws which result in regions, which cannot be classified reliably, although the pathologists detected remnants of nuclei. Cluster 1 and 4 vary largely in tissue morphology and image quality. Consequently, these three clusters show an almost uniform distribution of malignant and benign cells. A computational TMA analysis tool should reject such nuclei, in the same manner as a domain expert would go about it to avoid contamination of the whole patient sample (Fig. 7.7). Proceeding at these lines, an RF classifier is trained on the subset of nuclei from cluster 2 and 3, resulting in an OOB error of 19.4 %. This significant reduction nicely demonstrates the importance of quality assessment preceding classification. We are

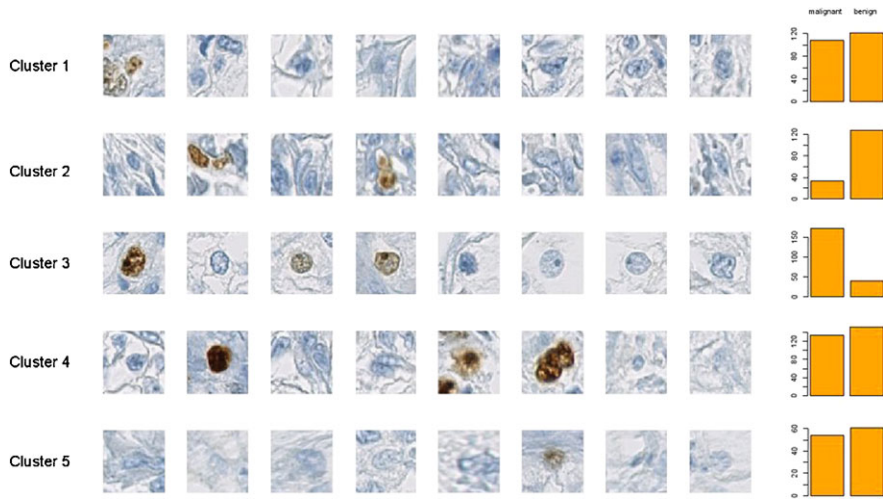


Fig. 7.5 Cluster of cell nuclei from renal cell carcinoma patients. (*Left*) Randomly drawn nuclei from the five cluster revealed by W-D-clustering. (*Right*) Within-cluster distribution of cancerous and benign nuclei. For most cluster the semantic interpretation of pathologists is in agreement with the class distribution, e.g., cluster 3 consists mainly of large nuclei which are clearly distinguishable from background. This kind of morphology is articulated in cancerous cells which are no longer embedded in cohesive tissue. Cluster 2, on the other hand, comprises small and elongated nuclei on cluttered background. This is characteristic for benign nuclei in healthy tissue and for endothelial cells in connective tissue. In contrast, the patches in cluster 5 are either blurred, distorted or show no clear structure which is mainly due to technical processing flaws which lead to image regions of poor quality. In addition, clusters 1 and 4 show no interpretable pattern and vary largely in tissue morphology and image quality. These three cluster also exhibit a uniform distribution of malignant and benign cells

convinced that this data cure approach is the key to solving one of the most severe problems in the design of computational TMA analysis tools.

7.4 Conclusion

A partitioning model is called shift invariant, if the choice of a partition is not influenced by additive constant shifts of the off-diagonal elements in D . If a model exhibits this invariance property, it is always possible to construct an underlying Euclidean embedding space without altering the partition, a situation which we describe as “structure preserving embedding”.

We have shown that the pairwise k -means cost function exhibits strict shift invariance, which—in terms of group structure— defines a structure preserving embedding model. However, this analysis is restricted to a certain cost function, and in particular to considering scenarios in which the number of clusters k is defined in advance. The latter requirement must be considered a severe shortcoming in most real applications, because information about the number of clusters usually rare. Therefore,

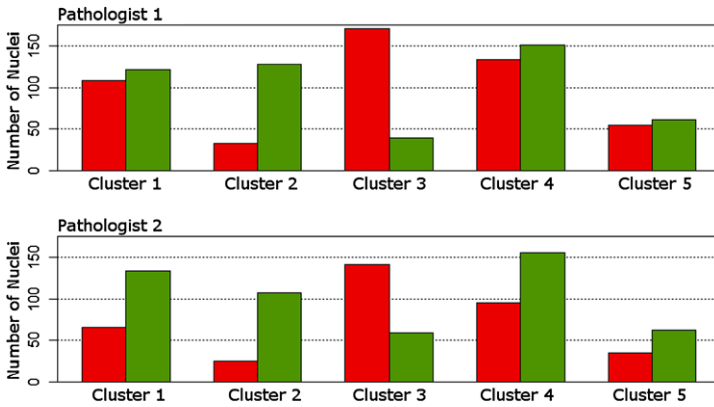


Fig. 7.6 Label distribution of each cluster per pathologist. Both experts agree that cluster 2 consists predominantly of normal nuclei, while cluster 3 comprises mostly cancerous nuclei

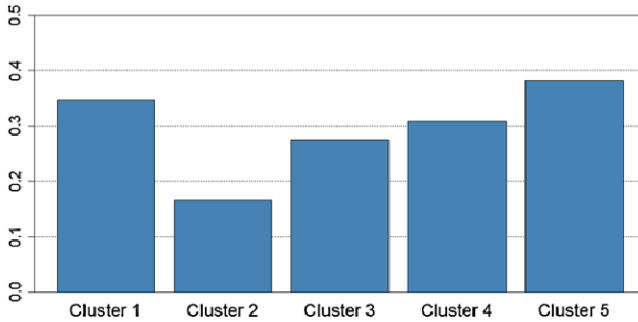


Fig. 7.7 Inter-observer misclassification error of two pathologists for each of the five clusters determined by Wishart–Dirichlet partitioning

we tried to broaden our viewpoint on pairwise clustering by considering a probabilistic version of the pairwise k -means model. The main idea is to construct a stochastic process on similarity matrices and use a Dirichlet process prior to estimate the number of “blocks” in a partitioning matrix. Concerning structure preserving embeddings defined by constant-shift embeddings, we have shown that the clustering model induced by this Wishart–Dirichlet model can absorb “moderate” shifts in the white-noise term. However, a particular problem of this model is that the process of estimating the number of clusters in a data-adaptive fashion is also affected by the shift: shifting increases the tendency to introduce new clusters, since under the shift the mutual similarities between all objects decrease. It seems that strict shift invariance can only be achieved if the number of clusters is fixed, which somehow contradicts our efforts to generalize the k -means setting.

Considering the relevance of structure preserving embedding for the overall goal of the SIMBAD project, namely the development of a new theory of similarity-based pattern recognition, our current view is ambivalent: strict structure preser-

vation could be proved only for a small set of clustering methods, like pairwise k -means and certain graph-based cut/association algorithms. All these algorithms require the user to fix the number of clusters in advance. A “relaxed” version of shift invariance holds for a probabilistic version of the pairwise k -means method, but we have to admit that shift invariance and estimation of the number of clusters might be two conflicting goals. As an alternative the number of clusters k can be estimated by the information theoretic approach to cluster validation using approximation set coding (ASC) (see Chap. 3 and [31]).

When it comes to building a theory on similarity-based pattern recognition, all these algorithms may be seen as “negative results”, since they are essentially blind against Euclidean or even metric violations. In other words, if one wants to learn something about clustering similarity data, one should look at different clustering procedures. While this result may be considered an interesting insight, it is still a very limited result due to the small number of algorithms that could be identified to fall into this category.

References

1. Jolliffe, I.T.: *Principal Component Analysis*. Springer, New York (1986)
2. Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw.* **12**(2), 181–201 (2001)
3. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.M.: Optimal cluster preserving embedding of non-metric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(12) (2003)
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley, New York (2001)
5. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* **31**(3), 264–323 (1999)
6. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
7. Puzicha, J., Hofmann, T., Buhmann, J.: A theory of proximity based clustering: structure detection by optimization. *Pattern Recognit.* **33**(4), 617–634 (1999)
8. Hofmann, T., Buhmann, J.: Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(1), 1–14 (1997)
9. Brucker, P.: On the complexity of clustering problems. In: Beckman, M., Kunzi, H.P. (eds.) *Optimization and Operations Research: Lecture Notes in Economics and Mathematical Systems*, pp. 45–54. Springer, Berlin (1978)
10. Torgerson, W.S.: *Theory and Methods of Scaling*. Wiley, New York (1958)
11. Young, G., Householder, A.S.: Discussion of a set of points in terms of their mutual distances. *Psychometrika* **3**, 19–22 (1938)
12. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**(5), 1299–1319 (1998)
13. McCullagh, P., Yang, J.: How many clusters? *Bayesian Anal.* **3**, 101–120 (2008)
14. Vogt, J., Prabhakaran, S., Fuchs, T., Roth, V.: The translation-invariant Wishart–Dirichlet process for clustering distance data. In: *Proceedings of the 27th International Conference on Machine Learning* (2010)
15. Prabhakaran, S., Boehm, A., Metzner, K.J., Roth, V.: Recovering networks from distance data. *J. Mach. Learn. Res.* **25**, 349–364 (2012)
16. Pitman, J.: Combinatorial stochastic processes. In: Picard, J. (ed.) *Ecole d’Ete de Probabilites de Saint-Flour XXXII-2002*. Springer, Berlin (2006)

17. MacEachern, S.N.: Estimating normal means with a conjugate-style Dirichlet process prior. *Commun. Stat., Simul. Comput.* **23**, 727–741 (1994)
18. Dahl, D.B.: Sequentially-allocated merge-split sampler for conjugate and non-conjugate Dirichlet process mixture models. Technical report, Department of Statistics, Texas A&M University (2005)
19. Ewens, W.: The sampling theory of selectively neutral alleles. *Theor. Popul. Biol.* **3**, 87–112 (1972)
20. Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Stat.* **9**, 249–265 (2000)
21. Blei, D., Jordan, M.: Variational inference for Dirichlet process mixtures. *Bayesian Anal.* **1**, 121–144 (2006)
22. Srivastava, M.S.: Singular Wishart and multivariate beta distributions. *Ann. Stat.* (2003)
23. McCullagh, P.: Marginal likelihood for distance matrices. *Stat. Sin.* **19**, 631–649 (2009)
24. Cox, T.F., Cox, M.A.A.: *Multidimensional Scaling*. Chapman & Hall, London (2001)
25. Roth, V., Laub, J., Buhmann, J.M., Müller, K.-R.: Going metric: denoising pairwise data. In: Thrun, S., Becker, S., Obermayer, K. (eds.) *Advances in Neural Information Processing Systems*, vol. 15, pp. 817–824. MIT Press, Cambridge (2003)
26. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. *Science* **315**, 972–976 (2007)
27. Tannapfel, A., Hahn, H.A., Katalinic, A., Fietkau, R.J., Kühn, R., Wittekind, C.W.: Prognostic value of ploidy and proliferation markers in renal cell carcinoma. *Cancer* **77**(1), 164–171 (1996)
28. Fuchs, T.J., Wild, P.J., Moch, H., Buhmann, J.M.: Computational pathology analysis of tissue microarrays predicts survival of renal clear cell carcinoma patients. In: *Medical Image Computing and Computer-Assisted Intervention. MICCAI 2008. Lecture Notes in Computer Science*, vol. 5242, pp. 1–8. Springer, Berlin (2008)
29. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
30. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: *ECCV 2004*, vol. 3021, pp. 469–481 (2004)
31. Buhmann, J.M.: Information theoretic model validation for clustering. In: *International Symposium on Information Theory*, Austin Texas, pp. 1398–1402. IEEE Press, New York (2010). doi:[10.1109/ISIT.2010.5513616](https://doi.org/10.1109/ISIT.2010.5513616)

Chapter 8

A Game-Theoretic Approach to Pairwise Clustering and Matching

Marcello Pelillo, Samuel Rota Bulò, Andrea Torsello, Andrea Albarelli,
and Emanuele Rodolà

Abstract Clustering refers to the process of extracting maximally coherent groups from a set of objects using pairwise, or high-order, similarities. Traditional approaches to this problem are based on the idea of partitioning the input data into a predetermined number of classes, thereby obtaining the clusters as a by-product of the partitioning process. In this chapter, we provide a brief review of our recent work which offers a radically different view of the problem and allows one to work directly on non-(geo)metric data. In contrast to the classical approach, in fact, we attempt to provide a meaningful formalization of the very notion of a cluster in the presence of non-metric (even asymmetric and/or negative) (dis)similarities and show that game theory offers an attractive and unexplored perspective that serves well our purpose. To this end, we formulate the clustering problem in terms of a non-cooperative “clustering game” and show that a natural notion of a cluster turns out to be equivalent to a classical (evolutionary) game-theoretic equilibrium concept. Besides the game-theoretic perspective, we exhibit also characterizations of our cluster notion in terms of optimization theory and graph theory. As for the algorithmic issues, we describe two approaches to find equilibria of a clustering game. The first one is based on the classical replicator dynamics from evolutionary game theory, the second one is a novel class of dynamics inspired by infection and immu-

M. Pelillo (✉) · A. Torsello · A. Albarelli
DAIS, Università Ca' Foscari, Venezia, Italy
e-mail: pelillo@dais.unive.it

A. Torsello
e-mail: torsello@dais.unive.it

A. Albarelli
e-mail: albarelli@dais.unive.it

S. Rota Bulò
Fondazione Bruno Kessler, Povo, Trento, Italy
e-mail: samyrota@gmail.com

E. Rodolà
Intelligent Systems and Informatics Lab, University of Tokyo, Tokyo, Japan
e-mail: rodola@isi.imi.i.u-tokyo.ac.jp

nization processes which overcome their limitations. Finally, we show applications of the proposed framework to matching problems, where we aim at finding correspondences within a set of elements. In particular, we address the problems of point-pattern matching and surface registration.

8.1 Introduction

Clustering is the problem of organizing a set of data elements into groups in a way that each group satisfies an internal coherency and external incoherency property. Researchers have focused their attention on this problem for many decades due to its broad applicability, and recently a new wave of excitement has spread across the machine learning community mainly because of the important development of spectral methods. At the same time, there is also growing interest around fundamental questions pertaining to the very nature of the clustering problem (see, e.g., [1, 31, 60]). Yet, despite the tremendous progress in the field, the clustering problem remains elusive and a satisfactory answer even to the most basic questions is still to come.

The vast majority of the existing approaches deal with a very specific version of the problem, which asks for *partitioning* the input data into coherent classes. Even the classical distinction between hierarchical and partitional algorithms [28] seems to suggest the idea that partitioning data is, in essence, what clustering is all about (as hierarchies are but nested partitions). The partitional paradigm is attractive as it leads to elegant mathematical and algorithmic treatments and allows us to employ powerful ideas from such sophisticated fields as linear algebra, graph theory, optimization, statistics, information theory, etc. However, there are several (far too often neglected) reasons for feeling uncomfortable with this oversimplified formulation. Probably the best-known limitation of the partitional approach is the typical (algorithmic) requirement that the number of clusters be known in advance, but there is more than that.

To begin, the very idea of a partition implies that *all* the input data will have to get assigned to some class. There are various applications for which it makes little sense to force all data items to belong to some group, a process which might result either in poorly-coherent clusters or in the creation of extra spurious classes. As an extreme example, consider the classical figure/ground separation problem in computer vision which asks for extracting a coherent region (the figure) from a noisy background [24, 49]. More recently, motivated by practical applications arising in document retrieval and bioinformatics, a conceptually identical problem has attracted some attention within the machine learning community and is generally known under the name of one-class clustering [16, 23].

The second intrinsic limitation of the partitional paradigm is even more severe as it imposes that each element cannot belong to more than one cluster. There are a variety of important applications, however, where this requirement is too restrictive. Examples abound and include, e.g., clustering micro-array gene expression

data (wherein a gene often participate in more than one process), clustering documents into topic categories, perceptual grouping, and segmentation of images with transparent surfaces. Typically, this is solved by relaxing the constraints imposed by crisp partitions in such a way as to have “soft” boundaries between clusters.

Finally, stemming from a natural assumption for central clustering frameworks, clustering approaches have traditionally worked under the assumption that the similarities satisfy metric properties, i.e., they are symmetric, non-negative, and satisfy the triangle inequality. However, recently there has been a strong interest in relaxing these requirements [27, 46, 59]. This is due to the fact that in many applications non-metric similarities arise naturally [25, 58]. More fundamentally, some researchers argue that human perception does not satisfy metric properties [27]. While the literature presents many approaches that lift the assumption of non-negativity and triangle inequality [27, 46], little progress has been made in relaxing the symmetry constraint. Note, however, that the limited progress in grouping with asymmetric affinities is not due to the lack of interest. In fact, there are many practical applications where asymmetric (or, more generally, non-metric) similarities do arise quite naturally. For example, such (dis)similarity measures are typically derived when images, shapes or sequences are aligned in a template matching process. In image and video processing, these measures are preferred in the presence of partially occluded objects [27]. Other examples include pairwise structural alignments of proteins that focus on local similarity [5], variants of the Hausdorff distance [18], normalized edit-distances, and probabilistic measures such as the Kullback–Leibler divergence. A common method to deal with asymmetric affinities is simply to symmetrize them, but in so doing we might lose important information that reside in the asymmetry (see, e.g., [12]). As argued in [27], the violation of metricity is often not an artifact of poor choice of features or algorithms, but it is inherent in the problem of robust matching when different parts of objects (shapes) are matched to different images (compare this with the analysis presented in Chap. 2 concerning non-Euclidean data). The same argument may hold for any type of local alignments. Corrections or simplifications of the original affinity matrix of the type described in the previous chapters may therefore destroy essential information, and is therefore important to devise algorithms which are able to work directly on the original data.

Although probabilistic model-based approaches do not suffer from several of the limitations mentioned above, here we suggest an alternative strategy. Instead of insisting on the idea of determining a partition of the input data, and hence obtaining the clusters as a by-product of the partitioning process, we propose to reverse the terms of the problem and attempt instead to derive a rigorous formulation of the very notion of a cluster. We found that game theory offers a very elegant and general perspective that serves well our purposes. Hence, in this chapter we describe a game-theoretic framework for clustering [38, 43, 52] which has found applications in fields as diverse as computer vision and bioinformatics. The starting point is the elementary observation that a “cluster” may be informally defined as a maximally coherent set of data items, i.e., as a subset of the input data C which satisfies both an *internal* criterion (all elements belonging to C should be highly similar to each other) and an *external* one (no larger cluster should contain C as a proper subset).

We then formulate the clustering problem as a non-cooperative *clustering game*, where the notion of a cluster turns out to be equivalent to a classical equilibrium concept from (evolutionary) game theory, as the latter reflects both the internal and external cluster conditions mentioned above. The clustering game is defined as follows: Assume a pre-existing set of objects O and a (possibly asymmetric and even negative) matrix of affinities A between the elements of O . Two players with complete knowledge of the setup play by simultaneously selecting an element of O . After both have shown their choice, each player receives a payoff, monetary or otherwise, proportional to the affinity that the chosen element has with respect to the element chosen by the opponent. Clearly, it is in each player's interest to pick an element that is strongly supported by the elements that the adversary is likely to choose. As an example, let us assume that our clustering problem is one of figure/ground discrimination, that is, the objects in O consist of a cohesive group with high mutual affinity (figure) and of non-structured noise (ground). Being non-structured, the noise gives equal average affinity to elements of the figures as to elements of the ground. Informally, assuming no prior knowledge of the inclination of the adversary, a player will be better-off selecting elements of the figure rather than of the ground.

Within this framework, clusters correspond to the ESSs of our non-cooperative game. The hypotheses that each object belongs to a cluster compete with one-another, each obtaining support from compatible edges and competitive pressure from the others. Competition will reduce the population of individuals that assume weakly supported hypotheses, while allowing populations assuming hypotheses with strong support to thrive. Eventually, all inconsistent hypotheses will be driven to extinction, while all the surviving ones will reach an equilibrium whereby they will all receive the same average support, hence exhibiting the internal coherency characterizing a cluster. As for the extinct hypotheses, they will provably have a lower support, thereby hinting to external incoherency. The stable strategies can be found using *replicator dynamics*, a classic formalization of a natural selection process [26, 57], or more powerful algorithms.

Our game-theoretic formulation of the clustering problem overcomes the aforementioned limitations of the majority of the clustering approaches in the literature. Indeed, it makes no assumption on the underlying (individual) data representation: like graph-based clustering, it does not require that the elements to be clustered be represented as points in a vector space; it makes no assumption on the structure of the affinity matrix, being able to work with asymmetric and even negative similarity functions alike; it does not require a priori knowledge on the number of clusters (since it extracts them sequentially); it leaves clutter elements unassigned; it allows extracting overlapping clusters [53]; it generalizes naturally to hypergraph clustering problems, i.e., in the presence of high-order affinities [44], in which case the clustering game is played by more than two players.

Outline The chapter is organized as follows. We provide basic game-theoretic notions and notation in Sect. 8.2. Section 8.3 presents the idea of the clustering game and provides different characterizations thereof. In Sect. 8.4, we describe algorithms

that can be used to find clusters according to the proposed framework. In Sects. 8.5 and 8.6, we present two effective applications of our clustering framework to the problem of matching, which is central to any recognition task where the object to be recognized is naturally divided into several parts, and the problem of surface alignment, which is a fundamental step in the reconstruction of three-dimensional objects.

8.2 Notations and Theoretical Background

According to classical game theory [21], a game of strategy between two players can be formalized as a triplet $\Gamma = (P, S, \pi)$, where $P = \{1, 2\}$ is the set of two “players” (or agents), $S = \{1, \dots, n\}$ is a set of *pure strategies* (or actions) available to each player, and $\pi : S^2 \rightarrow \mathbb{R}$ is a *payoff function*, which assigns a utility to each *strategy profile* $(s_1, s_2) \in S^2$, which is an (ordered) pair of pure strategies played by the different players.¹ The payoff function can also be represented as a 2-dimensional matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ such that $a_{ij} = \pi(i, j)$.

Evolutionary game theory originated in the early 1970s as an attempt to apply the principles and tools of game theory to biological contexts, with a view to model the evolution of animal, as opposed to human, behavior (see the classical work by J. Maynard Smith [35] who pioneered the field). It considers an idealized scenario whereby individuals are repeatedly drawn at random from a large, ideally infinite, population to play a two-player game. In contrast to classical game theory, here players are not supposed to behave rationally or to have complete knowledge of the details of the game. They act instead according to an inherited behavioral pattern, or pure strategy, and it is supposed that some evolutionary selection process operates over time on the distribution of behaviors. Here, and in the sequel, an agent with preassigned strategy $j \in S$ will be called a *j-strategist*. The state of the population at a given time t can be represented as an n -dimensional vector $\mathbf{x}(t)$, where $x_j(t)$ represents the fraction of *j*-strategists in the population at time t . Hence, the initial distribution of preassigned strategies in the population is given by $\mathbf{x}(0)$. The set of all possible states describing a population is given by

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{j \in S} x_j = 1 \text{ and } x_j \geq 0 \text{ for all } j \in S \right\}$$

which is called the *standard simplex*. Points in the standard simplex are also referred to as *mixed strategies* in game theory. As time passes, the distribution of strategies in the population changes under the effect of a selection mechanism which, by analogy with Darwinian process, aims at spreading the fittest strategies in the population

¹We note that although we restrict ourselves to games where all players share the same set of pure strategies and payoff function, in more general settings each agent can well be associated to its own pure strategy set and payoff function.

to the detriment of the weakest ones which, in turn, will be driven to extinction (we postpone the formalization of one such selection mechanism to Sect. 8.4). For notational convenience, we drop the time reference t from a population state and we refer to $\mathbf{x} \in \Delta$ as a population rather than population state. Moreover, we denote by $\sigma(\mathbf{x})$ the *support* of $\mathbf{x} \in \Delta$:

$$\sigma(\mathbf{x}) = \{j \in S : x_j > 0\}$$

which is the set of strategies that are alive in a given population \mathbf{x} .

We will find it useful to define the following function $u : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$:

$$u(\mathbf{y}^{(1)}, \mathbf{y}^{(2)}) = \sum_{(s_1, s_2) \in S^2} \pi(s_1, s_2) \prod_{i \in \{1, 2\}} y_{s_i}^{(i)} = \mathbf{y}^{(1)\top} \mathbf{A} \mathbf{y}^{(2)}. \quad (8.1)$$

We will also write \mathbf{e}^j to indicate the n -vector with $x_j = 1$ and zero elsewhere. Now, it is easy to see that the expected payoff earned by a j -strategist in a population $\mathbf{x} \in \Delta$ is given by

$$u(\mathbf{e}^j, \mathbf{x}) = (\mathbf{A}\mathbf{x})_j = \sum_{s \in S} a_{js, x_s},$$

while the expected payoff over the entire population is given by

$$u(\mathbf{x}, \mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x} = \sum_{j \in S} x_j (\mathbf{A}\mathbf{x})_j.$$

Given a population \mathbf{x} , we denote by $\tau_-(\mathbf{x})$ the set of pure strategies that perform worse than average, i.e.,

$$\tau_-(\mathbf{x}) = \{j \in S : u(\mathbf{e}^j, \mathbf{x}) < u(\mathbf{x}, \mathbf{x})\},$$

by $\tau_+(\mathbf{x})$ the set of strategies performing better than the average, i.e.,

$$\tau_+(\mathbf{x}) = \{j \in S : u(\mathbf{e}^j, \mathbf{x}) > u(\mathbf{x}, \mathbf{x})\},$$

and finally by $\tau_0(\mathbf{x})$ the set of strategies performing as the average, i.e.,

$$\tau_0(\mathbf{x}) = \{j \in S : u(\mathbf{e}^j, \mathbf{x}) = u(\mathbf{x}, \mathbf{x})\}.$$

A fundamental notion in game theory is that of an equilibrium [57]. Intuitively, an evolutionary process reaches an equilibrium $\mathbf{x} \in \Delta$ when every individual in the population obtains the same expected payoff and no strategy can thus prevail upon the other ones. Formally, $\mathbf{x} \in \Delta$ is a *Nash equilibrium* if

$$u(\mathbf{e}^j, \mathbf{x}) \leq u(\mathbf{x}, \mathbf{x}), \quad \text{for all } j \in S. \quad (8.2)$$

In other words, at a Nash equilibrium every agent in the population performs at most as well as the overall population expected payoff. This can also be compactly written

as $\tau_+(\mathbf{x}) \cap S = \emptyset$. A Nash equilibrium $\mathbf{x} \in \Delta$ can be equivalently characterized by the condition that

$$u(\mathbf{y}, \mathbf{x}) \leq u(\mathbf{x}, \mathbf{x}) \quad (8.3)$$

for all $\mathbf{y} \in \Delta$. We say that a Nash equilibrium \mathbf{x} is *strict* if (8.3) holds with strict inequality for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$.

Within a population-based setting, the notion of a Nash equilibrium turns out to be too weak as it lacks stability under small perturbations. This motivated J. Maynard Smith, in his seminal work [35], to introduce a refinement of the Nash equilibrium concept generally known as an Evolutionary Stable Strategy (ESS). Formally, assume that in a population $\mathbf{x} \in \Delta$, a small share ε of mutant agents appear, whose distribution of strategies is $\mathbf{y} \in \Delta$. The resulting post-entry population is then given by $\mathbf{w}_\varepsilon = (1 - \varepsilon)\mathbf{x} + \varepsilon\mathbf{y}$. Biological intuition suggests that evolutionary forces select against mutant individuals if and only if the expected payoff of a mutant agent in the postentry population is lower than that of an individual from the original population, i.e.,

$$u(\mathbf{y}, \mathbf{w}_\varepsilon) < u(\mathbf{x}, \mathbf{w}_\varepsilon). \quad (8.4)$$

Hence, a population $\mathbf{x} \in \Delta$ is said to be *evolutionary stable* if inequality (8.4) holds for any distribution of mutant agents $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$, granted the population share of mutants ε is sufficiently small. It can be shown [57] that \mathbf{x} is an ESS equilibrium if and only if it is a Nash equilibrium and the additional stability property $u(\mathbf{x}, \mathbf{y}) > u(\mathbf{y}, \mathbf{y})$ holds for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$ such that $u(\mathbf{y}, \mathbf{x}) = u(\mathbf{x}, \mathbf{x})$.

8.3 Clustering Games

An instance of the clustering problem can be described by an edge-weighted graph, which is formally defined as a triplet $G = (V, E, \omega)$, where $V = \{1, \dots, n\}$ is a finite set of *vertices*, $E \subseteq V \times V$ is the set of oriented edges and $\omega : E \rightarrow \mathbb{R}$ is a real-valued function which assigns a weight to each edge. Within our clustering framework, the vertices in G correspond to the objects to be clustered, the edges represent neighborhood relationships among objects, and the edge-weights reflect similarity among linked objects. Note that in our framework no assumption is made on the similarity function.

Given a graph $G = (V, E, \omega)$, representing an instance of a clustering problem, we cast it into a two-player *clustering game* $\Gamma = (P, V, \pi)$ where the players' pure strategies correspond to the objects to be clustered and the payoff function π is proportional to the similarity of the objects/strategies $(v_1, v_2) \in V^2$ selected by the players:

$$\pi(v_1, v_2) = \begin{cases} \omega(v_1, v_2) & \text{if } (v_1, v_2) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (8.5)$$

Our clustering game will be played within an evolutionary setting wherein the two players, each of which is assumed to play a pre-assigned strategy, are repeatedly drawn at random from a large population. Here, given a population $\mathbf{x} \in \Delta$, x_j ($j \in V$) represents the fraction of players that is programmed to select j from the objects to be clustered. A dynamic evolutionary selection process, as the one described in Sect. 8.4, will then make the population \mathbf{x} evolve according to a Darwinian survival-of-the-fittest principle in such a way that, eventually, the better-than-average objects will survive and the others will get extinct. It is clear that the whole dynamical process is driven by the payoff function π which, in our case, has been defined in (8.5) precisely to favor the evolution of highly coherent objects. Accordingly, the support $\sigma(\mathbf{x})$ of the converged population \mathbf{x} does represent a cluster, the non-null components of \mathbf{x} providing a measure of the degree of membership of its elements. Indeed, the expected population payoff $u(\mathbf{x}, \mathbf{x})$ can be regarded as a measure of the cluster's internal coherency in terms of the average similarity of the objects forming the cluster, whereas the expected payoff $u(\mathbf{e}^j, \mathbf{x})$ of a player selecting object $j \in V$ in \mathbf{x} measures the average similarity of object j with respect to the cluster.

We claim that, within this setting, the clusters of a clustering problem instance can be characterized in terms of the ESSs of the corresponding (evolutionary) clustering game, thereby justifying the following definition.

Definition 8.1 (ESS-cluster) Given an instance of a clustering problem $G = (V, E, \omega)$, an *ESS-cluster* of G is an ESS of the corresponding clustering game.

For the sake of simplicity, when it will be clear from context, the term ESS-cluster will be used henceforth to refer to either the ESS itself, namely the membership vector $\mathbf{x} \in \Delta$, or to its support $\sigma(\mathbf{x}) = C \subseteq V$.

The motivation behind the above definition resides in the observation that ESS-clusters do incorporate the two basic properties of a cluster, i.e.,

- *Internal coherency*: elements belonging to the cluster should have high mutual similarities;
- *External incoherency*: the overall cluster internal coherency decreases by introducing external elements.

The rest of this section is devoted to provide support to this claim.

8.3.1 A Combinatorial Characterization

In this section, we provide a complete combinatorial characterization of the clusters under our game-theoretic framework, or more generally of evolutionary stable strategies of two-person symmetric games, which we derived from the dominant set framework [38].

Let $S = \{1, \dots\}$ be the set of the objects to be clustered, let A be the objects' similarity matrix and let $C \subseteq S$ be a non-empty subset of objects. The (*average*) *weighted in-degree* of $i \in S$ with respect to C is defined as:

$$\text{awindeg}_C(i) = \frac{1}{|C|} \sum_{j \in C} a_{ij},$$

where $|C|$ denotes the cardinality of C . Moreover, if $j \in C$ we define

$$\phi_C(i, j) = a_{ij} - \text{awindeg}_C(j),$$

which is a measure of the similarity of object i with object j with respect to the average similarity of object j with elements in C . The *weight* of i with respect to C is

$$W_C(i) = \begin{cases} 1 & \text{if } |C| = 1, \\ \sum_{j \in C \setminus \{i\}} \phi_{C \setminus \{i\}}(i, j) W_{C \setminus \{i\}}(j) & \text{otherwise,} \end{cases}$$

while the *total weight* of C is defined as

$$W(C) = \sum_{i \in C} W_C(i).$$

Intuitively, $W_C(i)$ gives us a measure of the support that object i receives from the objects in $C \setminus \{i\}$ relative to the overall mutual similarity of the objects in $C \setminus \{i\}$. Here positive values indicate that i has high similarity to $C \setminus \{i\}$.

A non-empty subset of objects $C \subseteq S$ such that $W(T) > 0$ for any non-empty $T \subseteq C$ is said to be a *dominant set* if:

1. $W_C(i) > 0$, for all $i \in C$,
2. $W_{C \cup \{i\}}(i) \leq 0$, for all $i \notin C$.

The two previous conditions correspond to the two main properties of a cluster: the first regards internal homogeneity, whereas the second regards external heterogeneity. The above definition represents our formalization of the concept of a cluster, when A is the similarity matrix describing the clustering problem.

The *weighted characteristic vector* \mathbf{x}^C of a set $C \subseteq S$ is defined as

$$x_i^C = \begin{cases} \frac{W_C(i)}{W(C)} & \text{if } i \in C, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 8.1 *If $C \subseteq S$ is a dominant set with respect to affinity matrix A , then \mathbf{x}^C is an ESS for a two-player game with payoff matrix A .*

Conversely, if \mathbf{x} is an ESS for a two-person game with payoff matrix A , then $C = \sigma(\mathbf{x})$ is a dominant set with respect to A , provided that $C = \tau_0(\mathbf{x})$.

Proof See [52]. □

This result provides a generalization of the dominant set framework [38] to asymmetric affinities.

8.3.2 A Link to Optimization Theory

If we restrict our attention to symmetric payoff functions, then the notions of Nash equilibrium and ESS have a natural interpretation in terms of optimization theory. Let A be a symmetric payoff matrix and consider the following constrained program, also known as *standard quadratic program* [9]:

$$\begin{aligned} & \text{maximize} && u(\mathbf{x}, \mathbf{x}) = \mathbf{x}^T A \mathbf{x} \\ & \text{subject to} && \mathbf{x} \in \Delta \subset \mathbb{R}^n. \end{aligned} \tag{8.6}$$

A point \mathbf{x} satisfies the Karush–Kuhn–Tucker (KKT) conditions for problem (8.6), i.e., the first-order necessary conditions for local optimality [34], if there exists $n + 1$ real constants (Lagrange multipliers) μ_1, \dots, μ_n and λ , with $\mu_i \geq 0$ for all $i = 1, \dots, n$, such that

$$(A\mathbf{x})_i - \lambda + \mu_i = 0,$$

and $\sum_{i=1}^n x_i \mu_i = 0$. Note that, since both x_i and μ_i are nonnegative for all $i = 1, \dots, n$, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ implies $\mu_i = 0$. Hence, the KKT conditions can be rewritten as

$$u(\mathbf{e}^i, \mathbf{x}) = (A\mathbf{x})_i \begin{cases} = \lambda & \text{if } i \in \sigma(\mathbf{x}), \\ \leq \lambda & \text{otherwise,} \end{cases}$$

for some real constant λ .

It is immediate to see that $\lambda = u(\mathbf{x}, \mathbf{x})$. In fact,

$$u(\mathbf{x}, \mathbf{x}) = \sum_{i \in \sigma(\mathbf{x})} x_i u(\mathbf{e}^i, \mathbf{x}) = \sum_{i \in \sigma(\mathbf{x})} x_i \lambda = \lambda.$$

Therefore, we have that \mathbf{x} satisfies the KKT condition if for all $i = 1, \dots, n$, $u(\mathbf{e}^i, \mathbf{x}) \leq u(\mathbf{x}, \mathbf{x})$, which indeed corresponds to the Nash equilibrium condition. Hence, under symmetric payoff matrices, the Nash condition is equivalent to the necessary condition for local optimality in (8.6). Moreover, as shown in the following theorem, ESS equilibria can be characterized in terms of strict local solutions of (8.6).

Theorem 8.2 *Strict local maximizers of (8.6) are ESS equilibria of a two-player game with payoff matrix A and vice versa.*

Proof See [26]. □

8.3.3 A Link to Graph Theory

Let $G = (V, E)$ be an undirected graph without self-loops, where $V = \{1, 2, \dots, n\}$ is the set of vertices and $E \subseteq V \times V$ the set of edges. We define the *order* of a graph G as the cardinality of V . Two vertices $u, v \in V$ are *adjacent* if $(u, v) \in E$. A subset C of vertices in G is called a *clique* if all its vertices are mutually adjacent. It is a *maximal clique* if it is not a subset of other cliques in G . It is a *maximum clique* if it has maximum cardinality. The cardinality of a maximum clique of G is also called *clique number* and it is denoted by $\omega(G)$. The *adjacency matrix* of G is the $n \times n$ symmetric matrix $A_G = (a_{ij})$, where $a_{ij} = 1$ if $(i, j) \in E$, $a_{ij} = 0$ otherwise.

The adjacency matrix of an undirected graph can be regarded to as the similarity matrix of a clustering problem, and therefore our framework can be used to find the clusters. Due to this link to graph theory, it is interesting to see the interpretation of our game-theoretic notion of cluster in this context.

Consider the following constrained quadratic program:

$$\begin{aligned} & \text{maximize} && f_\alpha(\mathbf{x}) = \mathbf{x}^T (A_G + \alpha I) \mathbf{x} \\ & \text{subject to} && \mathbf{x} \in \Delta \subset \mathbb{R}^n, \end{aligned} \tag{8.7}$$

where n is the order of G , I the identity matrix, α is a real parameter, and where Δ is the standard simplex of the n -dimensional Euclidean space.

In 1965, Motzkin and Straus [36] established a connection between the maximum clique problem and the program in (8.7) with $\alpha = 0$. Specifically, they related the clique number of G to global solutions \mathbf{x}^* of the program through the formula $\omega(G) = (1 - f_0(\mathbf{x}^*))^{-1}$, and showed that a subset of vertices C is a maximum clique of G if and only if its characteristic vector $\mathbf{x}^C \in \Delta$ is a global maximizer of f_0 on Δ .² Pelillo and Jagota [40] extended the Motzkin–Straus theorem by providing a characterization of maximal cliques in terms of local maximizers of f_0 in Δ .

A drawback of the original Motzkin–Straus formulation is the existence of “spurious” solutions, i.e., maximizers of f_0 over Δ that are not in the form of characteristic vectors. This was observed empirically by Pardalos and Phillips [37] and formalized later by Pelillo and Jagota [40]. In principle, spurious solutions represent a problem since, while providing information about the order of the maximum clique, they do not allow us to easily extract its vertices. Fortunately, there is a straightforward solution to this problem which has been introduced by Bomze [8]. He, indeed, suggested to adopt the formulation in (8.7) and basically proved that for $0 < \alpha < 1$ all local maximizer of (8.7) are strict and in one-to-one correspondence with the characteristic vectors of the maximal cliques of G .

There is an interesting relation between our notion of cluster and graph theory that arises if we consider $A_G + \alpha I$ as the similarity matrix. As seen in the previous section, the first order necessary conditions for \mathbf{x} to be a local maximizer of (8.7)

²In the original paper, Motzkin and Straus proved the “only-if” part of this theorem. The converse, however, is a straightforward consequence of their result [40].

coincide with the conditions for \mathbf{x} to be a Nash equilibrium. Hence, local maximizers of (8.7) are indeed Nash equilibria, but the converse does not necessarily hold. On the other hand, we have that \mathbf{x} is an ESS if and only if it is a strict local maximizer of (8.7). Since strict local maximizers are in one-to-one correspondence with the maximal cliques of G , we have that the support of an ESS is indeed a maximal clique. Consequently, there exists a one-to-one relation between maximal cliques of a graph G and ESS-clusters of a clustering game with payoff matrix $A_G + \alpha I$ when $0 < \alpha < 1$ as stated by the following proposition.

Proposition 8.1 *Let $G = (V, E)$ be an undirected graph with adjacency matrix A_G and $0 < \alpha < 1$. A mixed strategy \mathbf{x} is an ESS of a symmetric two-player game with payoff matrix $A_G + \alpha I$ if and only if it is the characteristic vector of a maximal clique of G .*

Proof ESSs of $A_G + \alpha I$ are in one-to-one correspondence with the strict local maximizers of (8.7) [26] and \mathbf{x} is a strict local maximizer of $f_\alpha(\mathbf{x})$ if and only if it is the characteristic vector of a maximal clique of G [8]. Hence, the result follows. \square

Finally, an extension of this result to the case of directed graphs can be found in [52].

8.4 Algorithms

In the previous section, we introduced a game-theoretic notion of cluster, but we only mentioned at the way clustering effectively takes place. Summarizing, the intuition is to let non-rational individuals play the clustering game under an evolutionary setting, until the distribution of strategies reaches an equilibrium, which in turn provides us with a cluster. In order this to work, however, we have to specify some selection mechanisms that effectively drives the population to equilibrium, which, resembling a Darwinian process, spreads the fittest strategies in the population to the detriment of the weakest one, which in turn will be driven to extinction. The section starts introducing the replicator dynamics, i.e., the standard dynamics developed in evolutionary game theory. Afterwards, we present a new class of dynamics that have several desired features and are computationally more appealing than the replicator dynamics.

8.4.1 Replicator Dynamics

In evolutionary game theory, the assumption is made that the game is played over and over, generation after generation, and that the action of natural selection will

result in the evolution of the fittest strategies. A general class of evolution equations is given by the following set of ordinary differential equations [57]:

$$\dot{x}_i = x_i(t)g_i(\mathbf{x}) \quad (8.8)$$

for $i = 1, \dots, n$, where a dot signifies derivative with respect to time and $g = (g_1, \dots, g_n)$ is a function with open domain containing Δ . Here, the function g_i ($i \in S$) specifies the rate at which pure strategy i replicates. It is usually required that the growth function g is *regular* [57], which means that it is Lipschitz continuous and that $g(\mathbf{x})^\top \mathbf{x} = 0$ for all $\mathbf{x} \in \Delta$. The former condition guarantees us that the system of the differential equation (8.8) has a unique solution through any initial population state. The latter condition, instead, ensures that the simplex Δ is invariant under (8.8), namely, any trajectory starting in Δ will remain in Δ .

A point \mathbf{x} is said to be a *stationary* (or equilibrium) point for our dynamical systems, if $\dot{x}_i = 0$ ($i \in S$). A stationary point \mathbf{x} is (Lyapunov) *stable* if for every neighborhood U of \mathbf{x} there exists a neighborhood V of \mathbf{x} such that $\mathbf{x}(0) \in V$ implies $\mathbf{x}(t) \in U$ for all $t \geq 0$. A stationary point is said to be *asymptotically stable* if any trajectory starting in its vicinity will converge to it as $t \rightarrow \infty$.

Payoff-monotonic game dynamics represent a wide class of regular selection dynamics for which useful properties hold. Intuitively, for a payoff-monotonic dynamics the strategies associated to higher payoffs will increase at a higher rate. Formally, a regular selection dynamics (8.8) is said to be *payoff-monotonic* if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \Leftrightarrow \quad u(\mathbf{e}^i, \mathbf{x}) > u(\mathbf{e}^j, \mathbf{x})$$

for all $\mathbf{x} \in \Delta$ and $i, j \in S$.

Although this class contains many different dynamics, it turns out that they share a lot of common properties. To begin, they all have the same set of stationary points. Indeed, $\mathbf{x} \in \Delta$ is a stationary point under any payoff monotonic dynamics if and only if $u(\mathbf{e}^i, \mathbf{x}) = u(\mathbf{x}, \mathbf{x})$ holds for all $i \in \sigma(\mathbf{x})$ [57].

A well-known subclass of payoff-monotonic game dynamics is given by

$$\dot{x}_i = x_i \left(f(u(\mathbf{e}^i, \mathbf{x})) - \sum_{j \in S} x_j f(u(\mathbf{e}^j, \mathbf{x})) \right),$$

where $f(u)$ is an increasing function of u . These models arise in modeling the evolution of behavior by way of imitation processes, where players are occasionally given the opportunity to change their own strategies [57].

When f is the identity function, that is, $f(u) = u$, we obtain the standard continuous-time *replicator equations*,

$$\dot{x}_i = x_i (u(\mathbf{e}^i, \mathbf{x}) - u(\mathbf{x}, \mathbf{x})), \quad (8.9)$$

whose basic idea is that the average rate of increase \dot{x}_i/x_i equals the difference between the average fitness of strategy i and the mean fitness over the entire population.

Another popular model arises when $f(u) = e^{ku}$, where k is a positive constant. As k tends to 0, the orbits of this dynamics approach those of the standard, first-order replicator model, slowed down by the factor k ; moreover, for large values of k , the model approximates the so-called best-reply dynamics [26].

The replicator dynamics, and more in general any payoff monotonic dynamics, have the following properties[26, 57]:

Theorem 8.3 *Under any payoff monotonic dynamics the following hold true:*

- A Nash equilibrium is a stationary point;
- A strict Nash equilibrium is asymptotically stable;
- A stationary point \mathbf{x}^* that is the limit of an interior orbit, i.e., such that $\sigma(\mathbf{x}(t)) = S$ for all $t \geq 0$ and $\lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}^*$, is a Nash equilibrium;
- A stable stationary point is a Nash equilibrium;
- An ESS is asymptotically stable.

In general, the converses of the implications in Theorem 8.3 do not hold.

Furthermore, if we restrict our focus to symmetric payoff matrices, i.e., $A = A^\top$, then stronger properties hold, as stated in the following theorem.

Theorem 8.4 *If $A = A^\top$ then the following hold:*

- $u(\mathbf{x}, \mathbf{x})$ is strictly increasing along any non-constant trajectory of (8.9). In other words, for all $t \geq 0$ we have $\dot{u}(\mathbf{x}, \mathbf{x}) > 0$, unless \mathbf{x} is a stationary point. Furthermore, any such trajectory converges to a (unique) stationary point;
- \mathbf{x} is asymptotically stable if and only if \mathbf{x} is an ESS.

In order to implement the continuous-time replicator dynamics, one can resort to some iterative method like, e.g., the Runge–Kutta method, to find an approximate solution to the ordinary differential equations. Alternatively, one can adopt the discrete-time counterpart of (8.9), known as discrete-time replicator dynamics, which (assuming non-negative payoffs) is given by

$$x_i(t+1) = x_i(t) \frac{u(\mathbf{e}^i, \mathbf{x})}{u(\mathbf{x}, \mathbf{x})},$$

for $i \in S$. This equation is known to possess many of the dynamical properties of the continuous-time dynamics [57].

8.4.2 Infection and Immunization Dynamics

In order to overcome some computational problems afflicting standard evolutionary dynamics, we introduce a new class of evolutionary dynamics, inspired by infection and immunization processes.

Let $\mathbf{x} \in \Delta$ be the *incumbent* population state, \mathbf{y} be the *mutant* population invading \mathbf{x} and let $\mathbf{z} = (1 - \varepsilon)\mathbf{x} + \varepsilon\mathbf{y}$ be the population state obtained by injecting into \mathbf{x} a small share of \mathbf{y} -strategists. Then the *score function* of \mathbf{y} versus \mathbf{x} (introduced in [10]) is given by

$$h_{\mathbf{x}}(\mathbf{y}, \varepsilon) = u(\mathbf{y}, \mathbf{z}) - u(\mathbf{x}, \mathbf{z}) = \varepsilon u(\mathbf{y} - \mathbf{x}, \mathbf{y} - \mathbf{x}) + u(\mathbf{y} - \mathbf{x}, \mathbf{x}).$$

Following [11], we define the (*neutral*) *invasion barrier* $b_{\mathbf{x}}(\mathbf{y})$ of $\mathbf{x} \in \Delta$ against any mutant strategy \mathbf{y} as the largest population share $\varepsilon_{\mathbf{y}}$ of \mathbf{y} -strategists such that for all smaller positive population shares ε , \mathbf{x} earns a higher or equal payoff than \mathbf{y} in the post-entry population \mathbf{z} . Formally,

$$b_{\mathbf{x}}(\mathbf{y}) = \inf\{\varepsilon \in (0, 1) : h_{\mathbf{x}}(\mathbf{y}, \varepsilon) > 0\} \cup \{1\}.$$

Given populations $\mathbf{x}, \mathbf{y} \in \Delta$, we say that \mathbf{x} is *immune* against \mathbf{y} if $b_{\mathbf{x}}(\mathbf{y}) > 0$. Trivially, a population is always immune against itself. Note that \mathbf{x} is immune against \mathbf{y} if and only if either $u(\mathbf{y} - \mathbf{x}, \mathbf{x}) < 0$ or $u(\mathbf{y} - \mathbf{x}, \mathbf{x}) = 0$ and $u(\mathbf{y} - \mathbf{x}, \mathbf{y} - \mathbf{x}) \leq 0$. If $u(\mathbf{y} - \mathbf{x}, \mathbf{x}) > 0$, we say that \mathbf{y} is *infective* for \mathbf{x} . Hence, the set of infective strategies for \mathbf{x} is given by

$$\mathcal{Y}(\mathbf{x}) = \{\mathbf{y} \in \Delta : u(\mathbf{y} - \mathbf{x}, \mathbf{x}) > 0\}.$$

Consider $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$; clearly, this implies $b_{\mathbf{x}}(\mathbf{y}) = 0$. If we allow for an invasion of a share ε of \mathbf{y} -strategists as long as the score function of \mathbf{y} versus \mathbf{x} is positive, at the end we will have a share of $\delta_{\mathbf{y}}(\mathbf{x})$ mutants in the postentry population, where

$$\delta_{\mathbf{y}}(\mathbf{x}) = \inf\{\varepsilon \in (0, 1) : h_{\mathbf{x}}(\mathbf{y}, \varepsilon) \leq 0\} \cup \{1\}.$$

Note that if \mathbf{y} is infective for \mathbf{x} , then $\delta_{\mathbf{y}}(\mathbf{x}) > 0$, whereas if \mathbf{x} is immune against \mathbf{y} , then $\delta_{\mathbf{y}}(\mathbf{x}) = 0$. Further note that all the above concepts can be straightforwardly extended to contests with more than two participants and/or correlated individual behavior, where the score functions may be nonlinear in ε ; see, e.g., [11] and references therein. In our two-person context, score functions are (affine-)linear, so that there is a simpler expression for $\delta_{\mathbf{y}}(\mathbf{x})$:

$$\delta_{\mathbf{y}}(\mathbf{x}) = \begin{cases} \min\left\{\frac{u(\mathbf{x}-\mathbf{y}, \mathbf{x})}{u(\mathbf{y}-\mathbf{x}, \mathbf{y}-\mathbf{x})}, 1\right\} & \text{if } u(\mathbf{y} - \mathbf{x}, \mathbf{y} - \mathbf{x}) < 0, \\ 1 & \text{otherwise.} \end{cases} \quad (8.10)$$

It can be proven [42] that if we allow a population \mathbf{x} to be invaded by an infective strategy \mathbf{y} , and the extent of this infection is $\delta_{\mathbf{y}}(\mathbf{x})$, then the postentry population will become immune against \mathbf{y} . In formal terms, given $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ and $\mathbf{z} = [1 - \delta_{\mathbf{y}}(\mathbf{x})]\mathbf{x} + \delta_{\mathbf{y}}(\mathbf{x})\mathbf{y}$, we have that \mathbf{z} is immune against \mathbf{y} . The core idea of our method consists in selecting a strategy \mathbf{y} which is infective for the current population \mathbf{x} . By allowing for invasion as shown before, we obtain a new population \mathbf{z} which is immune to \mathbf{y} . This idea suggests the following class of new dynamics which for evident reasons is called *Infection and Immunization Dynamics* (InImDyn):

$$\mathbf{x}(t + 1) = \delta_{\mathcal{S}(\mathbf{x})}(\mathbf{x})[\mathcal{S}(\mathbf{x}) - \mathbf{x}] + \mathbf{x}, \quad (8.11)$$

where \mathbf{x} should be regarded to as $\mathbf{x}(t)$ and $\mathcal{S} : \Delta \rightarrow \Delta$ is a *strategy selection function*, which returns an infective strategy for \mathbf{x} if it exists, or \mathbf{x} otherwise:

$$\mathcal{S}(\mathbf{x}) = \begin{cases} \mathbf{y} & \text{for some } \mathbf{y} \in \Upsilon(\mathbf{x}) \text{ if } \Upsilon(\mathbf{x}) \neq \emptyset, \\ \mathbf{x} & \text{otherwise.} \end{cases} \quad (8.12)$$

By reiterating this process of *immunization*, we aim at reaching a population state \mathbf{x} that cannot be infected by any other strategy. If this is the case then \mathbf{x} is a fixed point under dynamics (8.11), but also a Nash strategy:

Theorem 8.5 *Let $\mathbf{x} \in \Delta$ be a strategy. Then the following statements are equivalent:*

- (a) $\Upsilon(\mathbf{x}) = \emptyset$, i.e., there is no infective strategy for \mathbf{x} ;
- (b) \mathbf{x} is a Nash strategy;
- (c) \mathbf{x} is a fixed point under dynamics (8.11).

Proof See [42]. □

The following result shows that the average payoff is strictly increasing along any non-constant trajectory of the dynamics (8.11), provided that the payoff matrix is symmetric.

Theorem 8.6 *Let $\{\mathbf{x}(t)\}_{t \geq 0}$ be a trajectory of (8.11). Then for all $t \geq 0$,*

$$u(\mathbf{x}(t+1), \mathbf{x}(t+1)) \geq u(\mathbf{x}(t), \mathbf{x}(t)),$$

with equality if and only if $\mathbf{x}(t) = \mathbf{x}(t+1)$, provided that the payoff matrix is symmetric.

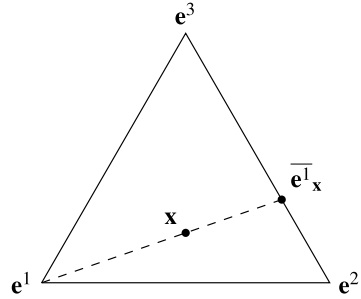
Proof See [42]. □

Theorem 8.6 shows that by running INIMDYN, under a symmetric payoff function, we strictly increase the population payoff unless we are at a fixed point, i.e., have already reached Nash equilibrium. This, of course, holds for any selection function $\mathcal{S}(\mathbf{x})$ satisfying (8.12). However, the way we choose $\mathcal{S}(\mathbf{x})$ may affect the efficiency of the dynamics. The next section introduces a particular selection function that leads to a well-performing dynamics for our purposes.

Depending on how we choose the function $\mathcal{S}(\mathbf{x})$ in (8.11), we may obtain different dynamics. One in particular, which is simple and leads to nice properties, consists in allowing only infective pure strategies or their respective co-strategies. This way, our equilibrium selection process closely resembles a vertex-pivoting method, as opposed to interior-point approaches like replicator dynamics or best-response dynamics [26].

If \mathbf{x} is not fixed under (8.11), i.e., is not a Nash strategy, straightforward intuition renders selection of an infective strategy in a way easier than it could seem at first

Fig. 8.1 Example of a co-strategy of the pure strategy e^1 with respect to \mathbf{x}



glance. Let \mathbf{x} be the current population and let \mathbf{y} be a strategy. The *co-strategy* of \mathbf{y} with respect to \mathbf{x} is given by

$$\bar{\mathbf{y}}_{\mathbf{x}} = (1 - \bar{\varepsilon})\mathbf{x} + \bar{\varepsilon}\mathbf{y},$$

where

$$\bar{\varepsilon} = \min\{\varepsilon \in \mathbb{R} : (1 - \varepsilon)\mathbf{x} + \varepsilon\mathbf{y} \in \Delta\} \leq 0.$$

For any strategy \mathbf{y} , if both $u(\mathbf{y} - \mathbf{x}, \mathbf{x})$ and $\bar{\varepsilon}$ are nonzero, then either $\mathbf{y} \in \Upsilon(\mathbf{x})$ or $\bar{\mathbf{y}}_{\mathbf{x}} \in \Upsilon(\mathbf{x})$ in an exclusive sense.

In Fig. 8.1, we can see that the co-strategy of e^i with respect to \mathbf{x} is the intersection between the simplex boundary and the half line originated in e^i and passing through \mathbf{x} . In this case, $\bar{\varepsilon} = x_i / (x_i - 1)$.

Consider the strategy selection function $\mathcal{S}_{\text{Pure}}(\mathbf{x})$, which finds a pure strategy i maximizing $|u(e^i - \mathbf{x}, \mathbf{x})|$, and returns e^i , $\bar{e}^i_{\mathbf{x}}$ or \mathbf{x} according to whether $i \in \tau_+(\mathbf{x})$, $i \in \tau_-(\mathbf{x}) \cap \sigma(\mathbf{x})$ or $i \in \tau_0(\mathbf{x})$: Let $\mathcal{M}(\mathbf{x})$ be a (randomly or otherwise selected) pure strategy such that

$$\mathcal{M}(\mathbf{x}) \in \arg \max\{u(e^i - \mathbf{x}, \mathbf{x}) : i \in \tau_+(\mathbf{x})\} \cup \{u(\mathbf{x} - e^i, \mathbf{x}) : i \in \tau_-(\mathbf{x}) \cap \sigma(\mathbf{x})\}.$$

Then $\mathcal{S}_{\text{Pure}}(\mathbf{x})$ can be written as

$$\mathcal{S}_{\text{Pure}}(\mathbf{x}) = \begin{cases} e^i & \text{if } i = \mathcal{M}(\mathbf{x}) \in \tau_+(\mathbf{x}), \\ \bar{e}^i_{\mathbf{x}} & \text{if } i = \mathcal{M}(\mathbf{x}) \in \tau_-(\mathbf{x}) \cap \sigma(\mathbf{x}), \\ \mathbf{x} & \text{otherwise.} \end{cases}$$

For obvious reasons, we refer to InImDyn with selection function $\mathcal{S}_{\text{Pure}}(\mathbf{x})$ as *Pure InImDyn*.

Note that the search space for an infective strategy is reduced from Δ to a finite set. Therefore, it is not obvious that $\mathcal{S}_{\text{Pure}}(\mathbf{x})$ is a well-defined selection function, i.e., it satisfies (8.12). However, one can prove [42] that there exists an infective strategy for \mathbf{x} if and only if $\mathcal{S}_{\text{Pure}}(\mathbf{x})$ is infective for \mathbf{x} .

Another property that holds for our new dynamics, which is shared also by the replicator dynamics, is the characterization of ESS equilibria in terms of asymptotically stable points of the dynamics under symmetric payoff matrices.

Algorithm 1: FindEquilibrium(A, \mathbf{x}, τ)

Require: $n \times n$ payoff matrix A , $\mathbf{x} \in \Delta$ and tolerance τ .

while $\varepsilon(\mathbf{x}) > \tau$ **do**

$\mathbf{y} \leftarrow \mathcal{S}_{\text{Pure}}(\mathbf{x})$

$\delta \leftarrow 1$

if $\pi(\mathbf{y} - \mathbf{x}) < 0$ **then**

$\delta \leftarrow \min[\frac{\pi(\mathbf{x} - \mathbf{y}|\mathbf{x})}{\pi(\mathbf{y} - \mathbf{x})}, 1]$

end if

$\mathbf{x} \leftarrow \delta(\mathbf{y} - \mathbf{x}) + \mathbf{x}$

end while

return \mathbf{x}

Theorem 8.7 *A state \mathbf{x} is asymptotically stable for INIMDYN with $\mathcal{S}_{\text{Pure}}$ as strategy selection function if and only if \mathbf{x} is an ESS, provided that the payoff matrix is symmetric.*

Proof See [42]. □

This selection function exhibits the nice property of rendering the complexity per iteration of our new dynamics linear in both space and time, as opposed to the replicator dynamics, which have quadratic space/time complexity per iteration.

Theorem 8.8 *Given the iterate $\mathbf{x}^{(t)}$ and its linear transformations $A\mathbf{x}(t)$ and $A^\top \mathbf{x}(t)$, both space and time requirement of one iteration step is linear in n , the number of objects.*

Proof See [45]. □

The only step of quadratic complexity is the first one, where we need to compute $A\mathbf{x}(0)$ and $A^\top \mathbf{x}(0)$. Even this can be reduced to linear complexity, if we start from a pure strategy \mathbf{e}^i , in which case we have $A\mathbf{x}(0) = A_i$ and $A^\top \mathbf{x}(0) = (A^\top)_i$. Note that the latter is impossible, e.g., for the replicator dynamics.

The algorithmic procedure for finding an equilibrium using INIMDYN with $\mathcal{S}_{\text{Pure}}$ is summarized in Algorithm 1. Note that as stopping criterion we compute the following quantity:

$$\varepsilon(\mathbf{x}) = \sum_i \min\{x_i, \pi(\mathbf{x} - \mathbf{e}^i|\mathbf{x})\}^2 < \tau, \quad (8.13)$$

which measures the degree of violation of the Nash conditions. Indeed, $\varepsilon(\mathbf{x}) = 0$ if and only if \mathbf{x} is a Nash equilibrium.

8.5 Game-Theoretic Matching

The problem of finding correspondences within a set of elements, or features, is central to any recognition task where the object to be recognized is naturally divided into several parts. In this context, graph-based representations have been used with considerable success due to their ability to capture concisely the relational arrangement of object primitives, in a manner which can be invariant to changes in object viewpoint. However, applications in which estimating a set of correspondences is a central task toward the solution range from object recognition, to 3D registration, to feature tracking, to stereo reconstruction [7, 30, 33]. Several matching algorithms have been proposed in the literature. Some can just be classified as ad hoc solutions to specific problems, but the vast majority cast the problem into an energy minimization framework and extract approximate optimizers of an objective function within a set of feasible correspondences. In general, the overall goal is to maximize the global or local coherence of the matched pairs with respect to some compatibility. In most cases, the objective function can be written as a monotonic transformation of the sum of pairwise interactions between matching hypotheses. This can be either the similarity between matched features, as in the graph-matching case [4, 19, 55], and often the set of feasible correspondences can be defined using only unary and binary relations. For instance, it is possible to guarantee a global one-to-one match and structural coherence using the association graph technique described by Barrow and Burstall [6]. Also adjacency and hierarchical constraints can be enforced on a local pairwise basis, as shown by the many techniques that cast the matching problem to an equivalent clique search in an auxiliary association graph [39, 41, 51]. Formulations that satisfy these conditions range from bipartite matching, to sub-graph isomorphism, to quadratic assignment, to edit-distance, and include a dual form of parameter estimation approaches such as Hough transform and RANSAC.

The previous sections introduced a novel game-theoretic clustering approach. In this section, we will build from that framework to introduce a matching approach based on the game-theoretic selection of correspondences between features to be matched. The first part will be devoted to the introduction of the novel selection process, while the second and third part will show applications of this frameworks to two important computer vision tasks.

We present a game-theoretic approach to correspondence estimation derived from the clustering approach presented in the previous section. The proposed approach is quite general since it can be applied to any formulation where both the objective function and the feasible set can be defined in terms of unary and pairwise interactions. The main idea is to model the set of possible correspondences as a set of game strategies. Specifically, we formulate the matching problem as a non-cooperative game where the potential associations between the items to be matched correspond to strategies, while payoffs reflect the degree of compatibility between competing hypotheses. A distinguishing feature of the proposed framework is that it allows one to naturally deal with general many-to-many matching problems even in the presence of asymmetric compatibilities.

8.5.1 Matching as a Non-cooperative Game

Before going into the details of the proposed framework, we need to introduce some notations and definitions that will be used throughout. Let O_1 and O_2 be the two sets of features that we want to match, we define the set of *feasible associations* $\mathbb{A} \subseteq O_1 \times O_2$ the set of relations between O_1 and O_2 that satisfy the unary constraints. Hence, each feasible association represents a possible matching hypothesis. We assume that we can compute a set of *pairwise compatibilities* $C : \mathbb{A} \times \mathbb{A} \rightarrow \mathbb{R}^+$ that measure the support that one association gives to the other. Here, the self compatibilities, i.e., the compatibilities that an association gives to itself, are assumed to be zero.

In this formulation, a *submatch* (or simply a *match*) is intuitively a set of associations, which satisfies the pairwise feasibility constraints, and two additional criteria: *high internal compatibility*, i.e., the associations belonging to the match are mutually highly compatible, and *low external compatibility*, i.e., associations outside the match are scarcely compatible with those inside. This definition of match allows us to abstract from the specific problem, since domain-specific information is confined to the definition of the compatibility function. Further, we are able to deal with many-to-many, one-to-many, many-to-one, and one-to-one relations in a uniform way, as we do not impose restriction on the way the associations are selected, but incorporate the constraints with the compatibilities.

The proposed approach generalizes the association graph technique described by Barrow and Burstall [6] to a context where structural constraints are continuous. Further, the approach can be seen as a proper generalization of [39] since, in case of symmetric 0,1 supports, the solutions of the ESSs maximize the same objective function.

We define a *matching game* as a clustering game over the associations. Assume that we have two sets of objects O_1 and O_2 , and a compatibility function C . Let $O = \{1, \dots, n\}$ be the enumeration of the set of associations \mathcal{A} , where $n = |\mathcal{A}|$. In the matching game, the set of feasible correspondences O forms the set of *pure strategies* (in the language of game-theory) available to the players and $A = (a_{ij})$ is an $n \times n$ payoff (or utility) matrix [56], where c_{ij} is the payoff that a player gains when playing the strategy i against an opponent playing strategy j . Within our matching setting, Nash equilibria are good candidates for a match, as they satisfy both the internal and external compatibility criteria. In fact, any association $i \in \sigma(\mathbf{x})$ of a Nash equilibrium \mathbf{x} receives from \mathbf{x} the same expected payoff $(A\mathbf{x})_i = \mathbf{x}^T A\mathbf{x}$, while associations not in $\sigma(\mathbf{x})$ receive a lower or equal support from associations of the match. Note, however, that the external criterion is not strict: there could exist associations not in $\sigma(\mathbf{x})$ that earn a payoff equal to $\mathbf{x}^T A\mathbf{x}$ like associations in the group, which may lead to a non-isolated Nash equilibrium and, thus, to an ambiguous match. Therefore, here we undertake an evolutionary game-theoretic analysis of the possible strategies available to each player.

8.5.1.1 Enforcing Hard Constraints

A main characteristic of the proposed approach is that associations pairs that have zero compatibility cannot be in the same selected submatch. This means that pairwise constraints can be enforced by forcing to zero the compatibility between associations that do not satisfy the constraints.

Theorem 8.9 *Consider a matching-game with compatibilities $A = (a_{ij})$ with $a_{ij} \geq 0$ and $a_{ii} = 0$. If $\mathbf{x} \in \Delta$ is an ESS then $a_{ij} > 0$ for all $i, j \in \sigma(\mathbf{x})$.*

For a proof see [3].

Theorem 8.9 shows that if we set a non-positive compatibility between two associations, then there exists no match containing them. This provides a way for expressing hard constraints in our matching framework such as one-to-one or one-to-many correspondences.

8.5.2 Point-Pattern Matching

In this set of experiments, our goal is to test the ability of the proposed framework to match corresponding features points between two instances of the same image with modified scale and orientation. The feature points are extracted from each image with the SIFT algorithm [33]. SIFT features are known to be highly repeatable under a large class of affine transformations and are very resilient to splitting or joining. Under these conditions, we need a very selective matcher which enforces a common global transformation to all the matched features. In [33], Lowe gauges the coherence of the transformation using RANSAC. This, however, requires a global threshold for the consensus, which limits the precision of the estimation.

The experiments were performed on the AloI database [22]. For each run we selected 20 images and randomly deformed them with an affine transformation with a scale variation between 0.5 and 2 and a rotation between 0.5 and 2.0 radians. We extracted the SIFT features from the original and transformed image and picked as candidate associations all the pairs with sufficiently similar descriptors. Each candidate association represents a single transformation and supports only associations with similar transformations. To measure the support between two associations, we project the first point of one association with the transformation of the other association. Then we measure the distance between the transformed point and the corresponding point in the first association. We repeat the operation reversing the role of the two associations obtaining the two distances d_1 and d_2 . The support is, then, $e^{-\max(d_1, d_2)}$. Once the best match is extracted, we have two alternatives to compute the final transformation: the first is an unweighted approach where we compute a simple average of the transformation parameters related to the associations in the match. The second approach weighs the transformation parameters with the proportion of the population playing the related strategy at equilibrium.

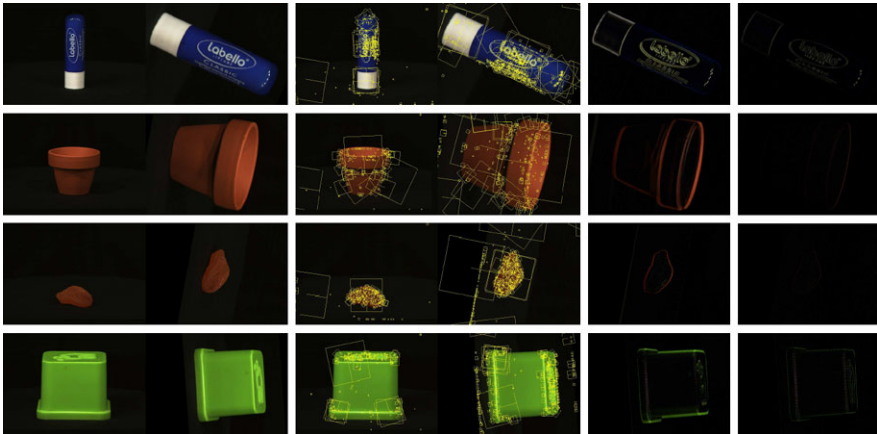


Fig. 8.2 Point pattern matching: the *first two columns* show the original images, the *third and fourth columns* show the extracted features, and the *fourth and fifth* show the allineation error using the transforms estimated using RANSAC (*fifth*) and our approach (*sixth*)

We compare our approach with RANSAC, where we determine the associations to agree within tolerance if $\max(d_1, d_2) < 5$ pixels. the value of 5 pixels was experimentally determined to be the one which gave the best results. Note that this threshold on the error limits the accuracy of RANSAC, while our approach, being parameter-less, does not suffer from this drawback.

Figure 8.2 shows the original images (first two columns), the extracted features (third and fourth columns), and the transformation error obtained using the two approaches (last two columns). The error is the difference between the original image transformed with the estimated transformation and the second image. The fifth column shows the error obtained using the transformation estimated with RANSAC, while the sixth column shows the difference using the transformation estimated using the weighted version of our approach. As can be seen our approach estimates the transformation with higher accuracy than RANSAC. So much so that the difference images are almost completely black. This is mainly due to the lack of a lower bound on the precision of the transformation, which for RANSAC is enforced by the consensus threshold.

Figure 8.3 plots the error in the estimation of translation, scale and rotation as we increase the variations in scale and orientation. The average and standard deviations are computed over 140 images. As can be seen, the weighted and unweighted versions of our approach have similar performance, with the weighted version exhibiting slightly lower error. On the other hand RANSAC show errors an order of magnitude larger in all conditions.

In an attempt to quantify the sensitivity of the approach to noise, we added an increasing amount of Gaussian noise to the rotated and scaled images before we computed the SIFT features. This introduces an increasing number of outliers as well as missing feature points. Figure 8.4 plots the Frobenius norm of the difference between the ground truth and the estimated transformation matrices as the standard

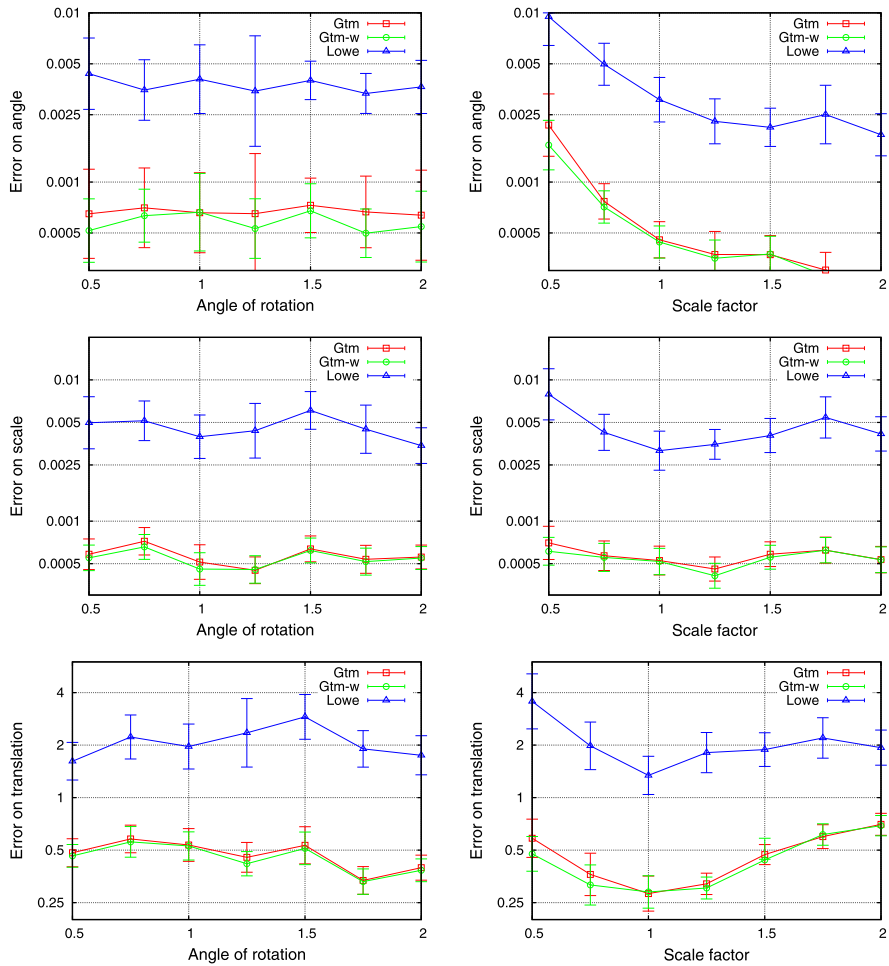
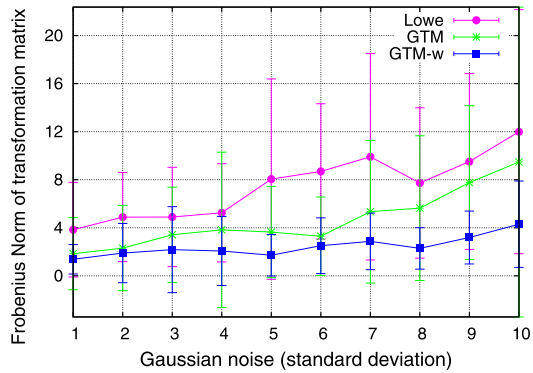


Fig. 8.3 Point pattern matching: error in the estimation of translation, scale and rotation as we increase the variations in scale and orientation. The plots in the first column show the error in rotation angle, scale and translation as a function of the rotation angle. The plots in the second column show the errors as a function of the scale factor

deviation of the Gaussian noise increases. For each noise level we selected 20 images and randomly deformed them with an affine transformation with a scale variation between 0.5 and 2 and a rotation between 0.5 and 2.0 radians. From the plot we can see that our approach maintains a much lower error as compared to RANSAC even at high noise levels. Further, we can see that, while the rate with which the error increases with noise is similar for RANSAC and the unweighted version of our approach, the weighted version appears to provide much lower error even with a high level of noise.

Fig. 8.4 Point pattern matching: sensitivity to noise. The *plot* displays the Frobenius norm of the difference between exact and estimated transformation errors under an increasing amount of Gaussian noise



8.6 Game-Theoretic Surface Alignment

Surface registration is a fundamental step in the reconstruction of three-dimensional objects. This is typically a two step process where an initial coarse motion estimation is followed by a refinement.

Coarse registration techniques can be roughly organized into three main classes: global methods, feature-based methods and technique based on RANSAC [20] or PROSAC [14] schemes. Global methods, such as PCA [15] or Algebraic Surface Model [50], exploit some global property of the surface and thus are very sensitive to occlusion. Feature-based approaches aim at the localization and matching of interesting points on the surfaces. They are more precise and can align surfaces that exhibit only partial overlap. Nevertheless, the unavoidable localization error of the feature points prevents them from obtaining accuracies on par with fine registration methods.

A completely different coarse registration approach is the one taken by RANSAC-based techniques. DARCES [13] is based on the random extraction of sets of mates from the surfaces and their validation based on the accuracy of the estimated transformation. The more recent Four Points Congruent Sets method [2] follows a similar route, but filters the data to reduce noise and performs early check in order to reduce the number of trials.

A recent and extensive review of many different methods can be found in [48].

In this section, we present a novel technique that allows obtaining a fine surface registration in a single step, without the need of an initial motion estimation. The main idea of our approach is to cast the selection of correspondences between points on the surfaces in a game-theoretic framework. This process yields a very robust inlier selection scheme that does not depend on any particular technique for selecting the initial strategies as it relies only on the global geometric compatibility between correspondences. This context diverges from the general matching scheme presented in the previous section in that only a few correspondences are sought. In fact, contrary to the tradition of graph matching, inlier selection processes are tuned to very low false positive correspondences, admitting in converse a large amount of false negatives.

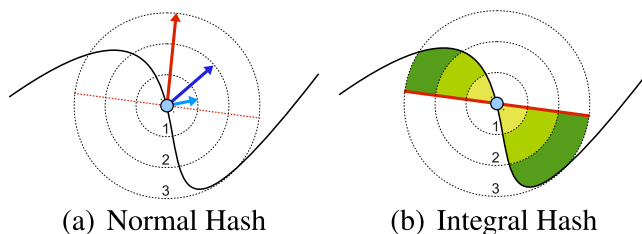


Fig. 8.5 Example of the two basic Surface Hashes proposed

In principle, by adopting our matching approach, all the points from both surfaces to registered could be used to build the matching strategies; in practice, however, this would lead to a very big set of candidates with a huge portion of outliers. We solve the problem by adopting very loose yet repeatable descriptors, and by adopting a game-theoretic approach to select only the distinctive points. In the remaining of this section, we will introduce the point selection process, then the matching process used to perform surface alignment and finally we will experimentally characterize its performance with respect to the state-of-the-art.

8.6.1 Interest Point Selection

Given the large number of points contained in typical 3D objects, it is not practical for any matching algorithm to deal with all of them. In addition, the isolation of a relatively small number of interest points can enhance dramatically the ability of the matcher to avoid false correspondences. We do this using a novel set of robust descriptors and a game-theoretic feature-selection approach. The *Normal Hash* (see Fig. 8.5(a)) is obtained by setting a reference on the average surface normal over a patch that extends to the largest scale (red arrow in figure) and then, for each smaller scale, calculating the dot product between the reference and the average normal over the reduced patches (blue arrows in figure). The rationale behind this measure lies in the observation that at the largest scale the average normal is more stable with respect to noise and that the dot product offers a concise representation of the relation between the vectors obtained at various scales. The *Integral Hash* (see Fig. 8.5(b)) is similar in spirit to the Normal Hash. In this case, we search for the best fitting plane (in the least squares sense) with respect to the surface patch associated to the largest scale. Then we calculate the volume enclosed between the surface and such a plane. In practice, it is not necessary to evaluate this volume accurately: even naive approximations, such as the sum of the distances of the surface points from the plane, have been shown empirically to provide a reasonable approximation. Note that Normal Hashes evaluated over n scales yield descriptor vectors of length $n - 1$ (since the larger scale is used only to calculate the reference normal), while Integral Hashes provide n -dimensional vectors. In Fig. 8.6, a Normal Hash of dimension 3 (respectively from (a) to (c)) evaluated over 4 scales is shown. Note that the descriptor is

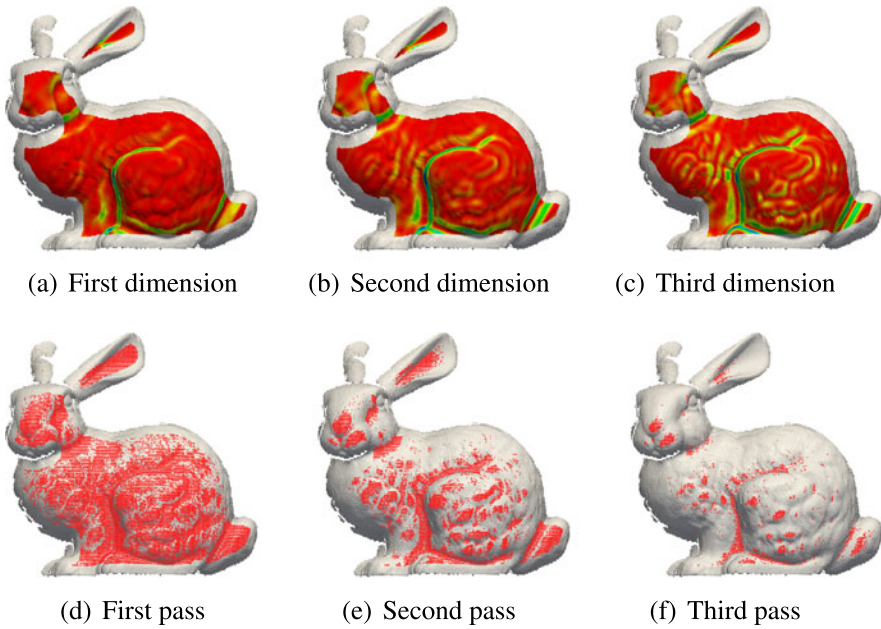


Fig. 8.6 Example of a 3-dimensional Normal Hash and the related detection process

not defined at the points for which the larger support is not fully contained in the surface, i.e., points close to the surface boundary.

In order to obtain discriminant descriptors, we screen out features exhibiting descriptors that are too common over the surface. This is in essence an anomaly detection problem and it is done eliminating the common strategy detected through a clustering game where the strategy set S corresponds to the set of all the surface points and the payoff matrix is defined by

$$\pi_{ij} = e^{-\alpha|d_i - d_j|}, \quad (8.14)$$

where d_i and d_j are the descriptor vectors associated to surface point i and j , and α is a parameter that controls the level of selectivity. We can initialize the set of retained features to the whole surface and run a sequence of Matching Games, eliminating the extracted clusters, until the desired number of points are left. At this point, the remaining features are those characterized by less-common descriptors which are more likely to represent good cues for the matching. It should be noted that by choosing large values for α the payoff function decreases more rapidly with the growth of the distance between the Surface Hashes, thus the Matching Game becomes more selective and fewer points survive. In the end, this results in a blander decimation and thus in a larger ratio of retained interest points. By converse, a small value for α leads to a more greedy filtering and thus to a more selective interest point detector. In Fig. 8.6 (from (d) to (f)), we show three steps of the evolutive interest point selection with respect to the 3-dimensional Normal Hash shown

from (a) to (c). In Fig. 8.6(d), we see that after a single pass of the Matching Game most of the surface points are still considered interesting, while after respectively two and three passes only very distinctive points (belonging to areas with less common curvature profile) are left.

8.6.2 Isometry-Enforcing Game

We will refer to the points belonging to the first surface with the term *model points*, while we will use the term *data points* with respect to the second surface. This distinction is captious since there is no actual difference in role between the two surfaces; however, it is consistent with the current registration literature and helps in defining an order within matches.

Given the set of all model points M and the set of all data points D , we need to construct a set of *matching strategies* $S \subset M \times D$ constructed on the selected interest points. To this end, we perform a discriminative point selection from the model surface, and from this we create the set S by selecting the k most similar points from the whole data model D , where the similarity is gauged through the Euclidean distance of the descriptors. There is, thus, an asymmetry in the role of the surfaces, where only the model M is sub-sampled through the discriminative point selection process, and then it drives the creation of the strategy S . When not otherwise stated, in our experiments we set k to be equal to 5. Limiting the number of correspondences per source feature to a constant value, we limit the growth of the number of strategies to be linear with the number of model points selected.

Since the set of strategies S is built by proposing several attainable matches for each considered model point, while the correct match is not guaranteed to be within the best k selected matches, it is obvious that the number of outliers in S will be far superior to the number of correct correspondences. In order to extract this minority of correct matches buried into S , our framework must exploit the consistency of any pair of those strategies with respect to some property.

In order to define a suitable payoff function, we need to assign to each pair of matching strategies a payoff that is inversely proportional to a measure of violation of the rigid-transformation constraint. This violation can be expressed in several ways, but since all the rigid transformations preserve Euclidean distances, we choose this property to express the coherence between matching strategies. Clearly, this isometry constraint is looser than the rigid-transformation constraint as it cannot prevent specular flips of the surfaces, but the global consistency provided by the game-theoretic framework ensures that only rigid alignments will prevail.

Definition 8.2 Given a function $\pi : S \times S \rightarrow \mathbb{R}^+$, we call it an *isometry-enforcing payoff function* if for any $((a_1, a_2), (b_1, b_2))$ and $((c_1, c_2), (d_1, d_2)) \in S \times S$ we have that $\| |a_1 - b_1| - |a_2 - b_2| \| > \| |c_1 - d_1| - |c_2 - d_2| \|$ implies $\pi((a_1, a_2), (b_1, b_2)) < \pi((c_1, c_2), (d_1, d_2))$.

An isometry-enforcing payoff function is a function that is monotonically decreasing with the absolute difference of the Euclidean distances between respective model and data points of the matching strategies compared. In other words, given two matching strategies, their payoff should be high if the distance between the model points is equal to the distance between the data points, and it should decrease as the difference between such distances increases.

Given a set of matching strategies S and an enumeration $O = \{1, \dots, |S|\}$ over it, an *isometry-enforcing game* is a clustering game where the population is defined as a vector $\mathbf{x} \in \Delta^{|S|}$ and the payoff matrix $A = (a_{ij})$ is defined as $a_{ij} = \pi(s_i, s_j)$, where $s_i, s_j \in S$ are enumerated by O and π is a symmetric one-to-one isometry-enforcing payoff function. Intuitively, x_i accounts for the percentage of the population that plays the i th matching strategy.

In theory, any rigidity-enforcing payoff function can be used to perform surface registration. Throughout the experimental section, we adopted

$$\pi((a_1, b_1), (a_2, b_2)) = \left(\frac{\min(|a_1 - a_2|, |b_1 - b_2|)}{\max(|a_1 - a_2|, |b_1 - b_2|)} \right)^\lambda, \quad (8.15)$$

where a_1, a_2, b_1 , and b_2 are respectively the two model (source) and data (destination) points in the compared matching strategies. This is derived from a *Lipschitz distance*, providing a *relative* measure of distortion of the global Euclidean metric. Parameter λ allows making the enforcement of the conservation of the Euclidean distance more or less strict.

Since, contrary to the matching setup, in the inlier selection framework we are only interested in a few good correspondences, even after converging to an ESS, we select only a small set of the support to estimate the rigid transformation. In particular, we keep only strategies whose population proportion is more than a given ration of the maximum surviving population.

8.6.3 Application to Surface Alignment

In order to explore the role of both the discriminant feature detector and the matching technique, we designed a wide range of experimental validations. First, we analyzed the sensitivity of the descriptor to several sources of noise and the influence of the number of scales (and thus of the size of the descriptor vector). Further, we studied the sensitivity of the matching algorithm to its parameters, with the goal of identifying an optimal parameterization (if any) and assess the stability of the method. Also a number of comparative test were made. Specifically, we analyzed the performance obtained by using our matcher with different feature detectors and the overall comparison with respect to other well-know registration pipelines.

All the experiments were performed on a personal computer equipped with an Intel Core i7 processor and 8 GB of memory. The dataset used, where not differently stated, was built upon publicly available models; specifically the Bunny [54], the

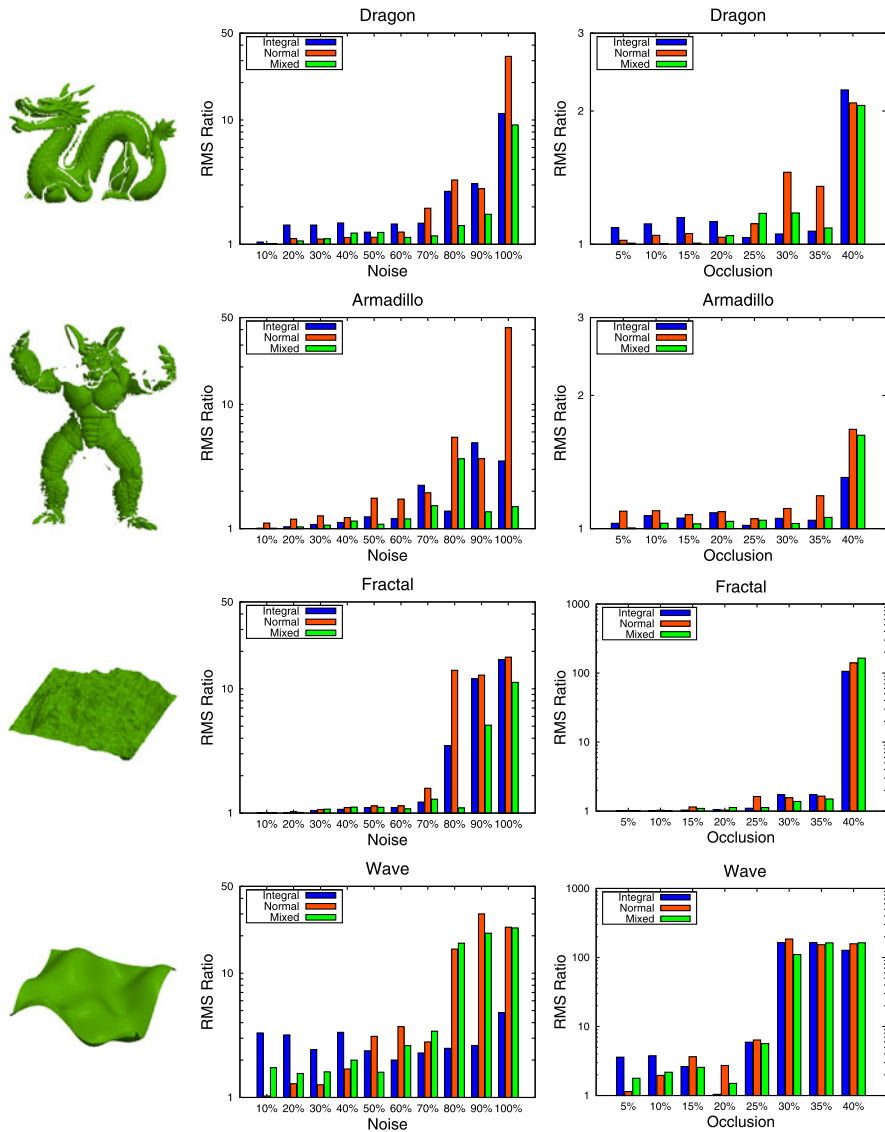


Fig. 8.7 Comparison of different descriptors using real and synthetic objects

Armadillo [32], and the Dragon [17] from the Stanford 3D scanning repository. To further assess the shortcomings of the various approaches, we used two synthetic surfaces representative of as many difficult classes of objects: a wave surface and a fractal landscape (see Fig. 8.7). Since a ground truth was needed for an accurate quantitative comparison, we generated virtual range images from the models and

then applied additive Gaussian noise to them. The descriptor used was a mixed Surface Hash with 3 scales.

8.6.3.1 Sensitivity Analysis of the Descriptor

The performance of different descriptors was tested for various levels of noise and occlusion applied to two surfaces obtained from real range scans (“armadillo” and “dragon” from Stanford) and two synthetic surfaces designed to be challenging for coarse and fine registration techniques (“fractal” and “wave”). The noise is a positional Gaussian perturbation on the point coordinates with its level (σ) expressed in terms of the percentage of the average edge length, while occlusion denotes the percentage of data and model surfaces removed. The RMS Ratio in the charts is the ratio of the root mean square error (RMS) obtained after registration and the RMS of ground truth alignment. The Normal and Integral Hashes were calculated over 3 levels of scale and the “Mixed” Hash is simply the juxtaposition of the previous two.

In Fig. 8.7, we see that all the descriptors obtain good results with real range images and the registration “breaks” only with very high levels of noise (on the same order of magnitude of the edge length). Interestingly, the Mixed Hash always obtains the best performance, even with high level of noise: This higher robustness is probably due to the orthogonality between the Normal and Integral Hashes. The behavior with the “fractal” synthetic surface is quite similar, by contrast all the descriptors seem to perform less well with the “wave” surface. This is due to the lack of distinctive features on the model itself, which indeed represents a challenge for any feature based registration technique [47]. The performance obtained with respect to occlusion is similar: all the descriptors achieve fairly good results and are resilient to high levels of occlusion (note that 40 percent occlusion is applied both to data and model). Overall the Mixed Hash appears to be consistently more robust. Since we found that the descriptors calculated over 3 levels of scale break at a certain level of noise, we were interested in evaluating if their performance can be improved by increasing their dimension.

In Fig. 8.8, we present the results obtained with different levels of scale for the Mixed Hash. The graphs show the average over all the surfaces and the associated RMS. It is interesting to observe that by reducing the scale level the technique becomes less robust, whereas its performance increases dramatically when the number of scales increases. With a scale level of 5 our approach can deal even with surfaces subject to Gaussian positional noise of σ greater than the edge length. Unfortunately, this enhanced reliability comes with a drawback: by using larger levels of scale the portion of boundary that cannot be characterized grows. In the right half of Fig. 8.8, the shrinking effect is shown for scale levels from 2 to 5.

8.6.3.2 Sensitivity to the Parameters of the Matcher

The game-theoretic matching technique presented basically depends on four parameters:

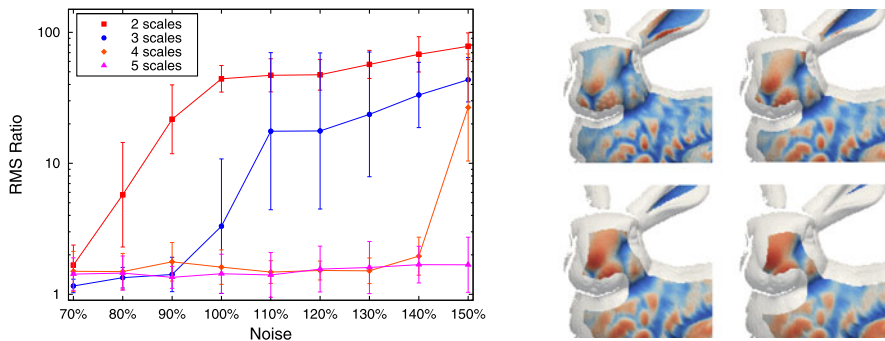


Fig. 8.8 Effect of scale on the matching accuracy

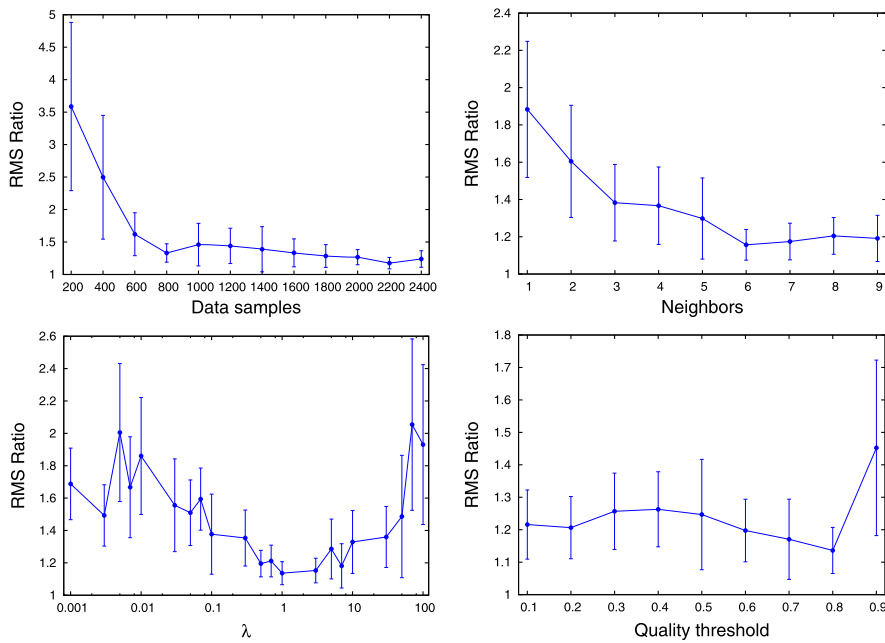


Fig. 8.9 Analysis of the sensitivity of the Game-Theoretic Matcher with respect to the parameters of the algorithm

- The number of points sampled from the model object;
- The number k of neighbors considered when building the initial set of candidates;
- The selectivity λ for the rigidity-enforcing payoff (8.15);
- The quality threshold used to deem a strategy as non-extinct upon convergence.

The first two parameters are related to the building of the set of strategies S . In Fig. 8.9, it can be seen that optimal results can be achieved with less than 1000 samples and that there is virtually no gain in using more than 6 neighbors.

The third parameter (λ) is related to the level of strictness with respect to the enforcement of the rigidity constraint: Higher values for λ will make the payoff function more steep, thus making the selection process more picky. By contrast, lowering λ will yield a payoff matrix with smaller variance, up to the limit value of 0, when the matrix assumes value 1.0 for all the strategies pairs that do not break the one-to-one constraint and 0 otherwise. As expected, our experiments show that very low or very high values for λ deliver poor results and, while there is clearly a larger variance that what has been captured by the experiments, the optimal value seems to be around 1.

Finally, the fourth parameter sets the ratio (with respect to the most successful match) used to classify a strategy as surviving or extinct. The last experiment of Fig. 8.9 shows that all the tested values below 0.8 give similarly good results. This simply means that there is good separability between extinct and non-extinct strategies, the former being very close to 0.

Overall, we can assess that the matching method has a very limited dependency on its parameters, which can easily be fixed at values that are both safe and efficient. The most influent parameter is probably λ ; however, a value of 1.0 (that indeed simplifies equation (8.15) to a simple ratio) appears to be optimal for our test set.

8.6.3.3 Comparison with Full Pipelines

The whole registration algorithm presented can be classified as a coarse method, since it does not require initialization. For this reason, we compared it with several other coarse techniques. Specifically, we implemented the whole Spin Images pipeline [29] and used the implementation supplied by the authors respectively for the MeshHOG/MeshDOG [61] and the Four Points Congruent Sets [2] methods. The latter method was initialized both with the parameters suggested by the authors and also with values for t and s that we manually optimized to get the best possible results from our dataset.

In the first row of Fig. 8.10, we present the results of this comparison. In these experiments, the occlusion is measured with respect to each range image and is applied in opposite directions of the overlapped area. That means that with an occlusion of 10 % the actual overlap is reduced by 20 %. The noise is an additive Gaussian noise with a standard error expressed as a percentage over the average edge length. The occlusion test has been made with noise at level 10 % and the noise test was performed with no occlusion. From the tests our method exhibits better results in both scenarios and breaks only with high levels of occlusion and noise. Note that the 4PCS method with parameters $t = 0.9$ and $s = 500$ does not always give a feasible solution with any occlusion greater than 10 %. With extreme levels of noise the 4PCS seems to get better and obtains lower RMS ratios than our method. The reduction in performance of our method is related to the breaking of the descriptors, that at such high levels of noise do not carry sufficient information any more. A clarification should finally be made about the apparent improvement that 4PCS seems to exhibit as noise increases. In fact, at high noise levels the RMS associated

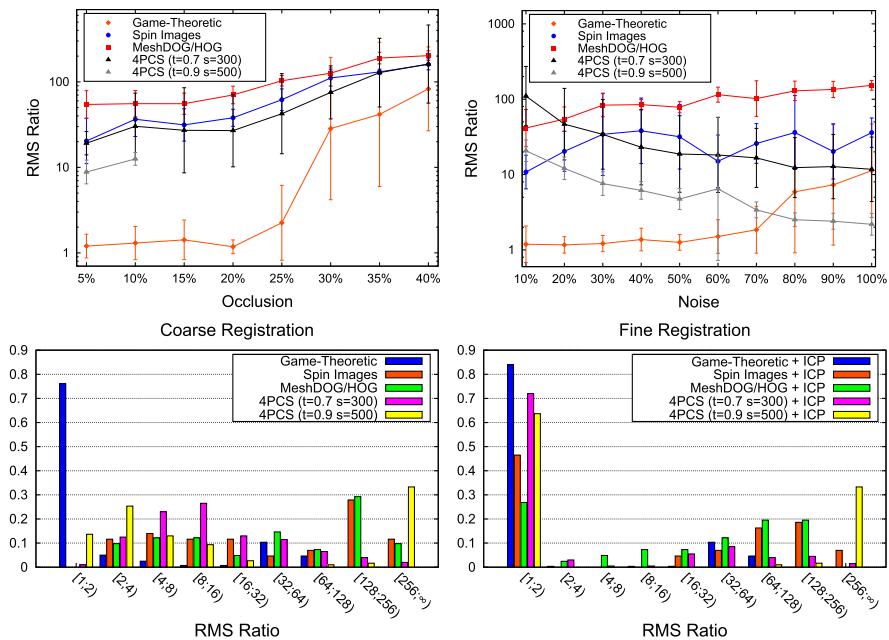


Fig. 8.10 Comparisons between our Game-Theoretic Registration technique and other widely used surface registration pipelines

to ground-truth motion is also high. In such conditions the additional error due to misalignment becomes less relevant in terms of contribution to the overall RMS ratio, which is dominated by random noise. Since 4PCS explores thoroughly the set of feasible motions until a solution with RMS low enough is found (depending on the stop criteria), it is expected to test more alignments when surfaces are noisier and thus yield lower RMS ratio values. However, it is easy to build simple examples where a solution can obtain a low RMS ratio (even lower than one) and still being far from the correct alignment. Figure 8.11 shows an example coarse registration obtained respectively with Spin Images, 4PCS, and the Game-Theoretic registration technique.

These results only indicate that GTR gives a better coarse registration; however, to seek a perfectly fair comparison, it is also needed to measure how much enhancement can be obtained by performing a fine registration step starting from the obtained coarse initialization. To this end, we applied the ICP algorithm starting from the initial motion estimated with the different methods with no occlusion and random noise values below 60 %. The results are shown in the bottom row of Fig. 8.10 with histograms obtained by binning the distance between model points and data surface along the normal vector. Normals that do not intersect the data surface are discarded. The size of the bins grows exponentially. The first histogram shows the distribution obtained from the coarse registration and the second reports

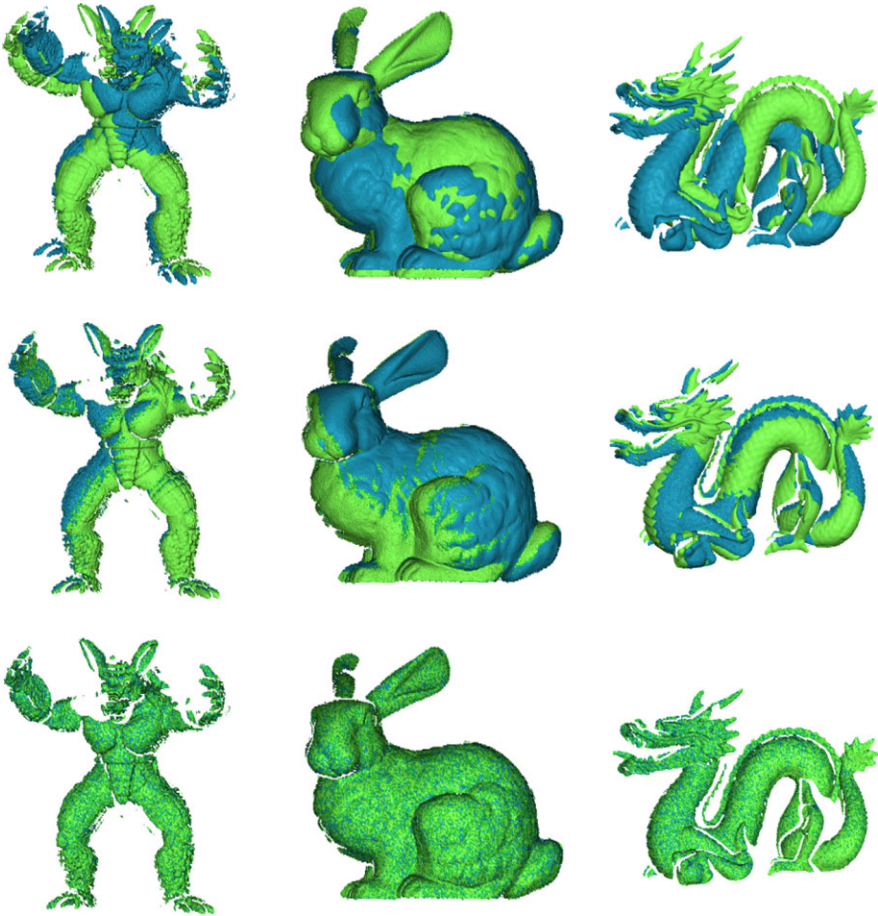


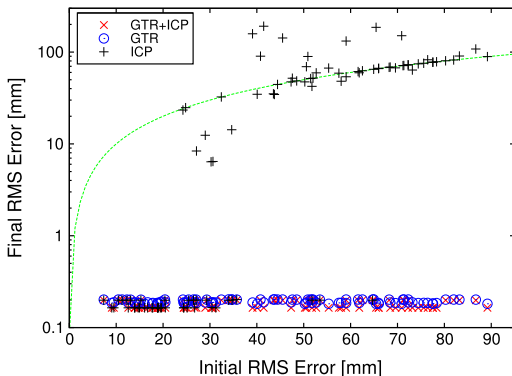
Fig. 8.11 Examples of surface registration obtained respectively with Spin Images (*first row*), MeshDOG (*second row*), 4PCS (*third column*) and our Game-Theoretic Registration technique (*last row*)

the enhancement obtained by applying ICP. Again, the results are favorable to our method, with very few points exhibiting large errors after refinement.

8.6.3.4 Quality of Fine Registration

In addition to the full pipeline comparisons, we also investigated how reliable the proposed approach would be if directly used as a fine registration technique. The goal of this test is two-fold: we want to evaluate our quality as a complete alignment tool and, at the same time, find the breaking point of traditional fine registration techniques.

Fig. 8.12 Comparison of fine registration accuracies (the green dashed line represents $y = x$)



The method we used for comparison is a best-of-breed ICP variant, similar to the one proposed in [54]. Point selection is based on Normal Space Sampling [47], and point-surface normal shooting is adopted for finding correspondences; distant mates or candidates with back-facing normals are rejected. To minimize the influence of incorrect normal estimates, matings established on the boundary of the mesh are also removed. The resulting pairings are weighted with a coefficient based on compatibility of normals, and finally a 5 %-trimming is used. Each test was performed by applying a random rotation and translation to different range images selected from the Stanford 3D scanning repository. Additionally, each range image was perturbed with a constant level of Gaussian noise with standard deviation equal to 12 % of the average edge length. We completed 100 independent tests and for each of them we measured the initial RMS error between the ground-truth corresponding points and the resulting error after performing a full round of ICP (ICP) and a single run of our registration method (GTR). In addition, we applied a step of ICP to the registration obtained with our method (GTR + ICP) in order to assess how much the solution extracted using our approach was further refinable.

A scatter plot of the obtained errors before and after registration is shown in Fig. 8.12. The final error is on a log-scale, so the dotted curve represent the points with identical initial and final error. We observe that ICP reaches its breaking point quite early; in fact, with an initial error above the threshold of about 20 mm it is unable to find a correct registration. By contrast, GTR is able to obtain excellent alignment regardless of the initial motion perturbation. Finally, applying ICP to GTR decreases the RMS only by a very small amount.

8.7 Conclusions

In this chapter, we have introduced a game-theoretic formulation of the clustering problem which is able to work with non-metric (dis)similarities (even asymmetric and negative ones). Within our framework, the problem of clustering a set of data elements is viewed as a non-cooperative clustering game and classical equilibrium

notions from evolutionary game theory turn out to provide a natural formalization of the notion of a cluster. Our game-theoretic perspective has the following attractive features: it makes no assumption on the underlying (individual) data representation, e.g., spectral clustering, it does not require that the elements to be clustered be represented as points in vector space; it does not require a priori knowledge on the number of clusters (since it extracts them sequentially); it leaves clutter elements unassigned (useful, e.g., in figure/ground separation or one-class clustering problems); it allows extracting overlapping clusters (see, e.g., [53]); and it can naturally handle high-order similarities. Besides the game-theoretic connotation, we have provided also a combinatorial characterization of our notion of a cluster and established conditions under which relations with optimization theory and graph theory exist. Furthermore, we have focused our attention on the algorithmic aspects of computing equilibria in our clustering game. Specifically, we have reviewed a class of dynamics developed within the evolutionary game theory, the replicator dynamics being one representative, that can be used to find equilibria in clustering games. In addition, we have proposed a new class of dynamics for the same purpose that overcomes some limitations of the classical evolutionary dynamics.

Finally, the proposed approach was adapted to address generic matching problems and inlier selection problems, where a low rate of false positive is required, even at the expense of a high number of false negatives. The approach applied to point-pattern matching and 3D reconstruction problems provided performance clearly at the state-of-the-art.

References

1. Ackerman, M., Ben-David, S.: Measures of clustering quality: a working set of axioms for clustering. In: *Advances in Neural Inform. Process. Syst. (NIPS)* (2008)
2. Aiger, D., Mitra, N.J., Cohen-Or, D.: 4-points congruent sets for robust surface registration. *ACM Trans. Graph.* **27**(3), 1–10 (2008)
3. Albarelli, A., Torsello, A., Rota Bulò, S., Pelillo, M.: Matching as a non-cooperative game. In: *Int. Conf. Comp. Vision (ICCV)* (2009)
4. Almohamad, H.A., Duffuaa, S.O.: A linear programming approach for the weighted graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(5), 522–525 (1993)
5. Altschul, S.F., Gish, W., Miller, W., Meyers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* **215**(3), 403–410 (1990)
6. Barrow, H., Burstall, R.M.: Subgraph isomorphism, matching relational structures and maximal cliques. *Inf. Process. Lett.* **4**(4), 83–84 (1976)
7. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
8. Bomze, I.M.: Evolution towards the maximum clique. *J. Glob. Optim.* **10**(2), 143–164 (1997)
9. Bomze, I.M.: On standard quadratic problems. *J. Glob. Optim.* **13**(4), 369–387 (1998)
10. Bomze, I.M., Pötscher, B.M.: *Game Theoretical Foundations of Evolutionary Stability*. Springer, Berlin (1989)
11. Bomze, I.M., Weibull, J.W.: Does neutral stability imply Lyapunov stability? *Games Econ. Behav.* **11**, 173–192 (1995)
12. Calana, Y.P., Cheplygina, V., Duin, R.P.W., Reyes, E.B.G., Orozco-Alzate, M., Tax, D.M.J., Loog, M.: On the informativeness of asymmetric dissimilarities. In: *Hancock, E.R., Pelillo, M.*

- (eds.) SIMBAD. Lecture Notes in Computer Science, vol. 7953, pp. 75–89. Springer, Berlin (2013)
13. Chen, C.S., Hung, Y.P., Cheng, J.B.: RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(11), 1229–1234 (1999)
 14. Chum, O., Matas, J.: Matching with PROSAC—progressive sample consensus. In: *CVPR*, pp. 220–226. IEEE Comput. Soc., Washington (2005)
 15. Chung, D.H., Yun, I.D., Lee, S.U.: Registration of multiple-range views using the reverse-calibration technique. *Pattern Recognit.* **31**(4), 457–464 (1998)
 16. Cramer, K., Talukdar, P.P., Pereira, F.: A rate-distortion one-class model and its applications to clustering. In: *Int. Conf. on Mach. Learning (ICML)* (2008)
 17. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: *Proc. 23rd ACM Annual Conf. on Computer Graphics and Interactive Techniques—SIGGRAPH'96*, pp. 303–312 (1996)
 18. Dubuisson, M.P., Jain, A.K.: A modified Hausdorff distance for object matching. In: *Int. Conf. Patt. Recogn. (ICPR)*, pp. 566–568 (1994)
 19. Edmonds, J.: Paths, trees, and flowers. *Can. J. Math.* **17**, 449–467 (1965). www.cs.berkeley.edu/~christos/classics/edmonds.ps
 20. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**(6), 381–395 (1981)
 21. Fudenberg, D., Tirole, J.: *Game Theory*. MIT Press, Cambridge (1991)
 22. Geusebroek, J.M., Burghouts, G.J., Smeulders, A.W.M.: The Amsterdam library of object images. *Int. J. Comput. Vis.* **61**(1), 103–112 (2005)
 23. Gupta, G., Ghosh, J.: Robust one-class clustering using hybrid global and local search. In: *Int. Conf. on Mach. Learning (ICML)* (2005)
 24. Herault, L., Horaud, R.: Figure-ground discrimination: a combinatorial optimization approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(9), 899–914 (1993)
 25. Ho, J., Ming-Hsuan, Y., Jongwoo, L., Kuang-Chih, L., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: *IEEE Conf. Computer Vision and Patt. Recogn. (CVPR)*, vol. 1, pp. 11–18 (2003)
 26. Hofbauer, J., Sigmund, K.: *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge (1998)
 27. Jacobs, D.W., Weinshall, D., Gdalyahu, Y.: Classification with nonmetric distances. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(6), 583–600 (2000)
 28. Jain, A.K., Dubes, R.C.: *Algorithms for Data Clustering*. Prentice Hall, New York (1988)
 29. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5), 433–449 (1999)
 30. Kim, J., Kolmogorov, V., Zabih, R.: Visual correspondence using energy minimization and mutual information. In: *IEEE Int. Conf. Computer Vision*, pp. 1033–1040 (2003)
 31. Kleinberg, J.M.: An impossibility theorem for clustering. In: *Advances in Neural Inform. Process. Syst. (NIPS)* (2002)
 32. Krishnamurthy, V., Levoy, M.: Fitting smooth surfaces to dense polygon meshes. In: *Proc. of SIGGRAPH*, vol. 96, pp. 313–324 (1996)
 33. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision*, vol. 20, pp. 91–110 (2003)
 34. Luenberger, D.G.: *Linear and Nonlinear Programming*. Addison-Wesley, Reading (1984)
 35. Maynard Smith, J.: *Evolution and the Theory of Games*. Cambridge University Press, Cambridge (1982)
 36. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. *Can. J. Math.* **17**, 533–540 (1965)
 37. Pardalos, P.M., Phillips, A.T.: A global optimization approach for solving the maximum clique problem. *Int. J. Comput. Math.* **33**, 209–216 (1990)

38. Pavan, M., Pelillo, M.: Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1), 167–172 (2007)
39. Pelillo, M.: Replicator equations, maximal cliques, and graph isomorphism. *Neural Comput.* **11**(8), 1933–1955 (1999)
40. Pelillo, M., Jagota, A.: Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Netw.* **2**, 411–420 (1995)
41. Pelillo, M., Siddiqi, K., Zucker, S.W.: Matching hierarchical structures using association graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(11), 1105–1120 (1999)
42. Rota Bulò, S., Bomze, I.M.: Infection and immunization: a new class of evolutionary game dynamics. *Games Econ. Behav.* **71**, 193–211 (2011)
43. Rota Bulò, S., Pelillo, M.: A game-theoretic approach to hypergraph clustering. In: *Advances in Neural Inform. Process. Syst. (NIPS)*, vol. 22, pp. 1571–1579 (2009)
44. Rota Bulò, S., Pelillo, M.: A game-theoretic approach to hypergraph clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1312–1327 (2013)
45. Rota Bulò, S., Pelillo, M., Bomze, I.M.: Graph-based quadratic optimization: a fast evolutionary approach. *Comput. Vis. Image Underst.* **115**, 984–995 (2011)
46. Roth, V., Laub, J., Kawanabe, M., Buhmann, J.M.: Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 1540–1551 (2003)
47. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: *Proc. of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, pp. 145–152 (2001)
48. Salvi, J., Matabosch, C., Fofi, D., Forest, J.: A review of recent range image registration methods with accuracy evaluation. *Image Vis. Comput.* **25**(5), 578–596 (2007)
49. Shashua, A., Ullman, S.: Structural saliency: The detection of globally salient features using a locally connected network. In: *Int. Conf. Comp. Vision (ICCV)* (1988)
50. Tarel, J.P., Civi, H., Cooper, D.B.: Pose estimation of free-form 3d objects without point matching using algebraic surface models. In: *Proceedings of IEEE Workshop Model Based 3D Image Analysis*, pp. 13–21 (1998)
51. Torsello, A., Hancock, E.R.: Computing approximate tree edit distance using relaxation labeling. *Pattern Recognit. Lett.* **24**, 1089–1097 (2003)
52. Torsello, A., Rota Bulò, S., Pelillo, M.: Grouping with asymmetric affinities: a game-theoretic perspective. In: *IEEE Conf. Computer Vision and Patt. Recogn. (CVPR)*, pp. 292–299 (2006)
53. Torsello, A., Rota Bulò, S., Pelillo, M.: Beyond partitions: Allowing overlapping groups in pairwise clustering. In: *Int. Conf. Patt. Recogn. (ICPR)* (2008)
54. Turk, G., Levoy, M.: Zipped polygon meshes from range images. In: *Proc. 21st ACM Annual Conf. on Computer Graphics and Interactive Techniques—SIGGRAPH'94*, pp. 311–318 (1994)
55. Umeyama, S.: An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(5), 695–703 (1988)
56. Weibull, J.: *Evolutionary Game Theory*. MIT Press, Cambridge (1995)
57. Weibull, J.W.: *Evolutionary Game Theory*. Cambridge University Press, Cambridge (1995)
58. Williams, J.W., Thornber, K.K.: A comparison of measures for detecting natural shapes in cluttered backgrounds. *Int. J. Comput. Vis.* (2000)
59. Yu, S., Shi, J.: Grouping with directed relationships. In: *Energy Minim. Methods in Computer Vision and Patt. Recogn.*, pp. 283–297 (2001)
60. Zadeh, R.B., Ben-David, S.: A uniqueness theorem for clustering. In: *Uncertainty in Artif. Intell* (2009)
61. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.P.: Surface feature detection and description with applications to mesh matching. In: *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognit.* Miami Beach, Florida (2009)

Part IV

Applications

Chapter 9

Automated Analysis of Tissue Micro-Array Images on the Example of Renal Cell Carcinoma

Peter J. Schüffler, Thomas J. Fuchs, Cheng Soon Ong, Volker Roth,
and Joachim M. Buhmann

Abstract Automated tissue micro-array analysis forms a challenging problem in computational pathology. The detection of cell nuclei, the classification into malignant and benign as well as the evaluation of their protein expression pattern by immunohistochemical staining are crucial routine steps for human cancer research and oncology. Computational assistance in this field can extremely accelerate the high throughput of the upcoming patient data as well as facilitate the reproducibility and objectivity of qualitative and quantitative measures. In this chapter, we describe an automated pipeline for staining estimation of tissue micro-array images, which comprises nucleus detection, nucleus segmentation, nucleus classification and staining estimation among cancerous nuclei. This pipeline is a practical example for the importance of non-metric effects in this kind of image analysis, e.g., the use of shape information and non-Euclidean kernels improve the nucleus classification performance significantly. The pipeline is explained and validated on a renal clear cell carcinoma dataset with MIB-1 stained tissue micro-array images and survival data of 133 patients. Further, the pipeline is implemented for medical use and research purpose in the free program TMARKER.

P.J. Schüffler (✉) · J.M. Buhmann
Swiss Federal Institute of Technology Zurich, Zurich, Switzerland
e-mail: peter.schueffler@inf.ethz.ch

J.M. Buhmann
e-mail: jbuhmann@inf.ethz.ch

T.J. Fuchs
California Institute of Technology, Pasadena, CA 91125, USA
e-mail: fuchs@caltech.edu

C.S. Ong
National ICT Australia, Melbourne, Australia
e-mail: chengsoon.ong@unimelb.edu.au

V. Roth
Computer Science Department, University of Basel, Basel, Switzerland
e-mail: volker.roth@unibas.ch

9.1 Introduction

The clinical workflow of cancer tissue analysis is composed of several estimation and classification steps which yield a diagnosis of the disease stage and a therapy recommendation. This subproject of SIMBAD proposes an automated system to model such a workflow which is able to provide more objective estimates of cancer cell detection and nuclei counts than pathologists had achieved in this study. Our image processing pipeline is tailored to renal cell carcinoma (RCC), which is one of the ten most frequent malignancies in Western societies. The prognosis of renal cancer is poor since many patients suffer already from metastases at the time of first diagnosis. The identification of biomarkers for prediction of prognosis (prognostic marker) or response to therapy (predictive marker) is therefore of utmost importance to improve patient prognosis. Various prognostic markers have been suggested in the past, but conventional estimation of morphological parameters is still most useful for therapeutical decisions.

Clear cell RCC (ccRCC) is the most common subtype of renal cancer and it is composed of cells with clear cytoplasm and typical vessel architecture. ccRCC shows an architecturally diverse histological structure, with solid, alveolar and acinar patterns. The carcinomas typically contain a regular network of small thin-walled blood vessels, a diagnostically helpful characteristic of this tumor. Most ccRCC samples show areas with hemorrhage or necrosis, whereas an inflammatory response is infrequently observed. The cytoplasm is commonly filled with lipids and glycogen, which are dissolved in routine histological processing, creating a clear cytoplasm surrounded by a distinct cell membrane (Fig. 9.1(d)). Nuclei tend to be round and uniform with finely granular and evenly distributed chromatin. Depending upon the grade of malignancy, nucleoli may be inconspicuous and small, or large and prominent. Very large nuclei or bizarre nuclei may occur [1].

The tissue micro-array (TMA) technology promises to significantly accelerate studies seeking for associations between molecular changes and clinical endpoints [2]. In this technology, tissue cylinders of 0.6 mm in diameter are punched from primary tumor blocks of hundreds of different patients and these cylinders are subsequently embedded into a recipient paraffin block (Fig. 9.1(a)–(b)). Slices from such array blocks can then be used for simultaneous *in situ* analysis of hundreds or thousands of primary tumors on DNA, RNA, and protein level (Fig. 9.1(b)–(c)). These results can then be integrated with expression profile data which is expected to enhance the diagnosis and prognosis of ccRCC [3–5]. The high speed of arraying, the lack of a significant damage to donor blocks, and the regular arrangement of arrayed specimens substantially facilitates automated analysis.

Although the production of tissue micro-arrays is an almost routine task for most laboratories, the evaluation of stained tissue micro-array slides remains tedious, time consuming and prone to error. Furthermore, the significant intratumoral heterogeneity of RCC results in high inter-observer variability. The variable architecture of RCC also results in a difficult assessment of prognostic parameters. Current image analysis software requires extensive user interaction to properly identify cell populations, to select regions of interest for scoring, to optimize analysis parameters,

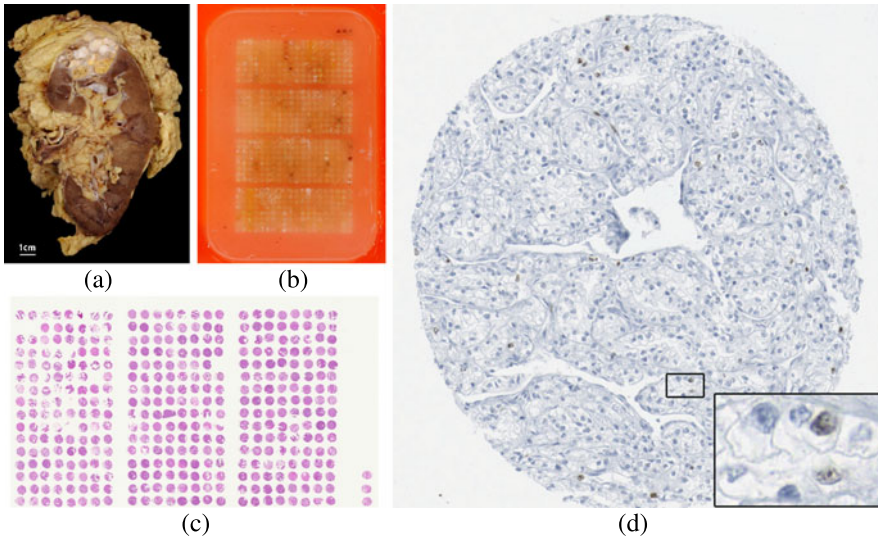


Fig. 9.1 Tissue Micro-array Analysis (TMA): Primary tissue samples are taken from a cancerous kidney (a). Then, small tissue cylinders of 0.6 mm in diameter are punched from the primary tumor block of different patients and arrayed in a recipient paraffin block (b). Slices of 0.6 μm thickness are cut off the paraffin block and are immunohistochemically stained (c). Image (d) depicts one TMA spot of clear cell renal cell carcinoma from our test set stained with the MIB-1 (Ki-67) antigen. Original size detail image is on the bottom right

and to organize the resulting raw data. Because of these drawbacks in current software, pathologists typically collect tissue micro-array data by manually assigning a composite staining score for each spot—often during multiple microscopy sessions over a period of days. Such manual scoring can result in serious inconsistencies between data collected during different microscopy sessions. Manual scoring also introduces a significant bottleneck that hinders the use of tissue micro-arrays in high-throughput analysis.

The prognosis for patients with RCC depends mainly on the pathological stage and the grade of the tumor at the time of surgery. Other prognostic parameters include proliferation rate of tumor cells and different gene expression patterns. Tannapfel et al. [6] have shown that cellular proliferation may prove to be another measure for predicting biological aggressiveness and, therefore, for estimating the prognosis. Immuno-histochemical assessment of the MIB-1 (Ki-67) antigen indicates that MIB-1 immunostaining (Fig. 9.1(d)) is an additional prognostic parameter for patient outcome. TMAs are highly representative of proliferation index and histological grade using bladder cancer tissue [7].

In the domain of cytology, especially blood analysis and smears, automated analysis is already established [8]. Histological tissue processing typically differs substantially from blood sample analysis with its homogeneous background. In blood samples, the cells are clearly distinguishable and vessels and connection tissue are typically absent. The isolation of cells simplifies the detection and segmentation

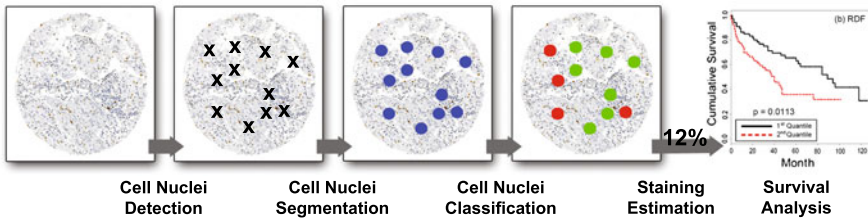


Fig. 9.2 Automated pipeline for tissue micro-array analysis. From left to right: (i) A TMA image is digitally stored for computational analysis; (ii) With object detection methods from computer vision, nuclei are identified in the image; (iii) The detected nuclei are subjected to segmentation algorithms discovering the shape of the nuclei; (iv) The segmented nuclei are conducted to feature extraction (mainly histogram-like features) and classified according to the clinical labels (here malignant/benign); (v) Among the cancerous nuclei, the amount of stained nuclei is calculated. Staining discrimination is done via thresholding in color information; (vi) In a larger patient cohort, the pipeline (i.e., staining estimation) is validated regarding a survival analysis. Validation also comprises the comparison of pathologists' staining estimation vs. the prediction of the computational pathology algorithm

process of the cells significantly. A similar simplification can be seen in the field of immunofluorescence imaging [9]. Only the advent of high resolution scanning technologies in recent years rendered it possible to consider an automated analysis of histological slices. Cutting-edge scanners are now able to scan slices with resolution, comparable to a $40\times$ lens magnification on a light microscope. In addition, the automated scanning of staples of slices enables an analysis in a high throughput manner.

9.2 Automated TMA Processing Pipeline

We propose an automated TMA processing pipeline which is enormously facilitated by the use of modern machine learning techniques. The pipeline is composed of following subsequent steps (cf. Fig. 9.2): (i) identification and detection of cell nuclei within a high resolution TMA image, (ii) segmentation of the detected nuclei, (iii) classification of the nuclei into malignant or benign, (iv) calculation of the percentage of tumor cells and protein expressing tumor cells. In this whole process, the biggest challenges for computational image processing and computer vision algorithms are the nucleus detection and segmentation. The automated TMA analysis is difficult, also because (i) the dyes are inhomogeneously dispersed in images; (ii) the cell nuclei might be located very closely to each other; and (iii) besides the nuclei, also other tissue fragments are stained in similar color and structure as the nuclei. The computational TMA assessment will benefit in following ways from automatic processing: with an automated pipeline, the TMA estimation is reproducible, objective and consistent. Also, grading can be performed cheaper, faster and with a higher throughput, since pathologists have only to confirm the grading results and judge uncertain borderline cases, instead of manually go over each

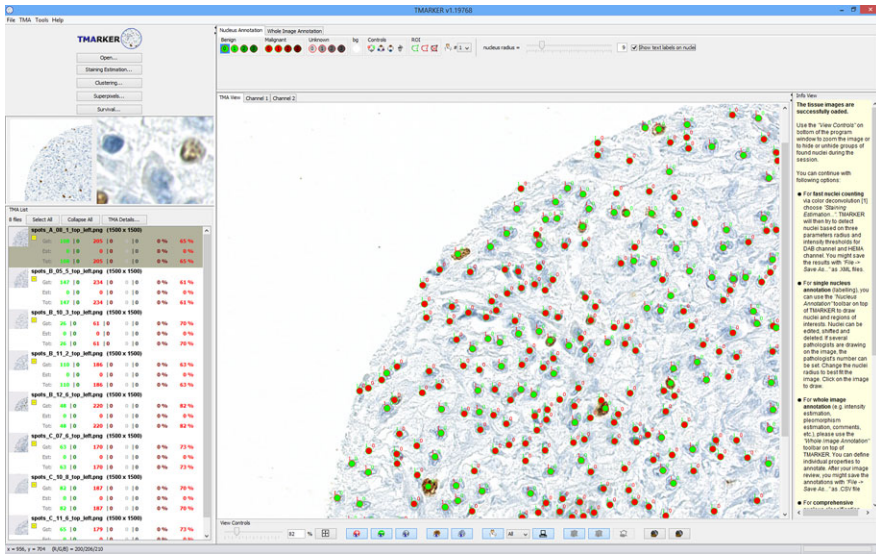


Fig. 9.3 Screenshot of TMARKER showing an image of MIB-1 stained renal clear cell carcinoma TMA. On the image, malignant and benign nuclei are marked

TMA image separately. We will outline the single steps of the pipeline in the next sections.

We also provide a Java implementation of this pipeline, called TMARKER, which facilitates the cell nuclei counting and staining estimation on immunohistochemical pathological tissue images based on the principles introduced in this chapter. TMARKER is open-source, freely available, and has a user-friendly graphical user interface (see Fig. 9.3). As a Java webstart program, it can be run without installation from any client on <http://www.comp-path.inf.ethz.ch>.

Training Data for the Pipeline The training data used in this project for the single tasks in the pipeline consist in total of 2382 manually detected cell nuclei from nine different TMA spots [10]. For each of these nuclei, two trained pathologists marked the center and the approximate radius of the nucleus (see Fig. 9.4). Based on the results and the exemplars from the classification labeling experiment, 202 cell nuclei out of the 2382 were selected as positive training examples. This set was increased to 1212 by rotating and flipping the original patches as well as the transposed patches. Additional 1291 negative examples were collected which do not contain a nucleus, but background structures and connecting tissue. All nuclei were scaled to a radius of 15 pixels and image patches of the size 65×65 pixels were extracted with a nucleus in the center of each patch. Therefore, the patches

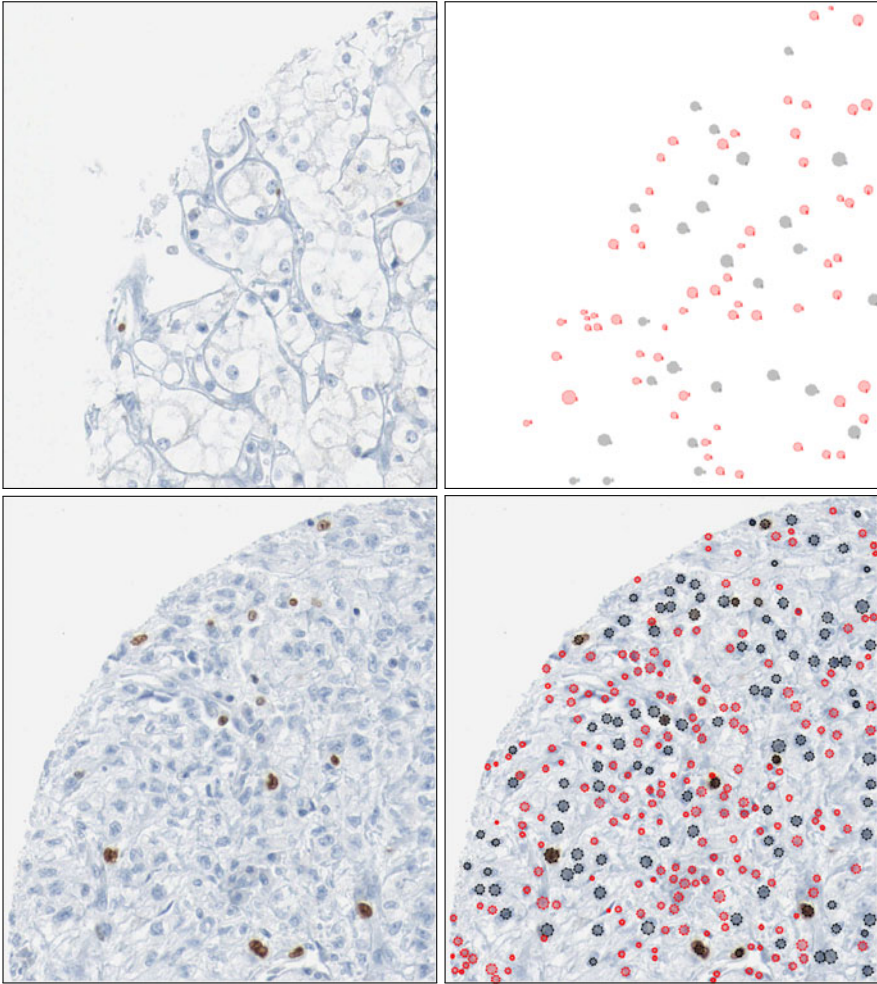


Fig. 9.4 (Left) Two top left quadrants of two ccRCC TMA spots as an example of the training dataset. (Right) A trained pathologist labeled all cell nuclei and classified them into malignant (black) and benign (red), without or with overlay. *Tissue Preparation and Scanning:* The TMA block was generated in a trial from the University Hospital Zürich. The TMA slides were immuno-histochemically stained with the MIB-1 (Ki-67) antigen and scanned on a Nanozoomer C9600 virtual slide light microscope scanner from HAMAMATSU Photonics K.K.. The 40× magnification resulted in a per pixel resolution of 0.23 μm. Finally, the spots of single patients were extracted as separate three channel color images of 3000 × 3000 pixels size. The dataset is published in [10]

contained a lot of surrounding area compared to the nuclei with a diameter of 30 pixels. In contrast to face detection, the surrounding of the objects is crucial for the classification. Nuclei often do not differ from connecting tissue by their color

or texture but only by their shape and their surroundings. Cancerous nuclei, for example, often show a bright corona around the nucleus membrane.

9.3 Nucleus Detection

The first step of the pipeline comprises the nucleus detection on the tissue images. Because of the heterogeneous nature of optical tissue images that frequently show a dense background structure which does not clearly differ from the structure of cell nuclei, we suggest a machine learning approach for nuclei detection on IHC stained tissue images. From the raw TMA image, we detect the cell nuclei by learning an ensemble of binary decision trees, using manually annotated images. This approach has been introduced in [11].

9.3.1 Tree Induction

The base learners for the ensemble are binary decision trees, designed to take advantage of large feature spaces. With minor modifications, tree learning follows the original approach for random forests described in [12]. A recursive formulation of the learning algorithm is given in procedure `LearnTree` (Procedure 1). The sub procedure `SampleFeature` returns a feature consisting of two rectangles uniformly sampled within a predefined window.

In accordance with [12], the Gini Index is used as splitting criterion, i.e., the Gini gain is maximized. At a given node, the set $S = \{s_1, \dots, s_n\}$ holds the samples for feature f_j . For a binary response $Y \in \{false, true\}$ and a feature f_j , the Gini Index of S is defined as

$$\widehat{G}(S) = 2 \frac{N_{false}}{|S|} \left(1 - \frac{N_{false}}{|S|} \right), \quad N_{false} = \sum_{s_i} I(f_j(s_i) = false), \quad (9.1)$$

where $|S|$ is the number of all samples at the current node and N_{false} denotes the number of samples evaluated to *false* by f_j . The Gini indices $\widehat{G}(S_L)$ and $\widehat{G}(S_R)$ for the left and right subset are defined similarly. The Gini gain resulting from splitting S into S_L and S_R with feature f_j is then defined as

$$\widehat{\Delta G}(S_L, S_R) = \widehat{G}(S) - \left(\frac{|S_L|}{|S|} \widehat{G}(S_L) + \frac{|S_R|}{|S|} \widehat{G}(S_R) \right), \quad (9.2)$$

where $S = S_L \cup S_R$. From this follows that a larger Gini gain is attended by a larger impurity reduction. Recently, [13] showed that the use of Gini gain can lead to selection bias because categorical predictor variables with many categories are preferred over those with few categories. In the proposed framework, this bias is not a problem due to the fact that the features are relations between sampled rectangles and therefore evaluate always to binary predictor variables.

Procedure 1: LearnTree()

Input: set of samples $S = \{s_1, s_2, \dots, s_n\}$; depth d ; max depth d_{\max} ; features to sample $mTry$

```

1 Init:  $\widehat{label} = null$ ;  $g = -\text{inf}$ ;  $N_{\text{left}} = null$ ;  $N_{\text{right}} = null$ 
2 if ( $d = d_{\max}$ ) OR ( $\text{isPure}(S)$ ) then
3   |  $\widehat{label} = \begin{cases} T & \text{if } |\{s_j = T\}| > |\{s_j = F\}|; j = 1, \dots, |S| \\ F & \text{otherwise} \end{cases}$ 
4 else
5   | for ( $i = 0, i < mTry, i++$ ) do
6     |  $f_i = \text{SampleFeature}()$ 
7     |  $S_L = \{s_j | f_i(s_j) = T\}$ ;  $S_R = \{s_j | f_i(s_j) = F\}$ ;  $j = 1, \dots, |S|$ 
8     |  $g_i = \widehat{\Delta G}(S_L, S_R)$ 
9     | if  $g_i > g$  then
10    | |  $f^* = f_i$ ;  $g = g_i$ 
11    | end
12  | end
13  |  $N_L = \text{LearnTree}(\{s_j | f^*(s_j) = T\})$ 
14  |  $N_R = \text{LearnTree}(\{s_j | f^*(s_j) = F\})$ 
15 end

```

9.3.2 Multiple Object Detection

For multiple object detection in a gray scale image, every location on a grid with step size δ is considered as an independent sample s which is classified by the ensemble. Therefore, each tree casts a binary vote for s being an object or background. The whole relational detection forests (RDF) ensemble predicts the probability of being class 1:

$$RDF(s) = \sum_{i|t_i(s)=1} \frac{1}{|\{i|t_i(s) = 1\}|}, \quad (9.3)$$

where t_i denotes the i th tree. This procedure results in an accumulator or probability map for the whole image.

The final centroids of detected objects are retrieved by applying weighted mean shift clustering with a circular box kernel of radius r . During shifting, the coordinates are weighted by the probabilities of the accumulator map. While this estimate leads to good results in most cases, homogeneous ridges in the accumulator can yield multiple centers with a pairwise distance smaller than r . Therefore, we run binary mean shift on the detection from the first run until convergence. The radius is predefined by the average object size. If the objects vary largely in size, the whole procedure can be employed for different scales. In accordance with [14], not the image but the features (resp., the rectangles) are scaled.

9.3.3 Performance Measure

One way to evaluate the quality of the nuclei detection is to consider true positive (TP), false positive (FP), and false negative (FN) rates. The calculation of these quantities is based on a matching matrix where each Boolean entry indicates if a machine extracted nucleus matches a hand labeled one or not within the average nucleus radius. To quantify the number of correctly segmented nuclei, a strategy is required to uniquely match a machine detected nucleus to one identified by a pathologist. We model this problem as a bipartite matching problem, where the bijection between extracted and gold-standard nuclei is sought inducing the smallest detection error [15]. This tuning prevents overestimating the detection accuracy of the algorithms. To compare the performance of the algorithms we calculated precision $Prec = TP/(TP + FP)$ and recall $Rec = TP/(TP + FN)$.

9.3.4 Implementation Details

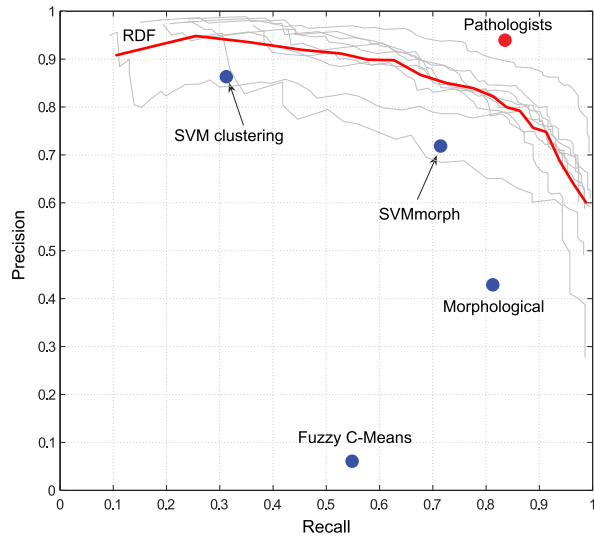
The ensemble learning framework was implemented in C# and the statistical analysis was conducted in R [16]. Employing a multi threaded architecture, tree ensembles are learned in real time on a standard dual core processor with 2.13 GHz. Inducing a tree for 1000 samples with a maximum depth of 10 and sampling 500 features at each split takes on average less than 500 ms. Classifying an image of 3000×3000 pixels on a grid with $\delta = 4$ takes approximately ten seconds using the non-optimized C# code.

Three-fold cross-validation was employed to analyze the detection accuracy of RDFs. The nine completely labeled patients were randomly split up into three sets. For each fold, the ensemble classifier was trained on data of six patients and tested on data of the remaining three. During tree induction, 500 features were sampled from the feature generator at each split. Trees were learned to a maximum depth of ten and the minimum leaf size was set to one. The forest converges after 150 steps to an out-of-bag (OOB) error of approximately 2 %. Finally, each pixel on the test images was classified and mean shift was run on a grid with step size $\delta = 5$.

9.3.5 Detection Results

Figure 9.5 shows a precision/recall plot for single patients and the average result of the RDF object detector. The algorithm is compared to point estimates of several state of the art methods: SVM clustering was successfully employed to detect nuclei in H&E stained images of brain tissue by [17]. SVMmorph is an unsupervised supervised support vector machine for filtering [18, 19]. The entry for the morphological approach for detection is combined with pathologists shows the mean

Fig. 9.5 Precision/Recall plot of cross validation results on the renal clear cell cancer (RCC) dataset. Curves for the nine single patients and their average (*bold*) are depicted for relational detection forests (RDF). RDF with the proposed feature base outperforms previous approaches based on SVM clustering [17], mathematical morphology, and combined methods [18]. The inter-pathologists' performance is depicted in the top right corner (*red dot*)



detection accuracy if alternately one expert is used as gold standard. On average, the pathologists disagree on 15 % of the nuclei.

Although only gray-scale features were used for RDF, it outperforms all previous approaches which also utilize texture and color. This observation can serve as a cue for further research that the shape information captured in this framework is crucial for good detection results.

9.4 Nucleus Segmentation

The segmentation of the nuclei is mainly used for shape describing feature extraction. Since malignant and benign nuclei typically differ in shape and size, these features promise to have a high discriminative power for classification. See Sect. 9.5.1 for nuclei feature extraction.

Two different ways of nucleus segmentation are introduced: (i) segmentation via graphcut and (ii) segmentation via superpixels. Both concepts showed promising results in shape discovery and description.

9.4.1 Segmentation via Graphcut

The cell nuclei within the rectangular image patches were segmented with an adjusted graphcut method [20–22]. Technically, we used an adapted version of the MATLAB graphcut wrapper as introduced in [23]. After gray-scaling the patches, the gray values of the pixels were bound to the sink node and reciprocally to the

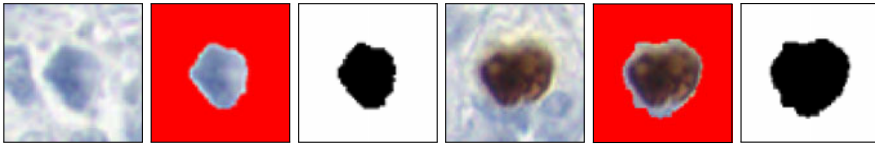


Fig. 9.6 Two examples of nucleus segmentation with graph cut: (*Left*) the original nucleus patch; (*Middle*) the segmentation via graphcut; (*Right*) the resulting shape of the nucleus

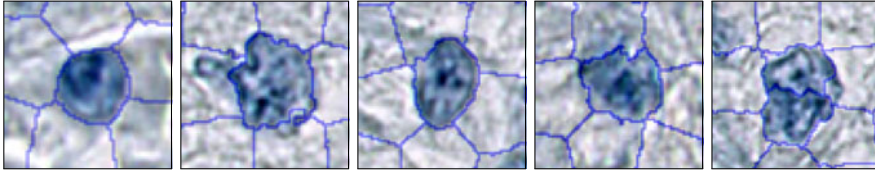


Fig. 9.7 Five examples of nucleus segmentation with superpixels in TMARKER. The shape of the nuclei is captured by the segmentation. In some cases, nuclei are cut into several superpixels (*right*)

source node. A circular shape prior was used to prefer roundish objects by weighting the binary potentials based on the pixels' distance to the center. The gray value difference between two adjacent pixels served as edge weight for the graph. After cutting the graph, the biggest connected component in the middle of a patch represented the nucleus. Some examples of graphcut segmentations can be seen in Fig. 9.6.

9.4.2 Segmentation via Superpixels

Superpixel algorithms partition a given image in smaller areas (superpixels) each with homogeneous image content. Therefore, in an optimal case, they are able to segment cell nuclei and separate them from background structures on TMA images, assuming that the superpixels are less or equal in size as the nuclei. The use of superpixels drastically reduces the amount of test samples (compared to a pixel-wise or shifting window algorithm) while simultaneously providing highly accurate segmentations of the nuclei. The superpixel algorithm is used in TMARKER for nuclei segmentation before nuclei detection, since it is not dependent on a priori detected nuclei. In fact, the superpixels are the basis in TMARKER of further cell nuclei detection and classification. We use an adopted version of the SLIC superpixel algorithm as introduced in [24]. Figure 9.7 shows the typical segmentation of an TMA image with superpixels.

9.5 Nucleus Classification

Nucleus classification is an important issue in the computer-aided tissue micro-array analysis. In short, this step comprises the decision that a given image patch shows a benign or a cancerous nucleus. Of course, such a nucleus classification plays not only an important role in the automated TMA analysis of renal cell carcinoma, but also in a high variety of different cancers as well as in the entire clinical field of tissue pathology.

In the SIMBAD project, we investigated the performance of different nucleus classification approaches within our dataset of eight TMA image spots of human renal clear cell carcinomas (see also [25]). The cell nuclei in the images are bluish stained with hematoxylin. Nuclei that express the proliferation protein MIB-1 are further stained with a brown agent. Therefore, the cell nuclei to be classified can be blue or brown. Recall that TMA image analysis is difficult, also because (i) the dyes are inhomogeneously dispersed in the image; (ii) the cell nuclei might be located very closely to each other; and (iii) besides the nuclei, also other tissue fragments are stained in similar color and structure as the nuclei.

For these experiments, the cell nuclei of the eight TMA spots were identified and labeled by two pathologists, which enabled us to extract small rectangular image patches around the nuclei as samples. Each patch shows one nucleus in the middle. The patches are the bases for all classification experiments and serve as feature sources. After introducing relevant feature representations, we will shortly outline four papers concerning the nucleus classification via image patches in the following sections. The last section finally presents an alternative for cell nuclei classification via superpixels rather than rectangular image patches. This method performs equally well and is integrated in the implemented Java program TMARKER.

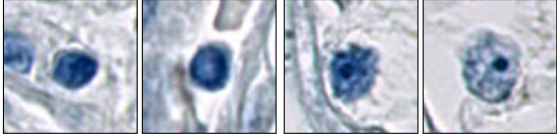
9.5.1 Image Features for Cell Nuclei

Renal cell carcinoma revealed one interesting aspect that the classification of cancerous cells can be achieved in a local fashion, i.e., patch-wise. There exist several identification guidelines of renal cell carcinoma cells for pathologists, as given in Table 9.1. We used these rules for the design of features employed in the machine learning approach. The following feature vectors were elaborated:

- **FG**—Histogram of foreground intensity (nucleus, 32-bin size histogram).
- **BG**—Histogram of background intensity (environment, 32-bin size histogram).
- **PROP**—Shape descriptors as derived from MATLAB's `regionprops` function.
- **FCC**—The Freeman Chain Code describes the cell nucleus boundary shape. The boundary shape was taken from the graphcut segmentation step (see above). A subsampling of the boundary with grid size 8 was performed to smooth the shape. Then, starting from an arbitrary point on the boundary, the boundary was redrawn in single steps. For each direction one has to go in one step, a number

Table 9.1 Guidelines used by pathologists for identifying renal clear cell carcinoma nuclei. Example image patches show typical nuclei

	Benign nucleus	Malignant nucleus
Shape	Roundish	Irregular
Nucleus membrane	Regular	Thick/thin irregular
Nucleus size	Smaller	Bigger
Nucleolus	None	Small dark spot in the nucleus
Nucleus texture	Smooth	Irregular



from 1 to 8 (for 8 orientations) describes the shape on this place. To be rotationally invariant, an 8-bin histogram of the FCC descriptor has been taken as feature vector. See [26] for a FCC implementation.

- **SIG**—The 1D signature has been implemented as described in [26]. From the center of the shape, a line is drawn to each pixel of the border line. The angles between the lines form the signature of the shape (with a maximal resolution of 1 degree). Also here, to be rotationally invariant, a 16-bin histogram of the signature has been taken as feature vector.
- **PHOG**—Pyramid histograms of oriented gradients were calculated over a level 3 pyramid and over the shapes as region of interest, as introduced in [27].
- **COL**—The separate intensity histograms over three color channels are calculated. The concatenation of all three histogram gives the color histogram.
- **LBP**—The local binary pattern for the whole superpixel is calculated as histogram over the local binary patterns of every pixel.

These features were subjected to train and test support vector machines (SVM). The training of the models was done either on all samples and labels given by one pathologist or on the subset of samples, on which both pathologists assigned the same label. The features as introduced above were concatenated to each other or taken solely.

9.5.2 Kernel Learning for Cell Nucleus Classification

In this approach, we investigated several distance measures for histogram and vectorial data [25]. In short, kernel matrices were calculated between pairwise feature vectors, with which support vector machines were trained. Since the feature vectors consist of two different types (vectorial and histogram based features), we used different corresponding kernels or distance measures. Several kernel functions and

Table 9.2 Commonly used kernels and distances for two scalar feature vectors. For the histogram features all kernels and distances were employed, while for the PROP feature only the top most three kernels were used

Kernel	Distance
Linear	Euclidean
Polynomial $d \in \{3, 5, 7, 10\}$	Intersection
Gaussian	Bhattacharya
Hellinger	Diffusion
Jensen Shanon	Kullback–Leibler
Total Variation	Earth Mover
χ^2	ℓ_1

distance measures for histograms have been investigated (see Table 9.2). The PROP features were only subjected to the linear, polynomial and Gaussian kernels, since they do not reflect a histogram like feature. The dissimilarity matrices D derived from the distance functions were centered and transformed to similarity matrices with zero mean. Also, the matrices were checked for being positive semidefinite, to serve as kernel matrices K . If needed, negative eigenvalues were mirrored:

$$D_{\text{centered}} = -0.5 * Q * D * Q, \quad Q = \begin{pmatrix} 1 - \frac{1}{n} & & -\frac{1}{n} \\ & \ddots & \\ -\frac{1}{n} & & 1 - \frac{1}{n} \end{pmatrix}, \quad (9.4)$$

$$K = V * |\Lambda| * V', \quad (9.5)$$

where n is the number of samples, V is the eigenvector matrix and Λ the eigenvalue matrix of D_{centered} .

Cell Nuclei Classification We investigated the classification performance of all different kernels, parameters and features using 10-fold cross-validation (CV) over all patches. The results clearly demonstrate that the data support automatic classification of cell nuclei into benign and malignant at a comparable performance level as the pathologists (see Fig. 9.8). The best performing kernels utilize all features: foreground and background histograms, shape descriptors and PHOG. The median misclassification error is 17 %. To confirm that we did not overfit the models, we chose the best kernel using a further cross-validation level on the training data. The found best kernel was then tested on a separate test subset of samples that was never used for training. This classifier achieved a similar median misclassification error of 18 %. In 6 out of 10 of the splits in the top level cross-validation, the diffusion distance (with all histogram features) combined with a linear kernel for the PROP features was identified as best performing SVM kernel.

Importance of Different Image Features The features that we considered can be grouped into intensity features (FG and BG), shape features (FCC, SIG and PROP) and PHOG that combines intensity gradients with a region of interest, i.e., the nucleus shape. To see how the different classes of features affect the performance of

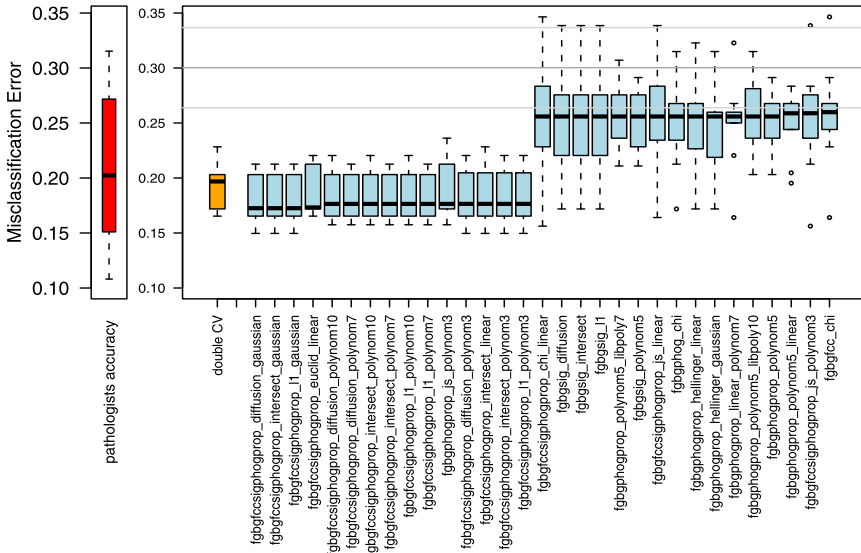


Fig. 9.8 (Left) The “performance” of the pathologists is computed from the confusion matrix between the labels of the two pathologists. (Right) Performance of kernels in nucleus classification. 15 best performing and 15 worse performing kernels (blue) are shown. Performance measure is the misclassification error in a 10-fold CV. The kernels’ names consist of the features involved (see Sect. 9.5.1) and the kernel or dissimilarity function (for histogram and non-histogram features, if needed). The orange bar represents the double CV result, indicating non-overfitting and the ability to classify new samples (see text). The horizontal line shows the mean and standard deviation of 100 permutation tests, indicating random level of prediction

classifiers, we performed a double CV over all kernels, separating the kernels into these three groups. Two conclusions could be drawn from the result in Fig. 9.9: (i) shape information improves classification performance, and (ii) the above mentioned feature classes measure different qualities of the data; combining these information improves the classifiers.

Effect of Classifier Performance on Staining Estimation Recall from the TMA processing pipeline in Fig. 9.2 that we are ultimately interested in estimating the fraction of cancerous cell nuclei that are stained. In Fig. 9.10, we document the absolute difference in error between the predicted fraction of staining (predicted staining estimation) and the fraction of staining indicated by the pathologists (observed staining estimation). First, we compared the best classifier in Fig. 9.8 to a random classifier. Our results show that a “good” classifier is also able to estimate the staining of the cancerous nuclei with higher accuracy than a random classifier (see Fig. 9.10 left). Since the fraction of stained cancerous nuclei is roughly 7 % in the data, a classifier that results in an estimate of “no staining” will have a relatively low error of 7 %. Figure 9.10 (right) demonstrates the positive relationship between nucleus classification and staining estimation. The better the classification, the lower the staining estimation error.

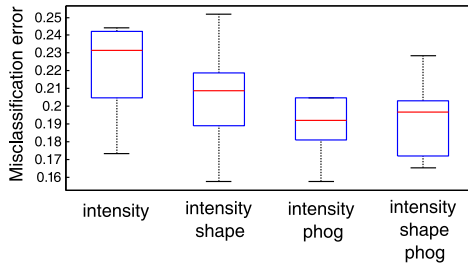


Fig. 9.9 Misclassification error of best kernels within a certain feature class (intensity: kernels using FG, BG; shape: kernels using FCC, SIG, PROP; phog: kernels using PHOG). Each bar shows the performance of the best kernel using a validation set and a double CV: in the inner CV, the best kernel in a feature class is chosen based on 90 % the samples. In the outer CV, this kernel is tested on the remaining 10 %. The plot shows that each additional feature class carries additional information for classification

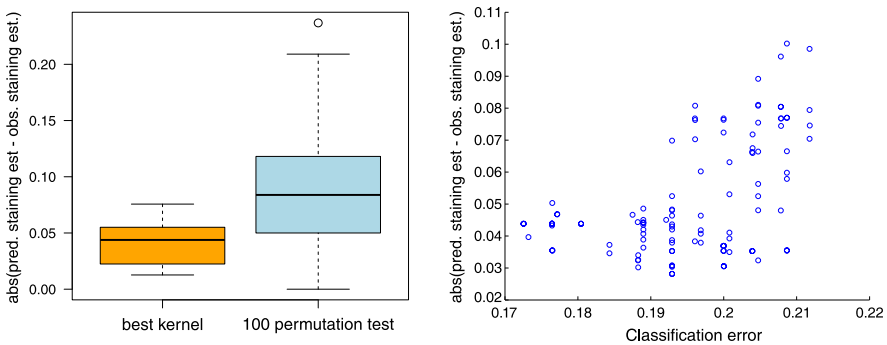


Fig. 9.10 Effect of nucleus classification performance on staining estimation. (Left) Comparison between the best classifier and a random classifier with 100 permutation tests on the staining estimation task. In a 10-fold CV, the classifier was trained and used to predict the fraction of stained vs. all cancerous nuclei in the test set. The absolute difference of the predicted fractions to the fractions based on the pathologists’ labels is shown in the plot. (Right) Relation between the classifiers’ classification performances and the staining estimation error (shown for the best 100 kernels). The staining error (absolute difference) of a classifier is calculated in the same way as in the left plot. The better the classification of a kernel (more left), the better its staining estimation (more down). The correlation coefficient $r = 0.48$

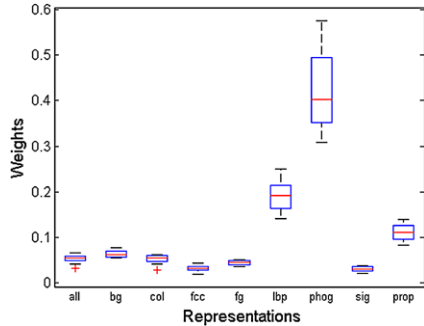
9.5.3 Multiple Kernel Learning for Cell Nucleus Classification

The classification accuracy for cell nuclei on TMA images can further be improved, e.g., with a multiple kernel learning approach, as reported in [28, 29]. Multiple kernels have the advantage that they combine different features or kernel functions and weigh these component according to their impact on the classification. In this section, we shortly outline these two studies that demonstrate linear and nonlinear MKL classification on our dataset.

Table 9.3 MKL accuracies (in %). accuracy (\pm std) of combining all kernels

	<i>svl</i>	<i>svp</i>	<i>svg</i>
SINGLE-BEST	76.5 \pm 3.7 (PHOG)	75.6 \pm 2.6 (PROP)	76.9 \pm 3.6 (PHOG)
MKL	81.3 \pm 3.6	72.0 \pm 3.3	76.9 \pm 3.6
VOTE	70.0 \pm 0.2	71.3 \pm 1.7	72.4 \pm 1.2

Fig. 9.11 Combination weights in MKL using the linear kernel. The comparable height weights of the PHOG and LBP kernels indicate their high impact for classification



Nuclei Classification Using Linear MKL In [28], the linear MKL formulation of Bach [6] has been applied to the nuclei data set to evaluate the performance of MKL on these type of image data. As a baseline, the best accuracy of a single SVM was 76.9 %. For most representations (except PHOG and COL), the accuracies of different kernels were comparable.

In this experiment, a single kernel was used combining all the feature sets extracted. Three SVM kernels were investigated: *svl* (linear), *svp* (polynomial), and *svg* (Gaussian). Table 9.3 shows the high accuracy of 81.3 % that was achieved using the linear kernel combining all representations. This experiment shows that the combination of information from multiple sources might be important and, by using MKL, the accuracy can be increased by about 5 %. The table also reveals the decrease in accuracy compared to the single best support vector machine, when using all kernels with *svp*. This phenomenon is analogous to combining all classifiers in classifier combination. If only relatively inaccurate classifiers are available, combining them all may decrease accuracy. Instead, it might be better to select a subset. From a medical viewpoint, this effect also shows that almost all the information is complementary and should be used to achieve better accuracy. In Fig. 9.11, the weights of MKL are plotted when using the linear kernel. As expected, the two best representations PHOG and PROP have high weights. But the representation LBP that has very low accuracy when considered as a single classifier increases the accuracy when considered in combination. This shows that when considering combinations, even a representation which is not very accurate alone may contribute to the combination accuracy.

Nuclei Classification Using Nonlinear MKL Besides the linear MKL approach, nonlinear kernel combinations are also possible, which can represent even more

Table 9.4 MKL classification accuracies on the cell nuclei dataset. The nonlinear combination of Gaussian kernels had the best classification accuracy of 83.3 %

SVM	<i>svl</i>	<i>svp</i>	<i>svg</i>	<i>svl + svp + svg</i>
	76.0 ± 3.4	72.7 ± 3.8	76.9 ± 2.7	NA
RBMKL	77.3 ± 4.0	77.2 ± 2.4	82.7 ± 3.6	81.8 ± 3.8
SimpleMKL	77.1 ± 3.3	77.3 ± 2.3	81.8 ± 3.8	81.6 ± 3.9
GLMKL	77.1 ± 3.5	76.5 ± 3.2	81.8 ± 4.3	81.8 ± 3.8
NLMKL	77.9 ± 3.9	79.2 ± 3.8	83.3 ± 3.6	83.1 ± 3.5

flexible structure in the model than linear approaches. The NLMKL approach on our cell nuclei dataset has been studied by Gönen et al. in [29]. In this experiment, the nonlinear kernel combination as proposed by Cortes [22] has been employed to combine the kernels *svl*, *svp* (degree 2) and *svg*. The data of 1273 nuclei samples (consensus set of the two pathologists) were divided into ten folds (with stratification). Using these folds, the SVMs *svl*, *svp*, *svg*, and MKL were trained. Four different MKL algorithms (RBMKL, SimpleMKL, GLMKL, and NLMKL) were employed to combine eight kernels calculated on nine feature representations (ALL, BG, COL, FCC, FG, LBP, PHOG, SIG, and PROP) with the same kernel function. Table 9.4 lists the results of best single-kernel SVMs and four MKL algorithms trained. With NLMKL, a high accuracy of 83.3 % by combining eight GAU kernels could be achieved. This result is better than all other MKL settings and single-kernel SVMs. In the last column of Table 9.4, the results of combining all possible feature representation and kernel function pairs (i.e., 24 kernels) in a single learner are shown. NLMKL is still the best MKL algorithm even though the average accuracy decreases to 83.1 %.

9.5.4 Hybrid Generative–Discriminative Nucleus Classification of Renal Cell Carcinoma

An exhaustive research on generative–discriminative hybrid models for nucleus classification has been performed by Ulaş et al. in [32] and Bicego et al. in [33] (see also Chap. 4).

Classification on the Generative Embedding Space Using pLSA In [32], they propose a hybrid generative/discriminative classification scheme and they have applied it to the detection of renal cell carcinoma (RCC) on tissue micro-array (TMA) images. In particular, they have used probabilistic latent semantic analysis (pLSA) as a generative model to perform generative embedding onto the free energy score space (FESS). Subsequently, they use information-theoretic kernels on these embeddings to build a kernel based classifier on the FESS. In the obtained space, different

Table 9.5 Accuracies with SVM. ORIG is the original histogram based feature approach, whereas PLSA stands for the proposed approach

	<i>svl</i>		<i>svp</i>		<i>svr</i>		<i>knn</i>	
	ORIG	PLSA	ORIG	PLSA	ORIG	PLSA	ORIG	PLSA
ALL	68.36	74.26	65.40	75.06	74.47	75.11	72.35	73.44
BG	72.88	70.82	66.79	71.50	74.22	71.92	74.25	71.29
COL	66.90	69.03	56.93	70.32	68.98	68.82	69.41	68.62
FCC	67.30	67.72	66.89	67.92	67.95	68.57	66.66	67.71
FG	70.68	71.97	64.12	72.62	70.49	71.09	69.79	70.48
LBP	68.61	69.43	42.36	70.70	68.79	70.47	71.13	70.29
PHOG	75.45	79.67	63.92	79.22	76.55	76.80	70.71	*74.69
SIG	67.72	68.34	58.64	67.69	67.72	67.72	63.50	67.72

classifiers have been tried, which have been compared with corresponding classifiers working on the original histograms (i.e., without the intermediate generative coding). Following classifiers were employed:

- (*svl*)—support vector machines with linear kernel (this represents the most widely employed solution with hybrid generative-discriminative approaches).
- (*svp*)—support vector machines with polynomial kernel: after a preliminary evaluation, the degree p was set to 2.
- (*svr*)—support vector machines with radial basis function kernel.
- (*knn*)— k -nearest neighbor classifier based on the Mahalanobis distance.

All results are reported in Table 9.5. The feature representations where the proposed approach outperforms the original classifiers are marked in bold (statistically significant difference with paired t -test, $p = 0.05$). In particular, results are averaged over ten runs.

Table 9.5 shows that the best accuracy using an SVM is 75.45 % whereas the best accuracy on the pLSA features is 79.22 %. For most representations (except LBP, PHOG and COL), the accuracies of different kernels on the original features do not exhibit large differences. We also observed that the data set cane classified as a difficult data set because some classifiers despite training only reached an accuracy equal to the prior class distribution of the data set (67 %). Except *svr*, the space constructed by pLSA always dominates the original space (except BG on *svl*) in terms of average accuracy. The bold face in the table shows feature sets where pLSA space is more accurate than the original space using 10-fold CV paired t -test at $p = 0.05$.

Application of IT Kernels on Generative Embedding Spaces Observing the success on generative embedding spaces, Bicego et al. conducted in [33] some experiments on these spaces using IT kernels developed in the context of non-vectorial data.

Table 9.6 Average accuracies (in percentage) using pLSA and FESS embeddings with SVMs. ORIG shows the baseline accuracies on the original feature space

	LIN	RBF	JS	JT	JT-W1	JT-W2
PLSA	76.78	76.99	79.31	80.17	74.22	80.17
FESS	77.41	76.17	73.21	78.87	72.31	79.96
ORIG	75.45	76.55	N/A			

Table 9.7 Average accuracies (in percentage) using pLSA and FESS embeddings with NN classifiers. ORIG shows the baseline accuracies on the original feature space with Mahalanobis distances

	MB	JS	JT	JT-W1	JT-W2
PLSA	66.41	68.97	72.53	72.74	68.75
FESS	67.11	67.08	72.53	71.27	71.08
ORIG	64.57	N/A			

In this setup, pLSA is trained in an unsupervised way, i.e., the pLSA model is learned ignoring the class labels. Table 9.6 presents the results using the posterior distribution (referred to as PLSA) and the FESS embedding with SVM classification; these results show that in the proposed hybrid generative-discriminative approach, the IT kernels outperform linear and RBF kernels. The first and second columns show the classification results of ψ and FESS scores classified using linear and RBF kernels which allows us to show the contribution of the IT kernels.

The results of the nearest neighbor (NN) classifier are shown in Table 9.7. Although NN is not a good choice for this experiment (baseline NN accuracy using Mahalanobis distance on the original data is 64.57 %), there is still an advantage of the IT kernels on the generative approach. An average accuracy of 72.74 % and 72.53 % using pLSA and FESS embeddings, respectively, can be achieved incorporating the similarities computed by the IT kernels in the NN classifier.

9.5.5 Cell Nucleus Classification of Renal Cell Carcinoma with Superpixels

Besides the patch-wise nucleus classification that has been analyzed in the previous studies, we depict here an alternative superpixel based classification approach as it is implemented in TMARKER. After having segmented the whole TMA image into superpixel, every superpixel is classified into malignant or benign. To train a classifier, the superpixels that overlap with nuclei labeled by the pathologists serve as training data. TMARKER provides basically three kinds of classifiers: random forests, support vector machines and Bayesian networks. All are derived from the Java WEKA package [34].

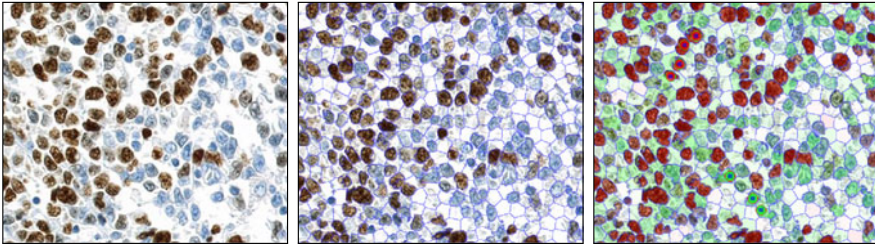


Fig. 9.12 Superpixels in TMARKER are used for segmentation, detection and classification. (*Left*) Part of the original image. (*Middle*) The image is segmented into superpixels. (*Right*) Superpixels are classified into red and green superpixels. As training set serve the labels of the user (*red* and *green* circles). The color intensity reflects the classification probability

According to the classification probability, the classification of the whole TMA image results in a probability map that visually discovers confident cases as well as borderline cases. Also, obviously wrongly classified superpixels can be identified quickly. Figure 9.12 shows a typical classification heatmap.

Extension to Detection and Classification If no prior nucleus detection algorithm has identified the relevant superpixels that represent a cell nucleus, this classification algorithm can easily be extended to a two stage classification algorithm. The first stage then classifies all superpixels into foreground (nucleus) or background. Afterwards, the second stage classifies all foreground nuclei into malignant or benign. In fact, this implies the use of two classifiers: The first classifier is trained on background samples and nuclei (both malignant and benign). Note that also a one-class classifier (e.g., one-class SVM) only trained on the nuclei would be possible for the first stage. The second classifier is only trained on malignant and benign nuclei.

Voronoi Sampling for Background Samples To establish a fully supervised training set for the first stage classifier, one needs background samples, which are generally not annotated by the domain experts. In our case, we would need locations in the TMA images that do not correspond to cell nuclei. In this context, we used a Voronoi sampling algorithm to establish this extra dataset [11]. In this approach, a Voronoi diagram is drawn with the cell nuclei as midpoints. The resulting diagram has node points exactly in the middle between the cell nuclei and the superpixels that include a Voronoi node point are considered background samples.

The extended classification algorithm with nucleus detection and classification via superpixels is implemented in TMARKER. The graphical user interface benefits from the visualization of the probability map, which facilitates the classification correction and user interaction strongly, e.g., an SVM classifier that shows the user on which superpixels it is unstable (low probability) can specifically be retrained on the information given by the user on exactly these superpixels.

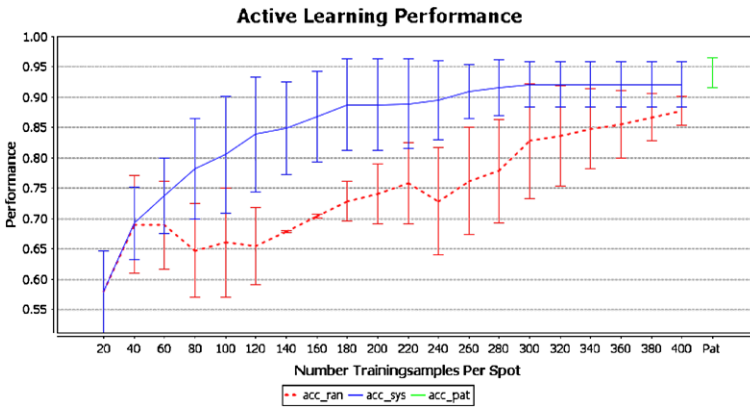


Fig. 9.13 Concept for the active learning approach in TMARKER. For given TMA images, initially 20 nuclei per image (ten per class) were selected as training set for a SVM separating malignant from benign nuclei. The classification accuracy relative to the gold standard to a pathologist is shown on the y-axis. Consecutively 20 additional nuclei per image were added repeatedly to the training (x-axis), such improving the classification performance. The additional nuclei were chosen randomly (“acc_r”) or systematically according to the (lowest) classification score (“acc_s”). The systematic approach saturates much faster. The classification accuracy reaches the range of the inter-pathologist variability (“acc_pat”)

Active Learning Approach for Detection and Classification In TMARKER, nucleus detection and classification is implemented in an active learning approach. After a preprocessing step for superpixel segmentation and feature extraction, the nucleus detection, classification and staining estimation are instantly performed while the user is labeling the image. The two stage classifiers for detection and classification are constantly updated with the user inputs. On the other hand, TMARKER provides immediate visual user feedback about the detection and classification result achieved so far as a probability map over the image. Therefore, the user can preferably label those nuclei that are borderline cases for classification. Thus, the detection and classification are considerably strengthened even after few steps of user input. This effect can be seen in Fig. 9.13, where we show that a systematic labeling of borderline cases leads faster to high classification accuracy than random labeling.

9.6 Survival Analysis

The reader may recall that the ultimate goal of TMA analysis is to determine the prognosis of the patient or to diagnose different cancer subtypes. The analysis of the proliferation marker MIB-1 enables the search for subgroups of patients which show different survival outcomes. Hence, the results of the previous two steps, cell detection and classification, can be used to estimate the proportion of cells with

particular properties (reflected by their staining with different antibodies), and ultimately their effect on patient prognosis.

9.6.1 Staining Classification

To differentiate a stained cell nucleus from a non-stained nucleus, a simple color model can be learned, when color labels are available. Based on the labeled nuclei, color histograms are generated for both classes based on the pixels within the average cell nuclei radius. To classify a nucleus on a test image the distance to the mean histograms of the both classes is calculated.

Since the dataset used for the development of TMARKER does not contain color information labels for stained and clear cell nuclei, the staining of a single nucleus is estimated by the mean color intensity of the red and blue channel of the superpixel overlaying. If the mean intensity of the red channel is higher than the mean intensity of the blue channel, the superpixel is considered as stained.

9.6.2 Kaplan–Meier Estimates

The patients are split in two (50 % : 50 %) groups based on the estimated percentage of cancerous nuclei which express MIB-1. Then the Kaplan–Meier estimator is calculated for each subgroup. This calculation involves first ordering the survival times from the smallest to the largest such that $t_1 \leq t_2 \leq t_3 \leq \dots \leq t_n$, where t_j is the j th largest unique survival time. The Kaplan–Meier estimate of the survival function is then obtained as

$$\hat{S}(t) = \prod_{j:t(j) \leq t} \left(1 - \frac{d_j}{r_j}\right) \tag{9.6}$$

where r_j is the number of individuals at risk just before t_j , and d_j is the number of individuals who die at time t_j .

To measure the goodness of separation between two or more groups, the log-rank test (Mantel–Haenszel test) is employed which assesses the null hypothesis that there is no difference in the survival experience of the individuals in the different groups. The test statistic of the log-rank test (LRT) is χ^2 distributed: $\hat{\chi}^2 = [\sum_{i=1}^m (d_{1i} - \hat{e}_{1i})]^2 / \sum_{i=1}^m \hat{v}_{1i}$ where d_{1i} is the number of deaths in the first group at t_i and $e_{1i} = n_{1j} \frac{d_i}{n_i}$ where d_i is the total number of deaths at time $t(i)$, n_j is the total number of individuals at risk at this time, and n_{1i} the number of individuals at risk in the first group.

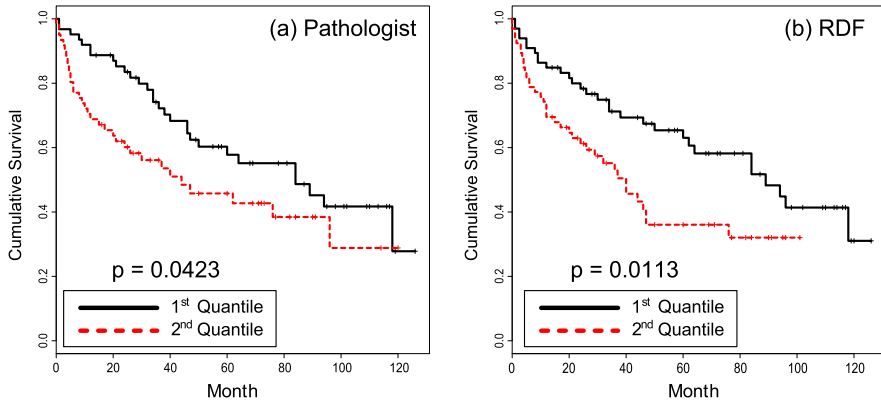


Fig. 9.14 Kaplan–Meier estimators showing significantly different survival times for renal cell carcinoma patients with high and low proliferating tumors. Compared to the manual estimation from the pathologist **(a)** ($p = 0.04$), the fully automatic estimation from the algorithm **(b)** performs better ($p = 0.01$) in terms of survival prediction on the partitioning of patients into two groups of equal size

9.6.3 Survival Estimation

One of the most important objectives and undisputed target in the medical domain relates to the survival of the patient. The experiments described in Sect. 9.1 show the large disagreement between pathologists for the estimation of staining. Therefore, the adaptation of an algorithm to the estimates of one pathologist or to a consensus voting of a cohort of pathologist is not desirable. Hence we validate the proposed algorithm against the right censored clinical survival data of 133 patients. In addition these results were compared to the estimations of an expert pathologist specialized on renal cell carcinoma. He analyzed all spots in an exceptionally thorough manner which required more than two hours. This time consuming annotation exceeds the standard clinical practice significantly by a factor of 10–20 and, therefore, the results can be viewed as an excellent human estimate for this dataset.

Figure 9.14 shows Kaplan–Meier plots of the estimated cumulative survival for the pathologist and the RDF. The further the survival estimates of the two groups are separated, the better the estimation. Quantifying this difference with log-rank test shows that the proposed algorithm is significantly ($p = 1.1 \cdot 10^{-2}$) better than the trained pathologist ($p = 4.2 \cdot 10^{-2}$).

9.7 Description of Software

TMARKER is written in Java v1.6 as a webstart application. It is platform independent and can be run from any computer with internet access and a Java Virtual Machine without installation. The program is published under the GNU general public

license. Because of the object oriented programming design of Java, TMARKER is modular and can easily be extended with new sub-pipelines, different image features or classifiers. It can be downloaded at <http://www.comp-path.inf.ethz.ch>.

9.8 Conclusion

We have proposed an automated pipeline to achieve objective and reproducible diagnosis of renal cell carcinoma. This pipeline involves three main components: cell nuclei detection from tissue micro-array images, nucleus segmentation and classification into cancerous and healthy cells, and summarizing this information and analyzing its effect on patient survival. This pipeline has been developed as open source software and is available on the SIMBAD website. The publicly available Java implementation TMARKER, which implements this pipeline, can be downloaded at <http://www.comp-path.inf.ethz.ch>. Further, this pipeline states an exhaustive example of dealing with challenges in medical imaging and computational pathology [35].

The images and comprehensive annotations by two pathologists provide a rich resource for future medical imaging research. Our publicly available data enables objective benchmarking of methods and algorithms. Furthermore, the predictions can be validated against the human annotations, leading to a deeper understanding of the variations between pathologists and its impact on designing tools to overcome this source of uncertainty. The TMA dataset is available at <http://www.mldata.org>.

Various novel pattern recognition approaches, which have been developed in the SIMBAD project, have been benchmarked on the dataset or parts of it. New kernel combination methods for nucleus classification emerged from these studies that have shed new light on automatic medical image processing in computational pathology.

Acknowledgement We thank Aydın Ulaş, Umberto Castellani, Vittorio Murino, Mehmet Gönen, Manuele Bicego, Pasquale Mirtuono, André Martins, Pedro M.Q. Aguiar and Mário A.T. Figueiredo for successful collaborations and inspiring ideas. We want to thank all our co-workers and SIMBAD partners for fruitful discussions.

References

1. Grignon, D.J., Eble, J.N., Bonsib, S.M., Moch, H.: Clear Cell Renal Cell Carcinoma. World Health Organization Classification of Tumours. Pathology and Genetics of Tumours of the Urinary System and Male Genital Organs. IARC Press, Lyon (2004)
2. Bubendorf Juha Kononen, L., Bärlund Anne Kallionimeni, M., Leighton Peter Schraml, S., Mihatsch, M.J., Torhorst, J., Kallionimeni, O.-P., Sauter, G.: Tissue microarrays for high-throughput molecular profiling of tumor specimens. *Nat. Med.* **4**(7), 844–847 (1998)
3. Takahashi, M., Rhodes, D.R., Furge, K.A., Kanayama, H.-o., Kagawa, S., Haab, B.B., Tean Teh, B.: Gene expression profiling of clear cell renal cell carcinoma: gene identification and prognostic classification. *Proc. Natl. Acad. Sci. USA* **98**(17), 9754–9759 (2001)

4. Schraml Holger Moch, P., Mirlacher Lukas Bubendorf, M., Gasser Juha Kononen, T., Kallioniemi, O.P., Mihatsch, M.J., Sauter, G.: High-throughput tissue microarray analysis to evaluate genes uncovered by CDNA microarray screening in renal cell carcinoma. *Am. J. Pathol.* **154**(4), 981–986 (1999)
5. Amin, M.B., Young, A.N., Lim, S.D., Moreno, C.S., Petros, J.A. Cohen, C., Neish, A.S., Marshall, F.F.: Expression profiling of renal epithelial neoplasms: a method for tumor classification and discovery of diagnostic molecular markers. *Am. J. Pathol.* **158**(5), 1639–1651 (2001)
6. Tannapfel, A., Hahn, H.A., Katalinic, A., Fietkau, R.J., Kühn, R., Wittekind, C.W.: Prognostic value of ploidy and proliferation markers in renal cell carcinoma. *Cancer* **77**(1), 164–171 (1996)
7. Nocito, A., Bubendorf, L., Maria Tinner, E., Süess, K., Wagner, U., Forster, T., Kononen, J., Fijan, A., Bruderer, J., Schmid, U., Ackermann, D., Maurer, R., Alund, G., Knönel, H., Rist, M., Anabitarte, M., Hering, F., Hardmeier, T., Schoenenberger, A.J., Flury, R., Jäger, P., Luc Fehr, J., Schraml, P., Moch, H., Mihatsch, M.J., Gasser, T., Sauter, G.: Microarrays of bladder cancer tissue are highly representative of proliferation index and histological grade. *J. Pathol.* **194**(3), 349–357 (2001)
8. Yang, L., Meer, P., Foran, D.J.: Unsupervised segmentation based on robust estimation and color active contour models. *IEEE Trans. Inf. Technol. Biomed.* **9**(3), 475–486 (2005)
9. Mertz, K.D., Demichelis, F., Kim, R., Schraml, P., Storz, M., Diener, P.-A., Moch, H., Rubin, M.A.: Automated immunofluorescence analysis defines microvessel area as a prognostic parameter in clear cell renal cell cancer. *Hum. Pathol.* **38**(10), 1454–1462 (2007)
10. Fuchs, T.J., Wild, P.J., Schöffler, P.J.: Labeled IHC images of RCC (2012). doi:[10.5881/LABELED-IHC-IMAGES-OF-RCC](https://doi.org/10.5881/LABELED-IHC-IMAGES-OF-RCC)
11. Fuchs, T.J., Haybaeck, J., Wild, P.J., Heikenwalder, M., Moch, H., Aguzzi, A., Buhmann, J.M.: Randomized tree ensembles for object detection in computational pathology. In: ISVC (1). *Lecture Notes in Computer Science*, vol. 5875, pp. 367–378. Springer, Berlin (2009)
12. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
13. Strobl, C., Boulesteix, A.-L., Augustin, T.: Unbiased split selection for classification trees based on the Gini index. *Comput. Stat. Data Anal.* **52**(1), 483–501 (2007)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *CVPR* (2001)
15. Kuhn, H.W.: The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955)
16. R Development Core Team: *R: a Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna (2009). ISBN 3-900051-07-0
17. Glotsos, D., Spyridonos, P., Cavouras, D., Ravazoula, P., Arapantoni Dadioti, P., Nikiforidis, G.: An image-analysis system based on support vector machines for automatic grade diagnosis of brain-tumour astrocytomas in clinical routine. *Med. Inform. Internet Med.* **30**(3), 179–193 (2005)
18. Fuchs, T.J., Lange, T., Wild, P.J., Moch, H., Buhmann, J.M.: Weakly supervised cell nuclei detection and segmentation on tissue microarrays of renal cell carcinoma. In: *Pattern Recognition. DAGM 2008. Lecture Notes in Computer Science*, vol. 5096, pp. 173–182. Springer, Berlin (2008)
19. Soille, P.: *Morphological Image Analysis: Principles and Applications*. Springer, New York (2003)
20. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004)
21. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(12), 1222–1239 (2001)
22. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 147–159 (2004)
23. Bagon, S.: *Matlab wrapper for graph cut* (2006)

24. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Suesstrunk, S.: SLIC Superpixels. Technical report, EPFL, EPFL (2010)
25. Schüffler, P.J., Fuchs, T.J., Soon Ong, C., Roth, V., Buhmann, J.M.: Computational TMA analysis and cell nucleus classification of renal cell carcinoma. In: Proceedings of the 32nd DAGM Conference on Pattern Recognition, pp. 202–211. Springer, Berlin (2010)
26. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital image processing using Matlab. 993475 (2003)
27. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR'07: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 401–408. ACM, New York (2007)
28. Schüffler, P.J., Ulaş, A., Castellani, U., Murino, V.: A multiple kernel learning algorithm for cell nucleus classification of renal cell carcinoma. In: Proceedings of the International Conference on Image Analysis and Processing, ICIAP'11 (2011). Page accepted
29. Gönen, M., Ulaş, A., Schüffler, P.J., Castellani, U., Murino, V.: Combining data sources non-linearly for cell nucleus classification of renal cell carcinoma. In: Pelillo, M., Hancock, E.R. (eds.) Proceedings of the International Workshop on Similarity-Based Pattern Analysis, SIMBAD'11. Lecture Notes in Computer Science, vol. 7005, pp. 250–260. Springer, Berlin (2011)
30. Bach, F.R., Lanckriet, G.R.G., Jordan, M.L.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21st International Conference on Machine Learning, pp. 41–48 (2004)
31. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In: Advances in Neural Information Processing Systems, vol. 22, pp. 396–404 (2010)
32. Ulaş, A., Schüffler, P.J., Bicego, M., Castellani, U., Murino, V.: Hybrid generative–discriminative nucleus classification of renal cell carcinoma. In: Pelillo, M., Hancock, E.R. (eds.) Proceedings of the International Workshop on Similarity-Based Pattern Analysis, SIMBAD'11. Lecture Notes in Computer Science, vol. 7005, pp. 77–88. Springer, Berlin (2011)
33. Bicego, M., Ulaş, A., Schüffler, P.J., Castellani, U., Mirtuono, P., Murino, V., Aguiar, P.M.Q., Martins, A., Figueiredo, M.A.T.: Renal cancer cell classification using generative embeddings and information theoretic kernels. In: Loog, M. (ed.) IAPR International Conference on Pattern Recognition in Bioinformatics, PRIB'11. Lecture Notes in Bioinformatics (accepted), vol. 7036. Springer, Berlin (2011)
34. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsl.* **11**, 10–18 (2009)
35. Fuchs, T.J., Buhmann, J.M.: Computational pathology: challenges and promises for tissue analysis. *Comput. Med. Imaging Graph.* **35**(7–8), 515–530 (2011)

Chapter 10

Analysis of Brain Magnetic Resonance (MR) Scans for the Diagnosis of Mental Illness

Aydın Ulaş, Umberto Castellani, Manuele Bicego, Vittorio Murino, Marcella Bellani, Michele Tansella, and Paolo Brambilla

Abstract We address the problem of schizophrenia detection by analyzing magnetic resonance imaging (MRI). In general, mental illness like schizophrenia or bipolar disorders are traditionally diagnosed by self-reports and behavioral observations. A new trend in neuroanatomical research consists of using MRI images to find possible connections between cognitive impairments and neuro-physiological abnormalities. Indeed, brain imaging techniques are appealing to provide a non-invasive diagnostic tool for mass analyses and early diagnoses. The problem is challenging due to the heterogeneous behavior of the disease and up to now, although the literature is large in this field, there is not a consolidated framework to deal with it. In this context, advanced pattern recognition and machine learning techniques can

A. Ulaş · U. Castellani · M. Bicego (✉) · V. Murino
Departmento di Informatica, University of Verona, Verona, Italy
e-mail: manuele.bicego@univr.it

A. Ulaş
e-mail: mehmetaydin.ulas@univr.it

U. Castellani
e-mail: umberto.castellani@univr.it

V. Murino
e-mail: vittorio.murino@univr.it

V. Murino
Istituto Italiano di Tecnologia (IIT), Genova, Italy

M. Bellani · M. Tansella · P. Brambilla
Department of Public Health and Community Medicine, Section of Psychiatry and Clinical Psychology, Inter-University Centre for Behavioural Neurosciences, University of Verona, Verona, Italy

M. Bellani
e-mail: marcella.bellani@univr.it

M. Tansella
e-mail: michele.tansella@univr.it

P. Brambilla
e-mail: paolo.brambilla@univr.it

be useful to improve the automatization of the involved procedures and the characterization of mental illnesses with specific and detectable brain abnormalities. In this book, we have exploited similarity-based pattern recognition techniques to further improve brain classification problem by employing the algorithms developed in the other chapters of this book. (This chapter is based on previous works (Castellani et al. in Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI'11, vol. 6892, pp. 426–433, 2011; Gönen et al. in Proceedings of the International Workshop on Similarity-Based Pattern Analysis, SIMBAD'11, vol. 7005, pp. 250–260, 2011; Ulaş et al. in Proceedings of the Iberoamerican Congress on Pattern Recognition, CIARP'11, vol. 7042, pp. 491–498, 2011; in IAPR International Conference on Pattern Recognition in Bioinformatics, PRIB'11, vol. 7036, pp. 306–317, 2011; and in Int. J. Imaging Syst. Technol. 21(2):179–192, 2011) by the authors and contains text, equations and experimental results taken from these papers.)

10.1 Introduction

Brain analysis techniques using Magnetic Resonance Imaging (MRI) are playing an increasingly important role in understanding pathological structural alterations of the brain [33, 70]. The ultimate goal is to identify structural brain abnormalities by comparing normal subjects with patients affected by a certain disease. Here, we focus on schizophrenia. Schizophrenia is a heterogeneous psychiatric disorder characterized by several symptoms such as hallucinations, delusions, cognitive and thought disorders [8]. Although genetic and environmental factors play a role in the disorder, its etiology remains unknown and substantial body of research has demonstrated numerous structural and functional brain abnormalities in patients with both chronic and acute forms of the disorder [66, 70].

Our main contribution here is to deal with schizophrenia detection as a binary classification problem—we have to distinguish between normal subjects and patients affected by schizophrenia [25]—by applying advanced pattern recognition techniques by exploiting the capability of similarity-based methods mentioned in the other chapters of this book to this problem.

We highlight that the problem of schizophrenia detection is very complex since the symptoms of the disease are different and related to different properties of the brain. Thus, although the literature has shown a large amount of promising methodological procedures to address this disease, up to now a consolidate framework is not available.

In this chapter, we have exploited different approaches to address schizophrenia detection. We have defined a general working pipeline composed of the four main steps: (i) data acquisition, (ii) region selection, (iii) data description, and (iv) classification. Each step may be instantiated in different ways, each one having pros and cons. Here, for each stage, we summarize the possible choices we adapted. Figure 10.1 shows the proposed overall scheme of the working pipeline and the involved possibilities. In summary:

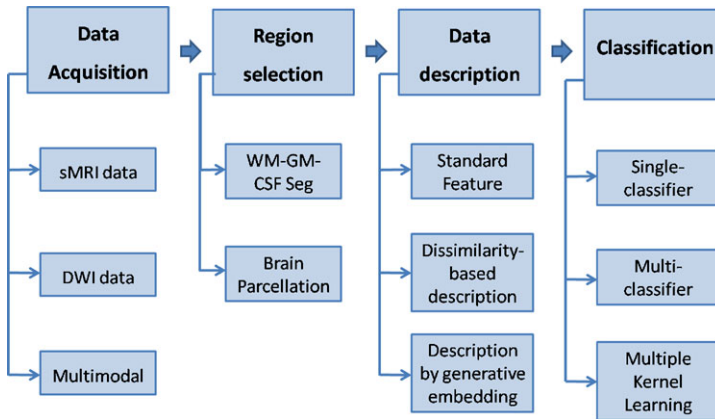


Fig. 10.1 Overall scheme of the proposed working pipeline

- *Data acquisition* regards the imaging technique employed to acquire data. Different acquisition modalities are encoding different brain information. We use Structural MRI to deal with morphological properties, and Diffusion Weighted Imaging (DWI) to evaluate functional aspects of the brain. Moreover, in order to integrate different sources of information, a multimodal approach is exploited.
- *Region selection* is necessary to focus the analysis on brain subparts. A common approach is to segment the whole brain among *White Matter* (WM), *Gray Matter* (GM), and *Cerebro-Spinal Fluid* (CSF). Another approach consists in extracting one or more Regions of Interest (ROIs) which are strictly related to the analyzed disease. The brain segmentation in ROIs is in general called *brain parcellation*.
- *Data description* aims at extracting the most useful information for the involved task, in our case brain classification. The standard approach consists in using *features*. According to the overall aim of this book, we exploited the possibility to go beyond features. Indeed, we have investigated two paradigms, derived from other chapters of the book, a dissimilarity-based description (Chap. 2) and description by generative embeddings (Chap. 4).
- *Classification* is the last step of the proposed pipeline. As simplest approach a single classifier has been employed. In order to integrate different sources of information at classification stage, we exploited two paradigms, multi-classifier approach and multiple kernel learning.

Roadmap The chapter is organized as follows: In Sect. 10.2, we present the state-of-the-art in schizophrenia detection. In Sects. 10.3 and 10.4, we introduce data acquisition and region selection, respectively. Then, data description phase is split into standard features (Sect. 10.5.1), dissimilarity-based description (Sect. 10.5.2), and description by generative embeddings (Sect. 10.5.3). We define our approaches of classification using ensembles and Multiple Kernel Learning in Sect. 10.6. We explain three case studies which utilize the working pipeline in Sects. 10.7, 10.8, and 10.9; and conclude in Sect. 10.10.

10.2 Related Work

Several works have been proposed for human brain classification in the context of schizophrenia research [70]. In the following, we have organized the state-of-the-art in (i) *shape-based* techniques and (ii) *classification-based* techniques.

10.2.1 Shape-Based Techniques

Standard approaches are based on detecting morphological differences on certain brain regions, namely Region Of Interests (ROIs). Usually, the aim is the observation of volume variations [7, 59, 70]. In general, ROI-based techniques require the manual tracing of brain subparts. In order to avoid such an expensive procedure, Voxel Based Morphometry (VBM) techniques have been introduced [4, 41] for which the entire brain is transformed onto a template, namely the stereotaxic space. In this fashion, a voxel-by-voxel correspondence is available for comparison purposes. In [41], a multivariate Voxel Based Morphometry approach method is proposed to differentiate schizophrenic patients from normal controls. Inferences about the structural relevance of gray matter distribution are carried out on several brain sub-regions. In [85], cortical changes in adolescent on-set schizophrenic patients are analyzed by combining Voxel-Based with Surface-Based Morphometry (SBM). A different approach consists in encoding the shape by a *global* region descriptor [32, 63, 76]. In [76], a new morphological descriptor is introduced by properly encoding both the displacement fields and the distance maps for amygdala and hippocampus. In [32], a ROI-based morphometric analysis is introduced by defining spherical harmonics and 3D skeleton as shape descriptors. Improvement of such a shape-descriptor-based approach with respect to classical volumetric techniques is shown experimentally. Although results are interesting, the method is not invariant to surface deformations and therefore it requires shape registration and data resampling. This pre-processing is avoided in [63], where the so-called Shape-DNA signature has been introduced by taking the eigenvalues of the Laplace–Beltrami operator as region descriptor for both the external surface and the volume. Although *global* methods can be satisfying for some classification tasks, they do not provide information about the localization of the morphological anomalies. To this aim, *local* methods have been proposed. In [77], the so called *feature-based* morphometry (FBM) approach is introduced. Taking inspiration from feature-based techniques proposed in computer vision, FBM identifies a subset of features corresponding to anatomical brain structures that can be used as disease biomarkers.

10.2.2 Classification-Based Techniques

In order to improve the capability of distinguishing between healthy and non-healthy subjects, learning-by-example techniques [27] are applied (see, for example, [25]).

Usually, geometric signatures extracted from the MRI data are used as feature vectors for classification purposes [30, 58, 87]. In [87], a support vector machine (SVM) has been employed to classify cortical thickness which has been measured by calculating the Euclidean distance between linked vertices on the inner and outer cortical surfaces. In [30], a new approach has been defined by combining deformation-based morphometry with SVM. In this fashion, multivariate relationships among various anatomical regions have been captured to characterize more effectively the group differences. Finally, in [58], a unified framework is proposed to combine advanced probabilistic registration techniques with SVM. The local spatial warps parameters are also used to identify the discriminative warp that best differentiates the two groups. It is worth to note that in most of the mentioned works, the involved classifier was a Support Vector Machine, but more general approaches are also proposed, see, e.g., [51]. Here, a set of image features which encode both general statistical properties and Law's texture features from the whole brain are analyzed. Such features are concatenated onto a very high dimensional vector which represents the input for a classic learning-by-example classification approach. Several classifiers are then evaluated such as decision trees or decision graphs. In [15], the authors proposed a neural network to measure the relevance of thalamic subregions implicated in schizophrenia. The study is based on the metabolite N-acetylaspartate (NAA) using in vivo proton magnetic resonance spectroscopic imaging. The diffusion of water in the brain characterized by its apparent diffusion coefficient (ADC), which represents the mean diffusivity of water along all directions gives potential information about the size, orientation, and tortuosity of both intracellular and extracellular spaces, providing evidence of disruption when increased [64]. DWI has been shown to be keen in exploring the microstructural organization of white matter, therefore providing intriguing information on brain connectivity [13, 78].

10.3 Data Acquisition

The data set involves a 124 subject database cared by the Research Unit on Brain Imaging and Neuropsychology (RUBIN) at the Department of Medicine and Public Health-Section of Psychiatry and Clinical Psychology of the University of Verona. The data set is composed of MRI brain scans of 64 patients recruited from the area of South Verona (i.e., 100,000 inhabitants) through the South Verona Psychiatric Case Register [2, 3, 82]. Additionally, 60 individuals without schizophrenia (control subjects) were also recruited.

10.3.1 MRI Data

MRI scans were acquired with a 1.5 T Magnetom Symphony Maestro Class Syngo MR 2002B (Siemens), and in total, it took about 19 minutes to complete an MR

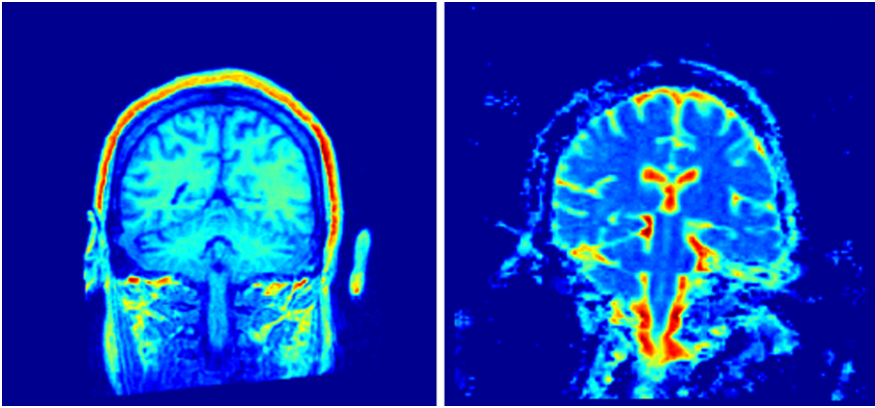


Fig. 10.2 Slices acquired by 3D Morphological technique (*left*) and Diffusion Weighting Imaging technique (*right*)

session. A standard head coil was used for radio frequency transmission and reception of the MR signal, and restraining foam pads were used to minimize head motion. T1-weighted images were first obtained to verify the participants head position and image quality (TR = 450 ms, TE = 14 ms, flip angle = 90° , FOV = 230×230 , 18 slices, slice thickness = 5 mm, matrix size = 384×512 , NEX = 2). Proton density (PD)/T2-weighted images were then acquired (TR = 2500 ms, TE = 24/121 ms, flip angle = 180° , FOV = 230×230 , 20 slices, slice thickness = 5 mm, matrix size = 410×512 , NEX = 2) according to an axial plane running parallel to the anterior–posterior (AC–PC) commissures to exclude focal lesions. Subsequently, a coronal 3-dimensional magnetization prepared rapid gradient echo (MP-RAGE) sequence was acquired (TR = 2060 ms, TE = 3.9 ms, flip angle = 15° , FOV = 176×235 , slice thickness = 1.25 mm, matrix size = 270×512 , inversion time = 1100) to obtain 144 images covering the entire brain. In Fig. 10.2 (left), we can see a slice of a subject acquired by using MRI.

10.3.2 DWI Data

Diffusion-weighted imaging (DWI) investigates molecular water mobility within the local tissue environment, providing information on tissue microstructural integrity. The diffusion of water in the brain is characterized by its apparent diffusion coefficient (ADC), which represents the mean diffusivity of water along all directions [75]. Thus, ADC gives potential information about the size, orientation, and tortuosity of both intracellular and extracellular spaces, providing evidence of disruption when increased [64]. ADC has also been used to explore regional grey matter microstructure, being higher in the case of potential neuron density alterations or volume deficit [62].

Diffusion-weighted echoplanar images in the axial plane parallel to the AC-PC line (TR = 3200 ms, TE = 94 ms, FOV = 230 × 230, 20 slices, slice thickness = 5 mm with 1.5-mm gap, matrix size = 128 × 128, echo-train length = 5; these parameters were the same for $b = 0$, $b = 1000$, and the ADC maps). Specifically, three gradients were acquired in three orthogonal directions. ADC maps (denoted by D_{ADC}) were obtained from the diffusion images with $b = 1000$, according to the following equation:

$$-bD_{ADC} = \ln[A(b)/A(0)],$$

where $A(b)$ is the measured echo magnitude, b is the measure of diffusion weighting, and $A(0)$ is the echo magnitude without diffusion gradient applied. In Fig. 10.2 (right), we can see a slice of a subject acquired by using DWI.

10.3.3 Multimodal Approach

A multimodal approach can be applied when different kinds of acquisition procedures are used for the same subject. As can be seen in Fig. 10.2, while MRI images are more reliable, DWI resolution is very low and it's hard to segment ROIs from these DWI images. In order to integrate such data, a *co-registration* procedure is necessary.

The co-registration consists in matching high-resolution (also known as T1-w) and DWI images defined in different coordinate systems. Open source libraries of National Library of Medicine *Insight Segmentation and Registration Toolkit* are adapted for the co-registration procedure, while Tcl/Tk code and VTK open source libraries are chosen for the graphic interface. Digital Imaging and Communications in Medicine format (DICOM) tag parameters necessary for the co-registration are: Image Origin, Image Spacing, Patient Image Orientation, and Frame of Reference.

Assuming the same anatomy topology for different studies, a Mutual Information technique based on Mattes algorithm is applied. An in-house software for multimodal registration was developed. The program 3D Slicer,¹ a free open source software for visualization and image computing, is employed for the graphic interface. The process was performed in several steps.

The source DWI study (*moving image*, see Fig. 10.3) is aligned through a roto-translational matrix with the T1-w data (*fixed image*); the two studies are acquired in straight succession with the same MR unit without patient repositioning; the parameters related to algorithm implementation are automatically defined; then, by applying a multi-resolution pyramid, we are able to reach a registration within eight iterations avoiding local minimal solution.

The results of the registration are visually inspected in a checkerboard, where each block alternately displayed data from each study, verifying alignment of

¹<http://www.slicer.org/>.

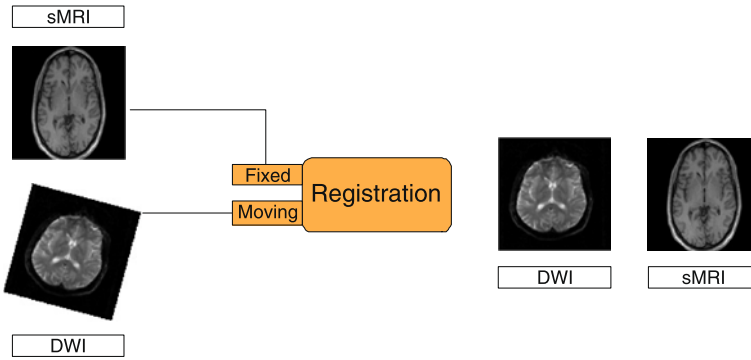


Fig. 10.3 Registration of sMRI to DWI

anatomical landmarks (ventricles, etc.) for confirmation. This procedure is needed because sMRI images have better resolution and the anatomy can better be seen for manual ROI segmentation. We use this procedure to extract ADC values for each of the ROIs instead of the whole image. Once the co-registration is carried out, a direct voxel-by-voxel comparison between the two data modalities becomes feasible and therefore any joint feature can be extracted.

10.4 Region Selection

The brain is a complex organ composed of different kinds of tissues related to different physiological properties of brain matter. Moreover, the brain can be segmented into well defined anatomical structures which are associated to specific functions of the brain. In order to improve the search of brain abnormalities, it is important to take into account of such kind of brain subdivisions. Two main paradigms are in general defined: (i) White matter (WM), Gray matter (GM), and Cerebrospinal Fluid (CSF) segmentation, and (ii) brain parcellation.

10.4.1 WM–GM–CSF Segmentation

WM–GM–CSF segmentation aims at decomposing the brain into its main kinds of tissues (see Fig. 10.4). In particular, white matter encloses mainly the axons which connect different parts of the brain, while gray matter contains neural cell bodies. Cerebrospinal fluid is a clear, colorless bodily fluid that occupies the ventricular system around and inside the brain and sulci.

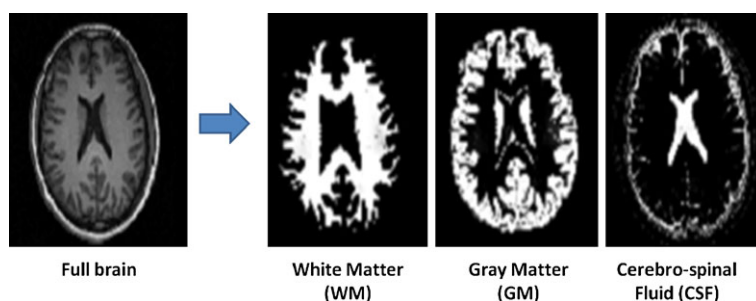


Fig. 10.4 Example of brain segmentation among White Matter (WM), Gray Matter (GM), and Cerebrospinal Fluid (CSF)

10.4.2 Brain Parcellation

The raw images acquired using a 1.5 T MRI machine have 144 slices and 384×512 resolution. These images are then transferred to PC workstations in order to be processed for ROI *tracing* which we adapted. Based on manual identification of landmarks, these slices are resampled and realigned by the medical personnel using the Brains2² software. The same software is used to manually trace the ROIs by drawing contours enclosing the intended region. This was carried out by a trained expert following a specific protocol for each ROI [7] without knowledge of the class labels. There are methods which automatically segment the ROIs, but their accuracy is lower than the manual methods so manual segmentation was preferred. The ROIs traced are 7 pairs (for the left and the right hemisphere, respectively) of disconnected image areas:

- Amygdala (*lamyg* and *ramyg*, in short)
- Dorso-lateral PreFrontal Cortex (*ldlpfc* and *rdlpfc*)
- Entorhinal Cortex (*lec* and *rec*)
- Heschl's Gyrus (*lhg* and *rhg*)
- Hippocampus (*lhippo* and *rhippo*)
- Superior Temporal Gyrus (*lstg* and *rstg*)
- Thalamus (*lthal* and *rthal*)

We select these ROIs because they have consistently been found to be impaired in schizophrenia and in a recent work, some of them have been found to support a specific altered neural network [21]. The Inter Rater Reliability (IRR) values for each brain hemisphere and ROI can be seen in Table 10.1 which shows us the reliability of the segmentation. Higher value means the segmentation is more reliable.

Additionally, another important ROI that is traced is the *intracranial volume* (ICV), that is the volume occupied by the brain in the cranial cavity leaving out the brainstem and the cerebellum. This information is extremely useful for normalizing volume values against differing overall brain sizes.

²<http://www.psychiatry.uiowa.edu/mhcr/IPLpages/BRAINS.htm>.

Table 10.1 IRR values for ROI segmentation

ROI	left	right
<i>amyg</i>	0.91	0.98
<i>dlpfc</i>	0.93	0.98
<i>ec</i>	0.94	0.96
<i>hg</i>	0.96	0.98
<i>hippo</i>	0.96	0.96
<i>stg</i>	0.93	0.99
<i>thal</i>	0.95	0.96

10.5 Data Description

In this section, we show how we describe the data to be used in classification.

10.5.1 Standard Features

In order to encode useful information in a compact representation, data descriptors are employed. The overall idea consists of representing the brain with a signature which summarizes brain characteristics, and using such signature for comparison purposes. We exploited several kinds of brain characteristics; each of them focusing on a specific aspect of the brain. In particular, we have employed histogram of image *intensities* to encode tissue characteristics, and *geometric* features to concentrate the analysis on shape properties of brain structures. We highlight that, according to standard feature-based approach, such descriptors could be directly used for brain classification. Since we aim at going beyond features in this book, we have exploited the new paradigm to deal with such brain characteristics by proposing new approaches for data description (as we will explain in Sects. 10.5.2 and 10.5.3).

In the following, we introduce (i) Intensity Histograms of sMRI, (ii) Histograms of Apparent Diffusion Coefficient values, (iii) basic geometric shape descriptors, and (iv) spectral shape descriptor.

10.5.1.1 Intensity Histograms of Structural MRI Images

From MRI data we compute scaled histograms of image intensities. In particular, we compute a histogram for each ROI. A major disadvantage of MRI compared to other imaging techniques is the fact that its intensities are not standardized. Even MR images taken for the same patient on the same scanner with the same protocol at different times may differ in content due to a variety of machine-dependent reasons, therefore, image intensities do not have a fixed meaning [54]. This implies a significant effect on the accuracy and precision of the following image processing, analysis, segmentation, and registration methods relying on intensity similarity.

A successful technique used to calibrate MR signal characteristics at the time of acquisition employs *phantoms* [29], by placing physical objects with known attributes within the scanning frame. Unfortunately, this technique is not always exploited, which is our present case. Alternatively, it is possible to apply bias correction (using software like SPM³ or FSL⁴) for the image intensities, and apply intensity rescaling afterwards. Here, we rescale intensities based on landmark matching from the ICV histograms [54] because it is easier to identify landmarks on the histograms that match the canonical subdivision of intracranial tissue into white matter, gray matter and cerebrospinal fluid. We select a rescaling mapping that conserves most of the signal in the gray matter—white matter area, corresponding to the two highest bumps in the range 60–90, since ROIs primarily contain those kinds of tissue.

10.5.1.2 Histograms of Apparent Diffusion Coefficient values

Although we don't have manually segmented ROIs for DWI images, we used a co-registration procedure to segment DWI images into ROIs. For this purpose, every subject's DWI image was registered into the corresponding structural MRI image. Then Apparent Diffusion Coefficient (ADC) values are calculated using these images. We form the histograms of ADC values and use them in our experiments. Since the ADC values are already normalized, we don't need to do another step of normalization on ADC histograms.

10.5.1.3 Basic Geometric Shape Descriptors

From the set of 2D ROIs of the shapes (slices) the 3D surface is computed as triangle mesh using marching cubes. A minimal smoothing operation is applied to remove noise and voxelization effect. We encode geometric properties of the surface using the *Shape Index* [44], which is defined as:

$$si = -\frac{2}{\pi} \arctan\left(\frac{k_1 + k_2}{k_1 - k_2}\right), \quad k_1 > k_2,$$

where k_1, k_2 are the principal curvatures of a generic surface point. The Shape Index varies in $[-1, 1]$ and provides a local categorization of the shape into primitive forms such as spherical cap and cup, rut, ridge, trough, or saddle [44]. Shape index is pose and scale invariant [44] and it has already been successfully employed in biomedical domain [5]. The shape index is computed at each vertex of the extracted mesh. Then, all the values are quantized and a histogram of occurrences is computed. Such histograms represent the descriptor of a given subject and it basically encodes the brain local geometry of a subject, disregarding the spatial relationships.

³<http://www.fil.ion.ucl.ac.uk/spm/>.

⁴<http://www.fmrib.ox.ac.uk/fsl/>.

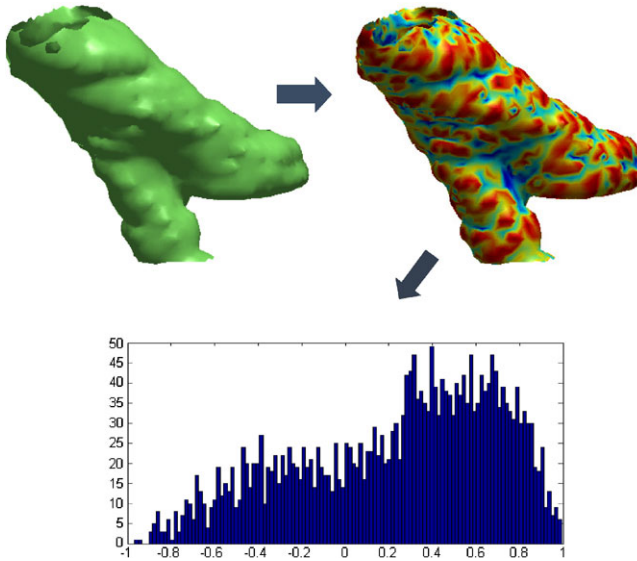


Fig. 10.5 Geometric feature extraction: 3D surface of the thalamus (*left*), the surface colored according with Shape Index values (*right*), and the histogram of Shape Index occurrences (*bottom*)

Figure 10.5 shows the 3D surface of the left-Thalamus (left), the surface colored according with Shape Index values (right), and the histogram of Shape Index occurrences (bottom). It is worth noting that convex regions (in blue) are clearly distinguished from concave regions (in red) by the Shape Index values. As a further step we also calculate the mean curvature using the same methodology:

$$m = \frac{k_1 + k_2}{2}.$$

10.5.1.4 Spectral Shape Descriptor

In this section, we describe a new shape descriptor, which is based on advanced *diffusion* geometry techniques. Considering a shape M as a compact Riemannian manifold [14], the heat diffusion on shape⁵ is defined by the *heat* equation:

$$\left(\Delta_M + \frac{\partial}{\partial t} \right) u(t, \mathbf{m}) = 0, \quad (10.1)$$

where u is the distribution of heat on the surface, $\mathbf{m} \in M$, Δ_M is the *Laplace–Beltrami* operator which, for compact spaces, has discrete eigendecomposition of

⁵In this section, we borrow the notation from [14, 73].

the form $\Delta_M \phi_i = \lambda_i \phi_i$. In this way, the *heat kernel* has the following eigendecomposition:

$$h_t(\mathbf{m}, \mathbf{m}') = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(\mathbf{m}) \phi_i(\mathbf{m}'), \quad (10.2)$$

where λ_i and ϕ_i are the i th eigenvalue and the i th eigenfunction of the Laplace–Beltrami operator, respectively. The heat kernel $h_t(\mathbf{m}, \mathbf{m}')$ is the solution of the heat equation with initial point heat source in \mathbf{m} at time $t = 0$, and heat value in ending point $\mathbf{m}' \in M$ after time t . The heat kernel is *isometrically invariant*, it is *informative*, and *stable* [73].

In the case of volumetric representations, the volume is sampled by a regular Cartesian grid composed by voxels, which allows the use of standard Laplacian in \mathbb{R}^3 as the Laplace–Beltrami operator. We use finite differences to evaluate the second derivative in each direction of the volume. The heat kernel on volumes is invariant to volume isometries, in which shortest paths between points inside the shape do not change. Note that in real applications, exact volume isometries are limited to the set of rigid transformations [61], however, also non-rigid deformations can faithfully be modeled as approximated volume isometries in practice. It is also worth noting that, as observed in [61, 73], for small t the autodiffusion heat kernel $h_t(\mathbf{m}, \mathbf{m})$ of a point \mathbf{m} with itself is directly related to the *scalar curvature* $s(\mathbf{m})$ [61]. More formally,

$$h_t(\mathbf{m}, \mathbf{m}) = (4\pi t)^{-3/2} \left(1 + \frac{1}{6} s(\mathbf{m}) \right). \quad (10.3)$$

In practice, Eq. (10.3) states that the heat tends to diffuse slower at points with positive curvature, and vice-versa. This gives an intuitive explanation about the geometric properties of $h_t(\mathbf{m}, \mathbf{m})$, and suggests the idea of using it to build a shape descriptor [73].

Global Heat Kernel Signature Once data are collected, a strategy to encode the most informative properties of the shape M can be devised. To this end, a global shape descriptor is proposed, which is inspired by the so-called *Heat Kernel Signature* (HKS) defined as:

$$\text{HKS}(x) = [h_{t_0}(x, x), \dots, h_{t_n}(x, x)], \quad (10.4)$$

where x is a point of the shape (i.e., a vertex of a mesh or a voxel) and (t_0, t_1, \dots, t_n) are n time values. To extend this approach to the whole shape, we introduce the following global shape descriptor:

$$\text{GHKS}(M) = [\text{hist}(H_{t_0}(M)), \dots, \text{hist}(H_{t_n}(M))], \quad (10.5)$$

where $H_{t_i}(M) = \{h_{t_i}(x, x), \forall x \in M\}$, and $\text{hist}(\cdot)$ is the histogram operator. Note that our approach combines the advantages of [14, 61] since it encodes the distribution of local heat kernel values and it works at multiscales. GHKS allows for shape

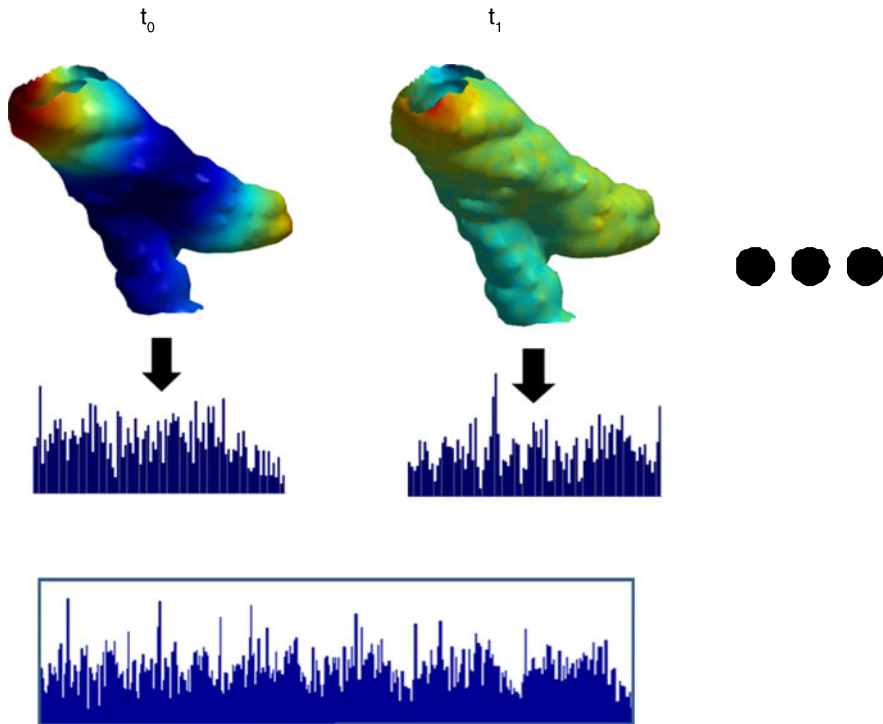


Fig. 10.6 GHKS: Each point of the shape is colored according to $h_{t_i}(x, x)$. Such values are collected into a histogram for each scale t_i . Finally, histograms are concatenated leading to the global signature

comparisons using minimal shape preprocessing, in particular, no registration, mapping, or remeshing is necessary. GHKS is robust to noise since it implicitly employs surface smoothing by neglecting higher frequencies of the shape. Finally, GHKS is able to encode isometric invariance properties of the shape [73] which are crucial to deal with shape deformations. Figure 10.6 shows a scheme of the proposed descriptor. Each point of the shape is colored according to $h_{t_i}(x, x)$. Such values are collected into a histogram for each scale t_i . Finally, histograms are concatenated leading to the global signature.

10.5.2 Descriptors on Dissimilarity Space

In this section, we describe data descriptors generated by employing similarity-based approach. In general, similarity-based approach aims at exploiting the discriminative properties of similarity measures per se, as opposed to standard feature-based approach. In fact, the similarity-based paradigm differs from typical pattern

recognition approaches where objects to be classified are represented by feature vectors. Devising pattern recognition techniques starting from similarity measures is a real challenge, and the main idea of this book. Among the different proposed techniques, in this work we investigated the use of the dissimilarity-based representation paradigm, introduced by Pekalska and Duin [55] and described in Chap. 2. Within this approach, objects are described using pairwise (dis)similarities to a representation set of objects. This offers the analyst a different way to express application background knowledge as compared to features. In a second step, the dissimilarity representation is transformed into a vector space in which traditional statistical classifiers can be employed. Unlike the related kernel approach, whose application is often restrained by technicalities like fulfilling Mercer's condition, basically any dissimilarity measure can be used.

Similarity-based approach is more versatile in dealing with different data representations (i.e., images, MRI volume, graphs, DNA strings, and so on) since for each kind of data the most suitable (dis)similarity measure can be chosen. In the following, we introduce several dissimilarity measures and define the dissimilarity space.

10.5.2.1 Dissimilarity Measures

Up to this level of the pipeline, data are characterized by histograms. Therefore, we can use histograms to devise similarity measures to be employed in the dissimilarity-based representation scheme. There are various dissimilarity measures that can be applied to measure the dissimilarities between histograms [18, 68]. Moreover, histograms can be converted to pdfs and dissimilarity measures between two discrete distributions can be used as well. All in all, we decided to study measures below.

Given two histograms S and M with n bins, we define the number of elements in S and M as $|S|$ and $|M|$, respectively.

Histogram Intersection It measures the number of intersecting values in each bin [74]:

$$\text{sim}(S, M) = \frac{\sum_{i=1}^n \min(S_i, M_i)}{\min(|S|, |M|)}.$$

Since this is a similarity measure, we convert it to a dissimilarity using $D = \min(|M|, |S|) \times (1 - \text{sim}(S, M))$.

Diffusion Distance In diffusion distance [50], the distance between two histograms is defined as a temperature field $T(x, t)$ with $T(x, 0) = S(x) - M(x)$. Using the heat diffusion equation $\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}$ which has a unique solution $T(x, t) = T(x, 0) \times \phi(x, t)$ where $\phi(x, t) = \frac{1}{(2\phi)^{1/2t}} \exp\{-\frac{x^2}{2t}\}$, we can compute D as:

$$D = \int_0^r k(|T(x, t)|) dt,$$

where $k(\cdot)$ is the L_1 -norm.

χ^2 Distance This metric is based on the χ^2 test for testing the similarity between histograms. It is defined as

$$D = \sum_{i=1}^n \frac{(S_i - M_i)^2}{S_i + M_i}.$$

It is a standard measure for histograms.

Earth Mover's Distance This distance was originally proposed by Rubner et al. [65]. It is basically defined as the cost to transform one distribution into another. It is calculated using linear optimization by defining the problem as a transportation problem. For 1D histograms, it reduces to a simple calculation [18] which was implemented in this study.

$$C_i = \left| \sum_{j=1}^i (S_j - M_j) \right|, \quad D = \sum_{i=1}^n C_i.$$

Similarly, we have considered the following dissimilarities between pdfs:

Bhattacharyya It is used to measure the similarity of discrete probability distributions p and q . It is defined as

$$D(p, q) = -\log \text{BC}(p, q),$$

where

$$\text{BC}(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}.$$

Kullback–Leibler (KL) Divergence Kullback–Leibler divergence is defined as

$$D(p, q) = \sum_{i=1}^n q_i \log \frac{q_i}{p_i}.$$

This measure is not a distance metric but a relative entropy since $D(p, q) \neq D(q, p)$, i.e., the dissimilarity matrix is not symmetric. There are various ways to symmetrize this dissimilarity. We simply used $D = D(p, q) + D(q, p)$ and the so-called Jensen–Shannon divergence: $D = \frac{1}{2}D(p, r) + \frac{1}{2}D(q, r)$, where r is the average of p and q .

10.5.2.2 Dissimilarity Space

Suppose that we have n objects and we have a dissimilarity matrix D of size $n \times n$. And suppose that the dissimilarity between two objects o and \hat{o} are denoted by $D(o, \hat{o})$. There are several ways to transform an $n \times n$ dissimilarity matrix D with elements $D(o, \hat{o})$ into a vector space with objects represented by vectors $X = \{x'_1, \dots, x'_o, \dots, x'_\hat{o}, \dots, x'_n\}$ [55]. Classical scaling (for proper Euclidean dissimilarities) and pseudo-Euclidean embedding (for arbitrary symmetric dissimilarities) yield vector spaces in which vector dissimilarities can be defined that produce the given dissimilarities D . As almost all dissimilarity measures studied here are non-Euclidean, classification procedures for these pseudo-Euclidean spaces are ill-defined, as, for instance, the corresponding kernels are indefinite.

A more general solution is to work directly in the *dissimilarity space*. It postulates an Euclidean vector space using the given dissimilarities to a representation set as features. As opposed to the previously mentioned techniques, it is not true anymore that dissimilarities in this space are identical to the given dissimilarities, but this is an advantage in case it is doubtful whether they really represent dissimilarities between the physical objects. As this holds in our case we constructed such a dissimilarity space using all available objects by taking X equal to D . In the dissimilarity space, basically any traditional classifier can be used. The number of dimensions, however, equals the number of objects in the representation set. Many classifiers will need dimension reduction techniques or regularization to work properly in this space.

A further refining of the scheme can be obtained by considering at the same time different dissimilarities (we have many, linked to different modalities, different zones of the brain or different methods to compute them), trying to combine them in a single dissimilarity space. Combined dissimilarity spaces can be constructed by combining dissimilarity representations. As in normal classifier combination [42, 45], a simple and often effective way is using an (weighted) average of the given dissimilarity measures:

$$D_{\text{combined}} = \frac{\sum \alpha_i D_i}{\sum \alpha_i}. \quad (10.6)$$

It is related to the sum-rule in the area of combining classifiers. The weights can be optimized for some overall performance criterion, or determined from the properties of the dissimilarity matrix D_i itself, e.g., its maximum or average dissimilarity. Here, we used equal weights while combining multiple dissimilarity matrices and all the dissimilarity matrices are scaled such that the average dissimilarity is one, i.e.,

$$\frac{D(o, \hat{o})}{\frac{1}{n(n-1)} \sum_{o, \hat{o}} D(o, \hat{o})} = 1. \quad (10.7)$$

This is done to assure that the results are comparable over the dissimilarities as we deal with dissimilarity data in various ranges and scales. Such scaled dissimilarities

are denoted as \tilde{D} . In addition, we assume here that the dissimilarities are symmetric. So, every dissimilarity $\tilde{D}(i, j)$ has been transformed by

$$\tilde{D}(i, j) := \frac{\tilde{D}(i, j) + \tilde{D}(j, i)}{2}. \quad (10.8)$$

10.5.3 Descriptors by Generative Embeddings

In this section, we define a new class of data descriptors based on generative embedding procedure (see Chap. 4). The overall idea consists in fitting a generative model on training data and using the generative process to define new data-dependent representations. Then, such representations can be plugged into a standard discriminative classifier for classification purposes. This approach is pursued by hybrid architectures of discriminative and generative classifiers which is currently one of the most interesting, useful, and difficult challenges for Machine Learning. The underlying motivation is the proved complementariness of discriminative and generative estimations: asymptotically (in the number of labeled training examples), classification error of discriminative methods is lower than that of generative ones [53]. On the other side, generative counterparts are effective with less, possibly unlabeled, data; further, they provide intuitive mappings among structure of the model and data features. Among these hybrid generative–discriminative methods, “generative embeddings” (also called generative score space) have grown in importance in the literature [11, 12, 39, 49, 56, 71, 72, 79].

Generative score space framework consists of two steps: first, one or a set of generative models are learned from the data; then a score (namely a vector of features) is extracted from it, to be used in a discriminative scenario. The idea is to extract fixed dimension feature vectors from observations by subsuming the process of data generation, projecting them in highly informative spaces called score spaces. In this way, standard discriminative classifiers such as support vector machines, or logistic regressors have achieved higher performances than a solely generative or discriminative approach.

Using the notation of [56, 71], such spaces can be built from data by mapping each observation x to the fixed-length score vector $\varphi_{\hat{F}}^f(x)$,

$$\varphi_{\hat{F}}^f(x) = \varphi_{\hat{F}} f(\{P_i(x | \theta_i)\}), \quad (10.9)$$

where $P_i(x | \theta_i)$ represents the family of generative models learned from the data, f is the function of the set of probability densities under the different models, and \hat{F} is some operator applied to it. In general, the generative score-space approaches help to distill the relationship between a model’s parameters θ and the particular data sample.

Generative score-space approaches are strictly linked to generative kernels family, namely kernels which compute similarity between points through a generative

model—the most famous example being the Fisher Kernel [39]. Typically, a generative kernel is obtained by defining a similarity measure in the score space, e.g., the inner product. Score spaces are also called model dependent feature extractors, since they extract features from a generative model.

In order to apply the generative embedding scheme to the MRI data, we should define a generative model able to explain and model what we have. Here, we adapted as generative model the probabilistic Latent Semantic Analysis (pLSA—[38]), a tool widely applied in the linguistic and in the computer vision community.

In the following, we first describe the basics of the pLSA, then explain how this model can be applied to our problem, finally describing the kind of generative embeddings we exploited.

10.5.3.1 Probabilistic Latent Semantic Analysis

In the Probabilistic Latent Semantic Analysis (PLSA—[38]), the input is a set of D documents, each one containing a set of words taken from a vocabulary of cardinality W . The documents are summarized by an occurrence matrix of size $W \times D$, where $n(w_j, d_i)$ indicates the number of occurrences of the word w_j in the document d_i . In PLSA, the presence of a word w_j in the document d_i is mediated by a latent *topic* variable, $z \in Z = \{z_1, \dots, z_Z\}$, also called *aspect* class, i.e.,

$$P(w_j, d_i) = \sum_{k=1}^Z P(z_k) P(w_j | z_k) P(d_i | z_k). \quad (10.10)$$

In practice, the topic z_k is a probabilistic co-occurrence of words encoded by the distribution $\beta(w) = p(w | z_k)$, $w = \{w_1, \dots, w_N\}$, and each document d_i is compactly (usually, $Z < W$) modeled as a probability distribution over the topics, i.e., $p(z | d_i)$, $z = \{z_1, \dots, z_Z\}$ (note that this formulation, derived from $p(d_i | z)$, provides an immediate interpretation).

The hidden distributions of the model, $p(w | z)$, $p(d | z)$ and $p(z)$, are learned using Expectation Maximization (EM) [26], maximizing the model data log-likelihood \mathcal{L} :

$$\mathcal{L} = \prod_{j=1}^W \prod_{i=1}^D n(w_j, d_i) \log(p(w_j, d_i)). \quad (10.11)$$

The E-step computes the posterior over the topics, $p(z | w, d)$, and the M-step updates the hidden distributions. Even if pLSA is a model for documents, it has been largely applied in other contexts, especially in computer vision [12, 23] but also in the medical informatics domain [9, 10, 17].

The idea under its application to the MRI domain is straightforward. In particular, we can assume that a given brain (or the particular ROI) represents the documents d , whereas the words w_j are the local features previously described. With such a point of view, the extracted histograms represent the counting vectors, able to describe

how much a visual feature (namely a word) is present in a given image (namely a document).

10.5.3.2 PLSA-Based Generative Embeddings

Once a generative model is trained, different spaces can be obtained. Generally speaking, we can divide them into two families: parameter-based and hidden variable-based. The former class derives the features on the basis of differential operations linked to the parameters of the probabilistic model, while the latter seeks to derive feature maps on the basis of the log-likelihood function of a model, focusing on the random variables rather than on the parameters.

Parameter-Based Score Space These methods derive the features on the basis of differential operations linked to the parameters of the probabilistic model.

The Fisher Score The Fisher score for the PLSA model has been introduced in [37], starting from the asymmetric formulation of PLSA. In this case, the log-probability of a document d_i is defined by

$$\begin{aligned} l(d_i) &= \frac{\log P(d_i, w)}{\sum_m n(d_i, w_m)} \\ &= \sum_{j=1}^W \hat{P}(w_j | d_i) \log \sum_{k=1}^Z P(w_j | z_k) P(d_i | z_k) P(z_k), \end{aligned} \quad (10.12)$$

where $\hat{P}(w_j | d_i) \equiv n(d_i, w_j) / \sum_m n(d_i, w_m)$ and where $l(d_i)$ represents the probability of all the word occurrences in d_i normalized by document length.

Differentiating Eq. (10.12) with respect to $P(z)$ and $P(w | z)$, the pLSA model parameters, we can compute the score. The samples are mapped in a space of dimension $W \times Z + Z$. The Fisher kernel is defined as the inner product in this space. We will refer to it as FSH.

TOP Kernel Scores Top Kernel and the tangent vector of posterior log-odds score space were introduced in [79]. Whereas the Fisher score is calculated from the marginal log-likelihood, TOP kernel is derived from Tangent vectors Of Posterior log-odds. Therefore, the two score spaces have the same score function (i.e., the gradient) but different score arguments, which, for TOP kernel $f(p(x | \theta)) = \log P(c = +1 | x, \theta) - \log P(c = -1 | x, \theta)$ where c is the class label. We will refer to it as TOP.

Log-Likelihood Ratio Score Space The log-likelihood ratio score space is introduced in [72]. Its dimensions are similar as for the Fisher score, except that the procedure is repeated for each class: a model θ_c per class is learned and the gradient is applied to each $\log p(x | \theta_c)$. The dimensionality of the resulting space is $C \times$ the dimensionality of the original Fisher score. We will refer to it as LLR.

Random Variable Based Methods These methods, starting from considerations in [56], seek to derive feature maps on the basis of the log-likelihood function of a model, focusing on the random variables rather than on the parameters in their derivation (as done in the parameter-based score spaces).

Free Energy Score Space (FESS) In the Free Energy Score Space [56], the score function is the free energy while the score argument is its unique decomposition into the terms that compose it.⁶ Free energy is a popular score function representing a lower bound of the negative log-likelihood of the visible variables used in the variational learning. For pLSA it is defined by the following equation:

$$\begin{aligned} \mathcal{F}(d_i) = & \sum_w n(d_i, w) \sum_z P(z | d, w) \log P(z | d, w) \\ & - \sum_w n(d_i, w) \sum_z P(z | d, w) \log P(d, w | z) P(z), \quad (10.13) \end{aligned}$$

where the first term represents the entropy of the posterior distribution and the second term is the cross-entropy. For further details on the free energy and on variational learning, see [31]; for the pLSA's free energy, see [38].

For pLSA this results in a space of dimension equal to $C \times 2 \times Z \times W$. In [56], the authors point out that, if the dimensionality is too high, some of the sums can be carried out to reduce the dimensionality of the score vector before learning the weights. The choice of the term to optimize is intuitive but guided by the particular application. In our case, as previously done in [49, 57], we perform the sums over the word indices, optimizing the contributing topics. The resulting score space has dimension equal to $C \times 2 \times Z$; we will refer to this score space as FESS.

Posterior Divergence Posterior Divergence score space is described in [49]. Like FESS, it takes into account how well a sample fits the model (cross-entropy terms in FESS) and how uncertain the fitting is (entropy terms in FESS, Eq. (10.13)) but it also assesses the change in model parameters brought by the input sample, i.e., how much a sample affects the model. These three measures are not simply stacked together, but they are derived from the incremental EM algorithm which, in the E-step only, looks at one or a few selected samples to update the model at each iteration. Details on posterior divergence score vector for pLSA and on its relationships with FESS case can be found in [49]. We will refer to this score space as PD.

Classifying with the Mixture of Topics of a Document Very recently, pLSA has been used as a dimensionality reduction method in several fields, like computer vision, bioinformatics, and medicine [10, 12, 17]. The idea is to learn a pLSA model to capture the co-occurrence between visual words [12, 17], or gene expressions [10], which represent the (usually) high-dimensional data description; co-occurrences are

⁶This is true once a family for the posterior distribution is given. See the original paper for details.

captured by the topics. Subsequently, the classification is performed using the topic distribution that defines a document as sample descriptor.

Since we are extracting features from a generative model, we are defining a score space which is the Z -dimensional simplex. In this case, the score argument f , a function of the generative model, is the topic distribution $P(z | d)$ (using Bayes' formula, one can easily derive $P(z | d)$ starting from $P(d | z)$), while the score function is the identity. We will refer to this score space as TPM.

In our experiments, for the two score spaces FESS and TPM, we include two versions. The first version is where we train one pLSA per class and concatenate the resulting feature vectors (we will refer these as FESS-1 and TPM-1), the second one is where we train a pLSA for the whole data without looking at the class label (we will refer these as FESS-2 and TPM-2). All in all, we have eight different score spaces: TPM-1, TPM-2, FESS-1, FESS-2, LLR, FSH, TOP, PD.

10.6 Classification

After data description step, a learning-by-example procedure is employed for brain classification in order to discriminate between healthy subjects and patients affected by schizophrenia. As a basic approach, when a single source of information is considered, a standard single classifier can be employed. From the medical point of view, this means that the relevance of a particular source of information is considered to characterize the brain abnormality. On the other hand, when several factors can be the possible cause of the disease, a multi-source classification strategy may be employed. Here, we have exploited two paradigms: (i) multi-classification, and (ii) Multiple Kernel Learning (MKL).

10.6.1 Multi-classifier

It is a well-known fact that there is no single most accurate classification algorithm, so methods have been proposed to combine classifiers based on different assumptions [1, 45]. Classifier combination (also called ensemble construction) can be done at different levels and in different ways: (i) sensor fusion, (ii) representation fusion, (iii) algorithm fusion, (iv) decision fusion, and others. Each classifier ((algorithm/parameter set/data representation) triplet) makes a different assumption about the data and makes errors on different instances and by combining multiple classifiers; the overall error can be decreased. Classifiers' making different errors on different parts of the space is called "diversity" (in a broad definition), and to achieve diversity different (i) learning algorithms, (ii) hyperparameters, (iii) input features, and (iv) training sets have been used [45, 83] .

There are various methods to combine classifiers; the simple method is to use voting [42] (or take an average over the outputs) which corresponds to fixed rules

which we applied when the classifiers created posterior probability outputs, i.e., $P(C_k | \mathbf{x}, E) = \sum_{i=1}^L P(C_k | \mathbf{x}, M_i)$, where E denotes the ensemble, $P(C_k | \mathbf{x}, E)$ is the posterior of the ensemble for class C_k , L is the number of classifiers to combine, $M_i, i = 1, \dots, L$ are the individual classifiers to combine, and $P(C_k | \mathbf{x}, M_i)$ is the posterior of classifier M_i . Voting does not require any parameter to be optimized and is simple. Other methods such as weighted averaging or more advanced methods require the estimation of other parameters. In previous works [20, 80], we used single classifier and multi-classifier approaches to schizophrenia detection with correlation analysis which serve as a baseline for our dissimilarity based analysis.

10.6.2 Multiple Kernel Learning (MKL)

The main idea behind SVMs [84] is to transform the input feature space to another space (possibly with a greater dimension) where the classes are linearly separable. After training, the discriminant function of SVM becomes $f(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b$, where \mathbf{w} is the vector of weights, b is the threshold, and $\Phi(\cdot)$ is the mapping function. Using the dual formulation and the kernel trick, one does not have to define this mapping function explicitly and the discriminant function can be written as

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b,$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ is the kernel function that calculates a similarity metric between data instances. Selecting the kernel function is the most important issue in the training phase; it is generally handled by choosing the best-performing kernel function among a set of kernel functions on a separate validation set.

In recent years, MKL methods have been proposed [6, 46] (for a review see [35]), for learning a combination k_η of multiple kernels instead of selecting only one:

$$k_\eta(\mathbf{x}_i, \mathbf{x}_j; \eta) = f_\eta(\{k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)\}_{m=1}^P; \eta), \quad (10.14)$$

where the combination function f_η forms a single kernel from P base kernels using the parameters η . Different kernels correspond to different notions of similarity and instead of searching which works best, the MKL method does the picking for us, or may use a combination of kernels. MKL also allows us to combine different representations possibly coming from different sources or modalities.

There is significant work on the theory and application of MKL, and most of the proposed algorithms use a linear combination function such as convex sum or conic sum. Fixed rules use the combination function in (10.14) as a fixed function of the kernels, without any training. Once we calculate the combined kernel, we train a single kernel machine using this kernel. For example, we can obtain a valid kernel by taking the mean of the combined kernels.

Instead of using a fixed combination function, we can also have a function parameterized by a set of parameters and then we have a learning procedure to optimize these parameters as well. The simplest case is to parameterize the sum rule as a weighted sum:

$$k_{\eta}(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\eta}) = \sum_{m=1}^P \eta_m k_m(\mathbf{x}_i^m, \mathbf{x}_j^m)$$

with $\eta_m \in \mathbb{R}$. Different versions of this approach differ in the way they put restrictions on the kernel weights [6, 46, 60]. For example, we can use arbitrary weights (i.e., linear combination), nonnegative kernel weights (i.e., conic combination), or weights on a simplex (i.e., convex combination). A linear combination may be restrictive, and nonlinear combinations are also possible [22, 34, 36, 48].

10.7 Case Study 1: Brain Classification on Dissimilarity Space

After presenting all the possible choices we made in the different parts of the pipeline, let us present some concrete systems. In particular, here we describe a method based on the dissimilarity-representation paradigm, whereas in Sect. 10.8 a method based on the generative embeddings is presented.

Concerning the taxonomies presented in Fig. 10.1, here we are using both sMRI and DWI approaches (namely a multimodal scheme), starting from the brain parcellation, employing the dissimilarity-based description and dissimilarity combination by classifying with a single classifier.

In particular, in these experiments [82], we use a 114 subject subset of the original data set (59 patients, 55 healthy controls). We used the intensity histograms from sMRI images (SMRI), ADC histograms from DWI images (ADC), and two geometric shape descriptors, shape index (SH) and mean curvature (MCUR). We used all the ROIs and used the dissimilarity space by computing the dissimilarities between the histograms and their corresponding pdfs. In summary, for each ROI and representation we use the following 13 measures:

- *hist-euclid*—Euclidean distance between histograms
- *hist-l1*— L_1 -distance between histograms
- *hist-intersect*—Intersection between histograms
- *hist-diffusion*—Diffusion distance between histograms
- *hist-chi*— χ^2 -distance between histograms
- *hist-emd*—Earth mover’s distance between histograms
- *pdf-euclid*—Euclidean distance between pdfs
- *pdf-l1*— L_1 -distance between pdfs
- *pdf-emd*—Earth mover’s distance between pdfs
- *pdf-bs*—Bhattacharyya distance between pdfs
- *pdf-kl*—Symmetrized KL divergence between pdfs
- *pdf-kl-orig*—Original, asymmetric KL divergence

- *pdf-js*—Jensen–Shannon divergence between pdfs

All in all, there are 14 ROIs and 13 different dissimilarity measures per modality, which yields a total of 182×4 dissimilarity matrices. In addition to these, we propose to merge the different dissimilarity matrices into one overall dissimilarity matrix per modality, potentially exploiting complementary information useful to improve the classification accuracy. We also test the accuracy of these combinations against combining classifiers in the original feature space (histograms and pdfs for each of the four modalities). For each test we evaluated the leave-one-out error. All differences in accuracy reported in this case study are significant at $p = 0.05$ using the paired t -test. The dissimilarity spaces have been built in a transductive way by using all available subjects for dissimilarity (of course, labels are ignored in this phase). Three classifiers are considered to compare the performances:

- Linear SVM classifier on the original feature space (called *svm*)
- The 1-Nearest Neighbor (NN) rule on the dissimilarity matrices (called *1nn*)
- Linear SVM classifier on the dissimilarity space (called *sv0*)

The linear SVM in dissimilarity space avoids complications that could arise from the dissimilarity measures being non-Euclidean because we treat the dissimilarities as features in this new space. While combining dissimilarities, we use for α_i in (Eq. (10.6)) the reciprocal of the number of dissimilarity matrices to be combined [47]. On the original feature space, the SVM classifiers produce posterior probability outputs, and these outputs are combined using the SUM rule [42]. So, on the original feature space, we combine after training the classifiers, whereas on the dissimilarity space, we combine before we do classification. The experiments are carried out using the Matlab package PRTools [28]. We designed three experiments to show the improvements of dissimilarity-based pattern recognition techniques and combination of dissimilarities using multiple ROIs and modalities:

1. ROI-based classification—For each modality, we report the highest accuracy that a classifier reaches without combination (on the original feature space and on the dissimilarity space). We use these results as a baseline for comparison.
2. Multi-ROI classification—In this set of experiments, for each modality, we fix the dissimilarity measure and combine all ROIs using this dissimilarity measure.
3. Multimodal classification—In this experiment, we go one step further to combine information coming from different sources by combining different dissimilarity matrices from different modalities.

We note that, throughout this section, we will use the following notation: every dissimilarity representation will be referred to as *MODALITY-roi-dissimilarity* (i.e., *SMRI-ldlpdfc-pdf-js* shows the dissimilarity matrix for the structural MRI of ROI *ldlpdfc* using the dissimilarity measure of Jensen–Shannon divergence). The modality, ROI, or the dissimilarity measure will be omitted when it's clear from the context.

Table 10.2 Best accuracies for each dissimilarity measure combining all ROIs on all modalities

Modality	SMRI			ADC			SH			MCUR		
	<i>svm</i>	<i>Inn</i>	<i>sv0</i>	<i>svm</i>	<i>Inn</i>	<i>sv0</i>	<i>svm</i>	<i>Inn</i>	<i>sv0</i>	<i>svm</i>	<i>Inn</i>	<i>sv0</i>
<i>lamyg</i>	68.42	64.04	78.07	64.04	57.89	62.28	45.61	68.42	64.91	43.86	61.40	57.02
<i>ramyg</i>	54.39	65.79	66.67	54.39	56.14	59.65	49.12	53.51	57.89	46.49	58.77	57.89
<i>ldlpfc</i>	60.53	62.28	76.32	56.14	51.75	61.40	52.63	62.28	57.89	49.12	53.51	53.51
<i>rdlpfc</i>	64.04	57.89	68.42	54.39	56.14	65.79	54.39	59.65	60.53	56.14	57.02	60.53
<i>lec</i>	64.04	56.14	64.91	53.51	62.28	61.40	46.49	51.75	54.39	47.37	53.51	53.51
<i>rec</i>	64.91	65.79	71.05	64.04	58.77	60.53	52.63	60.53	57.02	52.63	65.79	63.16
<i>lhg</i>	51.75	60.53	63.16	55.26	61.40	54.39	47.37	54.39	65.79	61.40	56.14	62.28
<i>rhg</i>	50.00	63.16	59.65	50.88	58.77	58.77	43.86	55.26	55.26	57.89	64.04	61.40
<i>lhippo</i>	63.16	60.53	72.81	52.63	57.02	59.65	55.26	50.00	57.02	53.51	58.77	62.28
<i>rhippo</i>	60.53	64.04	66.67	48.25	55.26	52.63	47.37	59.65	57.02	54.39	55.26	58.77
<i>lstg</i>	55.26	59.65	69.30	54.39	56.14	60.53	40.35	58.77	52.63	50.00	50.00	51.75
<i>rstg</i>	63.16	57.02	64.91	64.04	60.53	70.18	53.51	55.26	57.89	41.23	63.16	57.89
<i>lthal</i>	58.77	64.91	67.54	57.89	57.89	58.77	46.49	53.51	57.89	53.51	56.14	58.77
<i>rthal</i>	64.91	59.65	64.04	53.51	60.53	59.65	54.39	57.89	59.65	48.25	54.39	67.54

10.7.1 ROI-Based Classification

We evaluate the classification accuracies for each of the original dissimilarity matrices. Table 10.2 summarizes the results for all modalities. For each ROI the best performance is reported with respect to various dissimilarity measures (for details see [82]). The first column for each modality reports the accuracy estimates for *svm* using the original feature space (histograms and pdfs). The second column reports the maximum accuracy of *Inn* on different dissimilarity measures. The third column reports the leave-one-out accuracy estimates of the linear SVM in dissimilarity space. For SMRI, it shows clearly the improvements of our dissimilarity-based approach. Except for two ROIs (*rhg* and *rthal*), SVM classifier in the dissimilarity space is always better than classifiers in the standard space. While the best accuracy of standard approaches is 68.42 %, we can reach 78.07 % accuracy on dissimilarity space and the dissimilarity space accuracies are more stable.

For the other modalities, the results are similar. From the results for the ADC values extracted from DWI images, we can again see that when we switch to dissimilarity based classification, we get better accuracies (either *Inn* or *sv0*) except for two ROIs (*lamyg* and *rec*). We can again see that with a single ROI and dissimilarity measure, we can reach 70.18 % whereas the highest accuracy we can obtain in the original space is 64.04 %. The same pattern can be observed when we investigate SH and MCUR. Also in these modalities, the best accuracy can be achieved using dissimilarities. We can see that on SH, we reach 68.42 % using *Inn* and 65.79 % using *sv0*. The best accuracy using the features on the original space is 55.26 %. Also on MCUR, best accuracy is reached using *sv0*.

Table 10.3 Best accuracies for each dissimilarity measure combining all ROIs on all modalities

Measure	SMRI		ADC		SH		MCUR	
	<i>Inn</i>	<i>sv0</i>	<i>Inn</i>	<i>sv0</i>	<i>Inn</i>	<i>sv0</i>	<i>Inn</i>	<i>sv0</i>
<i>hist-l2</i>	62.28	71.05	50.00	60.53	57.89	57.89	49.12	50.88
<i>hist-l1</i>	62.28	74.56	46.49	64.91	58.77	60.53	50.00	51.75
<i>hist-intersect</i>	66.67	68.42	43.86	61.40	40.35	53.51	53.51	50.88
<i>hist-diffusion</i>	62.28	74.56	46.49	64.91	58.77	60.53	50.00	51.75
<i>hist-chi</i>	57.02	71.05	50.88	55.26	59.65	57.02	55.26	48.25
<i>hist-emd</i>	52.63	58.77	58.77	51.75	55.26	56.14	43.86	53.51
<i>pdf-l2</i>	57.02	74.56	57.02	60.53	50.88	55.26	57.02	51.75
<i>pdf-l1</i>	60.53	76.32	54.39	61.40	50.88	58.77	54.39	46.49
<i>pdf-emd</i>	59.65	75.44	57.89	53.51	60.53	60.53	50.00	52.63
<i>pdf-bc</i>	65.79	69.30	53.51	53.51	48.25	57.89	44.74	52.63
<i>pdf-kl</i>	66.67	70.18	55.26	48.25	52.63	59.65	48.25	49.12
<i>pdf-kl-orig</i>	64.04	64.91	49.12	51.75	57.02	59.65	55.26	46.49
<i>pdf-js</i>	65.79	71.93	52.63	54.39	48.25	60.53	53.51	48.25
<i>average</i>	60.53	76.32	51.75	60.53	54.39	60.53	54.39	49.12
<i>svm</i>	71.93		63.16		51.75		47.37	

10.7.2 Multi-ROI Classification

In this section, we will show our experiments where we combine multiple ROIs, fixing the modality and distance measure. We also conducted experiments by fixing the ROIs and combining multiple dissimilarity matrices using the same ROI. We see that the accuracy does not increase as compared to combining ROIs with fixed dissimilarity measure. This conforms to our previous studies, therefore here, we do not report on combination of distance measures with fixed ROI.

In this experiment, a multi-ROI approach is adapted to use all ROIs at the same time. All the dissimilarity matrices for each ROI are combined by averaging the normalized dissimilarity matrices. The second and third columns of Table 10.3 report the results on intensity histograms, using 1-NN rule on the dissimilarity matrices and the support vector classifiers in the dissimilarity spaces. Also in this case, the classification on the dissimilarity space clearly outperforms the standard approach. Moreover, the multi-ROI approach brings an improvement by confirming the complementary information enclosed onto the different brain subparts when we use *sv0* on the dissimilarity space. In most of the cases, the results from the averaged similarity matrices are better than the respective best single ROI results. The row average in Table 10.3 reports the error estimates computed on the overall dissimilarity matrix (combining all the measures and ROIs), which has the highest accuracy 76.32 % (same as combining all ROIs for *pdf-l1*) for both the standard approach and dissimilarity-based approach, respectively. The last row reports the accuracy of

combining all SVM classifiers in the original feature space. When we combine all the SVM classifiers on the original space, we get 71.93 % accuracy. This shows us that the dissimilarity space produces better results also when we consider classifier combination. We repeated the same experiments also with the other modalities. In Table 10.3, we can also see the results using the other modalities. We observe that again we get the most accurate results when we combine ROIs in the dissimilarity space using *sv0* except for mean curvature histograms where the best results are obtained using *Inn* (using dissimilarities again).

10.7.3 Combining Different Modalities

As a further step to understand how information from multiple sources can be combined to reach better classification accuracy, we develop another experiment where we combine information from multiple modalities. We have 182 dissimilarity matrices from each of the four modalities. It is not possible to exhaustively search the whole solution space to find the best solution (optimum subset for combination), so instead, we choose the most accurate four ROI-dissimilarity pairs from each modality and do an exhaustive search on the combination of these matrices to get the best result. We can see the selected dissimilarity matrices and their base accuracies in Table 10.4. With a total of 16 dissimilarity matrices (modality-ROI-dissimilarity triples), we can get the best accuracy 86.84 % (last row in Table 10.4), which contains two dissimilarity matrices from intensities (*ldlpfc-pdf-kl-orig* and *ldlpfc-pdf-bc* both having 75.44 % accuracy) and one dissimilarity matrix from shape index (*rdlpfc-hist-chi* with 60.53 % accuracy). This accuracy is the best accuracy, which has been reached using dissimilarity combination and cannot be reached using only one modality. Applying the same methodology, we can reach only 76.32 % accuracy with *Inn* and 83.33 % accuracy with *svm* on the original feature space. This also shows us why it is important to combine useful information from different sources to come up with better accuracy. We see that the accuracy can be increased when complementary information using different modalities are combined.

In a medical application, besides increasing accuracy, the interpretability of the results is also important. We use this experiment to deduce information on the use of ROIs, their complementary information, and how each modality relates to the detection of schizophrenia. For this purpose, we select all the combinations of distance matrices with accuracies above 82 % (we have 69 different combinations) and count the occurrences of dissimilarity matrices for every combination. From Table 10.4, we can see that most of the combinations include *ldlpfc* of SMRI and the shape index of *rthal*. This shows us that these two modality-ROI pairs contribute and complement other dissimilarity matrices, and by using these two in combination, we increase accuracy. After these two dissimilarity matrices, we see that mean curvature of *rthal* and shape index of *rdlpfc* are used in combination the most. These are followed by *ldlpfc* of histogram intensities and the mean curvature of *rec*. With ADC, we see that most used ROI is *rstg*, which has been selected 38 times. This

Table 10.4 Most accurate four dissimilarity matrices from each modality, their single performances, and number of occurrences in the combination of most accurate results

Selected dissimilarity	Accuracy	Occurrences
SMRI-ldlpfc-pdf-js	76.32	60
SH-rthal-hist-l1	59.65	57
MCUR-rthal-pdf-bc	67.54	52
SH-rdlpfc-hist-chi*	60.53	50
SMRI-ldlpfc-pdf-bc*	75.44	48
SMRI-ldlpfc-pdf-kl-orig*	75.44	48
MCUR-rec-pdf-l1	63.16	47
SMRI-lamyg-pdf-bc	78.07	42
ADC-rstg-hist-l2	65.79	38
SH-lamyg-hist-emd	64.91	38
ADC-rstg-pdf-bc	70.18	20
MCUR-rec-pdf-l2	63.16	17
ADC-rdlpfc-pdf-emd	65.79	14
ADC-rstg-pdf-js	65.79	9
SH-lhg-hist-intersect	65.79	8
MCUR-lhippo-pdf-emd	62.28	1
Dissimilarities with * in the optimal combination are	86.84	

also shows us that the DWI information is the least complementary modality in this scenario, and one can design experiments without this modality, focusing on the other modalities. We can use this information to decrease the costs of the operation, that is, not performing DWI analysis. Also we see that the most accurate dissimilarity matrix (SMRI-*lamyg-pdf-bc*) is the eighth most used dissimilarity when we consider combination. This interesting fact shows us that when doing combination, the complementary information is more important than individual accuracies.

Another interesting fact is that some ROIs are more discriminative when the structural information is considered, and some are more discriminative when we consider DWI. The ROIs selected from the structural analysis in this experiment are those considered crucial for the impaired neural network in schizophrenia and comply with current studies in the literature [21], in contrast DWI is particularly keen in exploring the microstructural organization of white matter, therefore providing intriguing information on brain connectivity [13], but does not have complementary contribution in this context.

With this analysis, we can open a new perspective of how to use each of these modalities to get better accuracies. One can use this information to setup new experiments considering the contributions of these ROIs on these modalities.

10.7.4 Discussion

In this case study, a novel approach based on dissimilarity-based pattern recognition is proposed for the detection of schizophrenic brains. Several dissimilarity measures are proposed to deal with histograms of different types for different ROIs. ROI-based classification on the dissimilarity space shows improvements of the standard NN rule and the support vector classifier on the original space. Moreover, a Multi-ROI classification strategy is obtained by simply averaging the similarity matrices observed in each ROI. Such an approach improves the single-ROI one, by highlighting the complementary information enclosed in the several ROIs. This confirms the benefit of combining dissimilarity information and fusing information from various regions in the brain.

We investigate further to combine information from multiple modalities such as intensities, ADC values and geometric information. We can see that some ROIs are discriminative when we use intensities; some are useful when DWI data is considered. Geometric properties of some ROIs play a part in schizophrenia detection. We show that we get the best accuracy when we combine multiple modalities.

We can interpret the results of combining multiple modalities to set up further experiments in this context. Our results show that the least contributing modality is the DWI. With this information, one can skip using this modality and focus more on histograms of intensities and geometric information. Also, one can use this result to reduce the costs of this operation, by not performing DWI measurements and without the patient undergoing further medical operations.

We would like to emphasize that in building the (combined) dissimilarities no parameters are optimized w.r.t. performance. The proposed approach of combining dissimilarities on the dissimilarity space opens new perspectives in neuroanatomy classification by allowing the possibility to exploit dissimilarity measures where one does not have to deal with technical difficulties such as the metric requirements of distance based classification and kernel restrictions of support vector machines.

10.8 Case Study 2: Brain Classification by Generative Embeddings

In this case study, we use *Heat Kernel Signatures* to extract histogram-based features from SMRI and use the generative embedding score spaces mentioned in Sect. 10.5.3 and apply IT kernels [52]. We used average hold out methodology with 30 repetitions using stratification. For estimating the C value of the SVM and q value for the IT kernels, we used 5-fold cross-validation on the training set. To estimate the number of topics, we used the Bayesian Information Criterion (BIC) [67], which penalizes the likelihood with a penalty term on the number of free parameters in a way that larger models which do not increase the likelihood significantly are discouraged. In the pLSA model, the number of free parameters is calculated as

$(D - 1)Z + (W - 1)Z + (Z - 1)$. Then the BIC becomes

$$\text{BIC} = \frac{1}{2}((D - 1)Z + (W - 1)Z + (Z - 1)) \log \sum_{j=1}^W \sum_{i=1}^D n(d_i, w_j).$$

10.8.1 Proposed Approach

Kernels on probability measures have been shown to be very effective in classification problems involving text, images, and other types of data [24, 40]. Given two probability measures p_1 and p_2 , representing two objects, several information theoretic kernels (ITKs) can be defined [52]. In this work, we use the Jensen–Shannon kernel (JS), Jensen–Tsallis kernel (JT), and weighted JT kernel (since results were similar, we omit the weighted JT kernel (version B) [52]—we will refer to weighted JT kernel (version A) as JT-W). Once the generative model is estimated, the generative score spaces are calculated.

The approach herein proposed consists in defining a kernel between two observed objects x and x' as the composition of the score function with one of the JT kernels presented above. Formally,

$$k(x, x') = k_q^i(\phi_\Theta(x), \phi_\Theta(x')), \quad (10.15)$$

where $i \in \{\text{JT}, \text{A}, \text{B}\}$ indexes one of the Jensen–Tsallis kernels, and ϕ_Θ is one of the generative embeddings defined in Sect. 10.5.3. Notice that this kernel is well defined because all the components of ϕ_Θ^{FE} are non-negative.

We consider two types of kernel-based classifiers: K -NN and SVM. Recall that positive definiteness is a key condition for the applicability of a kernel in SVM learning. It was shown in [52] that k_q^A is a positive definite kernel for $q \in [0, 1]$, while k_q^B is a positive definite kernel for $q \in [0, 2]$. Standard results from kernel theory [69, Proposition 3.22] guarantee that the kernel k defined in (10.15) inherits the positive definiteness of k_q^i , thus can be safely used in SVM learning algorithms.

10.8.2 Results

We compare the results of our proposed approach with the linear kernel as a reference (which is the most used solution in the hybrid generative discriminative approach case, e.g., the Fisher Kernel). As classifiers we used Support Vector Machines and K-Nearest Neighbor (with K set to 1, i.e., the nearest neighbor rule). When possible, the classifiers have been applied also in the original domain (namely without the application of the generative embedding step).

Results are displayed in Table 10.5. In the table, “NN” stands for nearest neighbor results, while “SVM” refers to SVM results. “Linear” is the linear kernel, whereas

Table 10.5 Results on the brain classification task. See text for details

Embedding	Linear		JS		JT		JT-W	
	NN	SVM	NN	SVM	NN	SVM	NN	SVM
TPM-1	51.56	50.00	54.22	59.56	50.33	62.67	58.44	64.33
TPM-2	61.00	68.56	58.89	68.89	54.33	65.78	63.11	70.22
FESS-1	56.11	50.00	56.89	50.00	50.00	36.89	58.44	50.00
FESS-2	62.89	50.00	62.67	60.00	50.00	67.44	60.11	72.00
LLR	57.33	50.00	58.78	61.56	50.00	63.78	61.44	63.56
FSH	61.78	50.00	58.44	70.22	55.33	67.33	61.89	69.89
TOP	51.89	50.00	51.89	50.00	50.00	50.00	50.00	50.00
PD	75.22	50.00	74.78	50.00	62.67	80.56	72.56	80.78
ORIG	61.00	77.00	60.22	74.33	50.33	70.56	50.00	73.78

“JS”, “JT” and “JT-W” stand for Jensen–Shannon, Jensen–Tsallis, and Weighted Jensen–Tsallis kernels, respectively. The acronyms of the generative embeddings follow the notation described in Sect. 10.5.3: “TPM-1” is the posterior topic mixture for a single pLSA, “TPM-2” is the posterior topic mixture starting from one pLSA per class, “FESS-1” is the Free Energy Score Space for a single pLSA, “FESS-2” is the Free Energy Score Space obtained starting from one pLSA per class, “LLR” is the Log-Likelihood Ratio score space, “FSH” is the Fisher Score space, “TOP” is the TOP kernel score space and “PD” is the Posterior Divergence Score space. The standard errors of means, in all runs, were all less than 2.52.

From the table, different observations may be drawn:

- In almost all cases, the use of IT kernels over generative embeddings outperforms the linear kernel over the same embeddings, this being really evident in some cases.
- At the same time, the intermediate use of a generative embedding is almost always beneficial with respect to use the linear and the IT kernels on the original space.
- It is evident from the table that the best generative embedding is the very recently proposed Posterior Divergence Score Space. It seems that this generative embedding has a slight preference to be used with the IT kernels.
- There is no significant difference among the various IT kernels, even if it may be argued that the Weighted Jensen–Tsallis one is the most positive.
- Comparing the classifiers, there is no huge difference between the SVM and the Nearest Neighbor performances, thus confirming the goodness of the devised similarity measure.

10.9 Case Study 3: Scale Selection by MKL

In this case study, we use *Heat Kernel Signatures* to extract histogram based features (see also [16]) using different scales and using these as different sources for

Multiple Kernel Learning paradigm. The data is extracted from sMRI scans of the left thalamus of 30 schizophrenic patients and 30 healthy controls. Several kernels are computed (i.e., one kernel per scale), and a set of weights are estimated for the kernel combination. In this fashion, we can choose the most discriminative scales by selecting those associated to the highest weights, and vice versa. Moreover, kernel combination leads to a new similarity measure which increases the classification accuracy. It is important to note that in our approach we aim at selecting the best shape characteristics for classification purposes, hence, our selection is driven by the performance of a Support Vector Machine (SVM) classifier.

10.9.1 Methodology

The contribution of geometric features extracted at each scale are combined by employing the MKL strategy as described in Sect. 10.6.2. Each shape representation r_i is associated to a kernel k_m by leading to $n = P$ kernels. Indeed, both the weights (η_1, \dots, η_P) and the SVM parameters are estimated. In order to obtain the best classification accuracy, according to the *max-margin* paradigm an *alternating* approach is used between the optimization of kernel weights and the optimization of the SVM classifier. In each step, given the current solution of kernel weights, MKL solves a standard SVM optimization problem with the combined kernel. Then, a specific procedure is applied to update the kernel weights. Once the MKL procedure is completed, we obtain a two-fold advantage: (i) we can select the best scale contributions by keeping only the scales associated to the highest weights, and (ii) we can compose a new kernel from the weighted contributions of the best scales, which can be evaluated for classification purposes.

10.9.2 Experimental Protocol

In our experiments, we apply leave-one-out (LOO) cross-validation to assess the performance of the technique. Since LOO is used as the cross-validation technique, we do not report standard deviations or variances. We compare our results using k -fold paired t -test at $p = 0.05$. We collect geometric features at 11 scales generating different shape representations r_{01}, \dots, r_{11} . In practice, each representation r_i is a feature vector x_i which is plugged in the MKL framework. We employ the dot product as basic kernel function (i.e., linear kernel) since it avoids the estimation of free kernel parameters. Different strategies to combine the different shape representations have also been evaluated:

- **Single Best Kernel (Single-best)**—An SVM is trained separately per each representation. Therefore, the performances of the classification are evaluated separately at each scale. By doing so, we can evaluate the independent contributions coming from the different sources of information and select the best one.

Table 10.6 Single-kernel SVM accuracies

r01	r02	r03	r04	r05	r06	r07	r08	r09	r10	r11
75.00	78.33	76.67	76.67	73.33	*66.67	68.33	70.00	76.67	71.67	70.00

Table 10.7 MKL accuracies

SVM	SVM-con	RBMKL	SMKL	GLMKL
*78.3 (10, 11.7)	83.3 (8.3, 8.3)	*81.7 (10, 8.3)	86.7 (6.7, 6.7)	85.0 (8.3, 6.7)

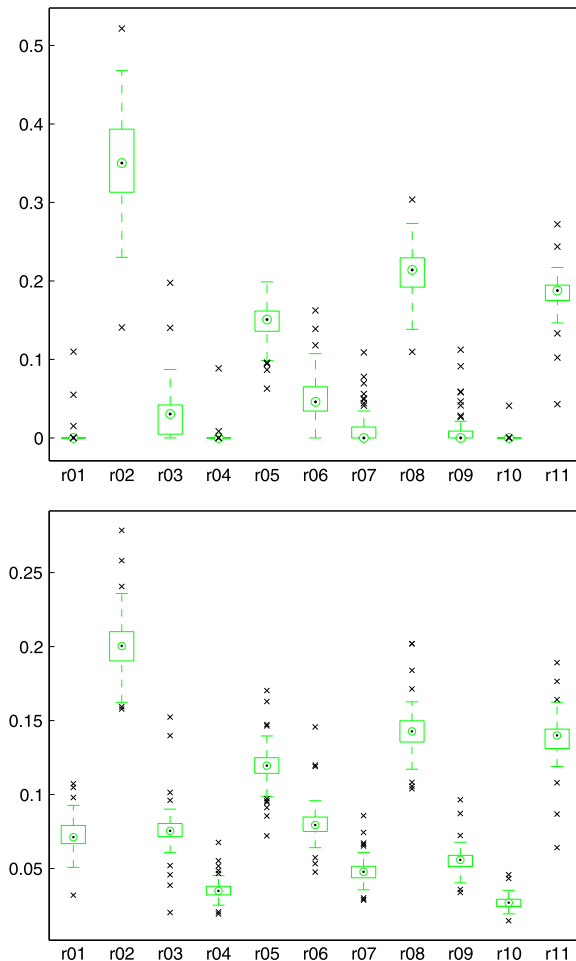
- Feature concatenation (SVM-con)—The contributions coming from the different sources are concatenated into a single feature vector. Then, a single SVM is employed for classification [19].
- Rule-based MKL (RBMKL)—As baseline MKL approach, the so-called rule-based method is evaluated: the kernels computed at each scale are combined by simply taking their average (i.e., $\forall m, \eta_m = 1/P$).
- Simple MKL (SMKL)—A simple but effective MKL algorithm is employed [60] by addressing the MKL problem through a weighted 2-norm regularization formulation with additional constraint on the weights that encourages sparse kernel combination.
- Group Lasso MKL (GLMKL)—It denotes the group Lasso-based MKL algorithms proposed by [43, 86]. A closed form solution for optimizing the kernel weights based on the equivalence between group-lasso and MKL is proposed. In our implementation, we used l_1 -norm on the kernel weights and learned a convex combination of the kernels.

10.9.3 Results

The first evaluation scores are shown in Table 10.6, which reports the single-best kernel accuracies for all feature representations. We can observe that the best performance is obtained at 78.33 % using r02 which is shown as bold face in the table. The entries marked with “*” show the accuracies which are statistically significantly less accurate than the best algorithm using k -fold paired t -test at $p = 0.05$.

Second, concatenating the features in a single vector leads to 83.33 % accuracy. Third, using the proposed three different MKL algorithms, we combined the eleven kernels by introducing the weights η_m . Table 10.7 reports the results of the best single-kernel SVM, the accuracy of the concatenated feature set, and the three MKL-based algorithms trained. The values in parentheses show the percentage of controls classified as schizophrenia and the percentage of patients classified as healthy, respectively. We achieve an accuracy of 86.67 %, reached by combining eleven kernels with the SMKL approach. This result is better than all other MKL settings and single-kernel SVMs. Further, GLMKL achieves 85 % accuracy which is

Fig. 10.7 Combination weights in MKL using the linear kernel: (*top*) using SMKL, (*bottom*) using GLMKL



still higher than that reached by the feature concatenation method. We can also note that we cannot overcome SVM-con when we use RBMKL, as the latter gives equal weight to each kernel. In fact, if there are inaccurate representations in the given set, the overall mean combination accuracy may be less of that reached using the single best. Conversely, when the weights are automatically estimated, such as in SMKL and GLMKL the selection of the most reliable information is carried out by the MKL procedure and the overall performance improves.

In Fig. 10.7, we plotted the weights of MKL for SMKL and GMKL algorithms to show the coherency of the weights. As expected, the best representation is r02, which has the highest weights. Although the other representations with high weights (r08, r11 and r05) do not provide accurate single-kernel SVMs results, their contributions to the overall accuracy in the combination is higher than those given by the other kernels. This demonstrates that when considering combinations, even a

Table 10.8 MKL accuracies on the selected subset of representations

SVM	SVM-con	RBMKL	SMKL	GLMKL
*78.3 (10, 11.7)	*83.3 (6.7, 10)	*83.3 (6.7, 10)	88.3 (6.7, 5)	85.0 (6.7, 8.3)

representation which does not lead to precise results may contribute to raise the overall combination accuracy. Moreover, we can also deduce that these four representations are the most useful in discriminating between healthy and schizophrenic subjects, and we may focus the attention on these properties only.

Using this information, we also performed the above pipeline using only these four representations, and we can observe the results in Table 10.8. Using this subset, we get the highest accuracy with SMKL, reaching 88.33 % of accuracy. We can also observe an increase in RBMKL.

10.9.4 Discussion

We have shown in general that MKL algorithms perform better than both single-best kernel SVMs and feature concatenation strategies. We have also observed that RBMKL (which does not compute weights while combining kernels) does not outperform the feature concatenation approach. Conversely, when the kernel combination is carried out by estimating proper weights, a substantial improvement is instead obtained. The kernel weights also allow us to extract useful information: it is interesting to observe that, for both MKL algorithms with the highest accuracy, four representations have the maximum effect (i.e., the highest weights), namely $r02$, $r08$, $r11$, and $r05$, with $r02$ being the best single-kernel. We use this information to select a smaller number of representations to reduce the costs of the feature extraction phase. Finally, we can also observe that by using such subset we can reach the best accuracy overall.

10.10 Conclusions

We have defined a set of new approaches to deal with schizophrenia detection from MRI images. We have proposed a working pipeline which takes into account different aspects of the disease. We have successfully applied the dissimilarity-based technique described in Chap. 2 to our medical application. In particular, we have shown that brain classification on dissimilarity space reaches a substantial improvement over standard feature-based approaches. Moreover, we have shown that combining dissimilarities represents a natural and effective approach to merge different sources of information. In this fashion, we were able to exploit complementary information about different parts of the brain, different acquisition modalities, and

different brain properties. Moreover, we have shown that our new paradigm to define data descriptors by generative embedding (see Chap. 4) is very effective and works well on our medical application. This research has opened new perspectives in the medical application which have been envisaged by our work. In particular, we have shown that a further improvement can be obtained by adapting random subspace method [81] to create the dissimilarity space.

Furthermore, we are working on employing advanced dissimilarity-based techniques to encode shape properties. Our preliminary results have shown an improvement by using Multiple Kernel Learning to improve the diffusion based shape description. Finally, we have shown in our experiments that DWI data was not important to improve the classification accuracy when multimodal approach was employed. This encourages us to exploit more advanced imaging techniques such as Diffusion Tensor MRI or Functional MRI to further improve schizophrenia detection.

References

1. Alpaydm, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
2. Amaddeo, F., Tansella, M.: Information systems for mental health. *Epidemiol. Psichiatr. Soc.* **18**(1), 1–4 (2009)
3. Andreone, N., Tansella, M., Cerini, R., Versace, A., Rambaldelli, G., Perlini, C., Dusi, N., Pelizza, L., Balestrieri, M., Barbui, C., Nose, M., Gasparini, A., Brambilla, P.: Cortical white-matter microstructure in schizophrenia. diffusion imaging study. *Br. J. Psychiatry* **191**, 113–119 (2007)
4. Ashburner, J., Friston, K.J.: Voxel-based morphometry-the methods. *NeuroImage* **11**(6), 805–821 (2000)
5. Awate, S.P., Yushkevich, P., Song, Z., Licht, D., Gee, J.C.: Multivariate high-dimensional cortical folding analysis, combining complexity and shape, in neonates with congenital heart disease. In: Proceedings of the 21st International Conference on Information Processing in Medical Imaging, IPMI'09, pp. 552–563 (2009)
6. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21st International Conference on Machine Learning, pp. 41–48 (2004)
7. Baiano, M., Perlini, C., Rambaldelli, G., Cerini, R., Dusi, N., Bellani, M., Spezzapria, G., Versace, A., Balestrieri, M., Mucelli, R.P., Tansella, M., Brambilla, P.: Decreased entorhinal cortex volumes in schizophrenia. *Schizophr. Res.* **102**(1–3), 171–180 (2008)
8. Bellani, M., Brambilla, P.: The use and meaning of the continuous performance test in schizophrenia. *Epidemiol. Psichiatr. Soc.* **17**(3), 188–191 (2008)
9. Bicego, M., Lovato, P., Ferrarini, A., Delle Donne, M.: Biclustering of expression microarray data with topic models. In: Proceedings of the International Conference on Pattern Recognition, pp. 2728–2731 (2010)
10. Bicego, M., Lovato, P., Oliboni, B., Perina, A.: Expression microarray classification using topic models. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC'10, New York, NY, USA, pp. 1516–1520 (2010)
11. Bicego, M., Pekalska, E., Tax, D.M.J., Duin, R.P.W.: Component-based discriminative classification for hidden Markov models. *Pattern Recognit.* **42**, 2637–2648 (2009)
12. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via pLSA. In: Proceedings of the European Conference on Computer Vision, ECCV'06, pp. 517–530 (2006)

13. Brambilla, P., Tansella, M.: Can neuroimaging studies help us in understanding the biological causes of schizophrenia? *Int. Rev. Psychiatry* **19**(4), 313–314 (2007)
14. Bronstein, A.M., Bronstein, M.M.: Shape recognition with spectral distances. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 1065–1071 (2011)
15. Browne, A., Jakary, A., Vinogradov, S., Fu, Y., Deicken, R.: Automatic relevance determination for identifying thalamic regions implicated in schizophrenia. *IEEE Trans. Neural Netw.* **19**(6), 1101–1107 (2008)
16. Castellani, U., Mirtuono, P., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., Brambilla, P.: A new shape diffusion descriptor for brain classification. In: Fichtinger, G., Martel, A., Peters, T. (eds.) *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI'11. Lecture Notes in Computer Science*, vol. 6892, pp. 426–433 (2011)
17. Castellani, U., Perina, A., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., Brambilla, P.: Brain morphometry by probabilistic latent semantic analysis. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI'10, MICCAI*, pp. 177–184 (2010)
18. Cha, S.H., Srihari, S.N.: On measuring the distance between histograms. *Pattern Recognit.* **35**(6), 1355–1370 (2002)
19. Chang, C.C., Lin, C.J.: In: *LIBSVM: a Library for Support Vector Machines* (2001). <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
20. Cheng, D.S., Bicego, M., Castellani, U., Cerruti, S., Bellani, M., Rambaldelli, G., Atzori, M., Brambilla, P., Murino, V.: Schizophrenia classification using regions of interest in brain MRI. In: *Proceedings of Intelligent Data Analysis in Biomedicine and Pharmacology, IDAMAP'09*, pp. 47–52 (2009)
21. Corradi-Dell'Acqua, C., Tomelleri, L., Bellani, M., Rambaldelli, G., Cerini, R., Pozzi-Mucelli, R., Balestrieri, M., Tansella, M., Brambilla, P.: Thalamic-insular disconnectivity in schizophrenia: evidence from structural equation modeling. *Hum. Brain Mapp.* **33**, 740–752 (2012)
22. Cortes, C., Mohri, M., Rostamizadeh, A.: Learning non-linear combinations of kernels. In: *Advances in Neural Information Processing Systems*, vol. 22, pp. 396–404 (2010)
23. Cristani, M., Perina, A., Castellani, U., Murino, V.: Geo-located image analysis using latent representations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008)
24. Cuturi, M., Fukumizu, K., Vert, J.P.: Semigroup kernels on measures. *J. Mach. Learn. Res.* **6**, 1169–1198 (2005)
25. Davatzikos, C.: Why voxel-based morphometric analysis should be used with great caution when characterizing group differences. *NeuroImage* **23**(1), 17–20 (2004)
26. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* **39**(1), 1–38 (1977)
27. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. Wiley-Interscience, New York (2000)
28. Duin, R.P.W.: *Prtools, a Matlab toolbox for pattern recognition version 4.0.14* (2005). <http://www.prtools.org/>
29. Edelstein, W.A., Bottomley, P.A., Pfeifer, L.M.: A signal-to-noise calibration procedure for NMR imaging systems. *Med. Phys.* **11**, 180–185 (1984)
30. Fan, Y., Shen, D., Gur, R.C., Gur, R.E., Davatzikos, C.: COMPARE: classification of morphological patterns using adaptive regional elements. *IEEE Trans. Med. Imaging* **26**(1), 93–105 (2007)
31. Frey, B.J., Jovic, N.: A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(9), 1392–1416 (2005)
32. Gerig, G., Styner, M., Shenton, M.E., Lieberman, J.A.: Shape versus size: improved understanding of the morphology of brain structures. In: *Proceedings of the International Conference on Medical Image Computing, MICCAI'01*, pp. 24–32 (2001)

33. Giuliani, N.R., Calhouna, V.D., Pearlson, G.D., Francis, A., Buchanan, R.W.: Voxel-based morphometry versus region of interest: a comparison of two methods for analyzing gray matter differences in schizophrenia. *Schizophr. Res.* **74**(2–3), 135–147 (2005)
34. Gönen, M., Alpaydin, E.: Localized multiple kernel learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 352–359 (2008)
35. Gönen, M., Alpaydin, E.: Multiple kernel learning algorithms. *J. Mach. Learn. Res.* **12**, 2181–2238 (2011)
36. Gönen, M., Ulaş, A., Schüffler, P.J., Castellani, U., Murino, V.: Combining data sources non-linearly for cell nucleus classification of renal cell carcinoma. In: Pelillo, M., Hancock, E.R. (eds.) Proceedings of the International Workshop on Similarity-Based Pattern Analysis, SIMBAD'11. Lecture Notes in Computer Science, vol. 7005, pp. 250–260. Springer, Berlin (2011)
37. Hofmann, T.: Learning the similarity of documents: an information-geometric approach to document retrieval and categorization. In: Proceedings of the Conference on Advances in Neural Information Processing Systems, NIPS'02, pp. 914–920 (2000)
38. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.* **42**(1–2), 177–196 (2001)
39. Jaakkola, T.S., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Proceedings of the Conference on Advances in Neural Information Processing Systems, NIPS'98, Cambridge, MA, USA, vol. 11, pp. 487–493 (1998)
40. Jebara, T., Kondor, R., Howard, A.: Probability product kernels. *J. Mach. Learn. Res.* **5**, 819–844 (2004)
41. Kawasaki, Y., Suzuki, M., Kherif, F., Takahashi, T., Zhou, S.Y., Nakamura, K., Matsui, M., Sumiyoshi, T., Seto, H., Kurachi, M.: Multivariate voxel-based morphometry successfully differentiates schizophrenia patients from healthy controls. *NeuroImage* **34**(1), 235–242 (2007)
42. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3), 226–239 (1998)
43. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: l_p -norm multiple kernel learning. *J. Mach. Learn. Res.* **12**, 953–997 (2011)
44. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. *Image Vis. Comput.* **10**, 557–565 (1992)
45. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, New York (2004)
46. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.* **5**, 27–72 (2004)
47. Lee, W.J., Duin, R.P.W., Loog, M., Ibba, A.: An experimental study on combining Euclidean distances. In: 2nd International Workshop on Cognitive Information Processing (CIP), pp. 304–309 (2010)
48. Lewis, D.P., Jebara, T., Noble, W.S.: Nonstationary kernel combination. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 553–560 (2006)
49. Li, X., Lee, T.S., Liu, Y.: Hybrid generative-discriminative classification using posterior divergence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'11, pp. 2713–2720 (2011)
50. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'06, vol. 1, pp. 246–253 (2006)
51. Liu, Y., Teverovskiy, L., Carmichael, O., Kikinis, R., Shenton, M., Carter, C.S., Stenger, V.A., Davis, S., Aizenstein, H., Becker, J.T., Lopez, O.L., Meltzer, C.C.: Discriminative MR image feature analysis for automatic schizophrenia and Alzheimer's disease classification. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention, MICCAI'04, pp. 393–401 (2004)
52. Martins, A.F.T., Smith, N.A., Xing, E.P., Aguiar, P.M.Q., Figueiredo, M.A.T.: Nonextensive information theoretic kernels on measures. *J. Mach. Learn. Res.* **10**, 935–975 (2009)
53. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: Proceedings of the Conference on Advances in Neural Infor-

- mation Processing Systems, NIPS'02, vol. 14, pp. 841–848 (2002)
54. Nyúl, L.G., Udupa, J.K., Zhang, X.: New variants of a method of MRI scale standardization. *IEEE Trans. Med. Imaging* **19**(2), 143–150 (2000)
 55. Pekalska, E., Duin, R.P.W.: *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, Singapore (2005)
 56. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: Free energy score space. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems, NIPS'09*, vol. 22, pp. 1428–1436 (2009)
 57. Perina, A., Cristani, M., Castellani, U., Murino, V., Jojic, N.: A hybrid generative/discriminative classification framework based on free-energy terms. In: *Proceedings of the IEEE International Conference on Computer Vision, ICCV'09*, pp. 2058–2065 (2009)
 58. Pohl, K.M., Sabuncu, M.R.: A unified framework for MR based disease classification. In: *IPMI'09: Proceedings of the 21st International Conference on Information Processing in Medical Imaging*, pp. 300–313 (2009)
 59. Pruessner, J., Li, L., Serles, W., Pruessner, M., Collins, D., Kabani, N., Lupien, S., Evans, A.: Volumetry of hippocampus and amygdala with high-resolution MRI and three-dimensional analysis software: minimizing the discrepancies between laboratories. *Cereb. Cortex* **10**(4), 433–442 (2000)
 60. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simple MKL. *J. Mach. Learn. Res.* **9**, 2491–2521 (2008)
 61. Raviv, D., Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Volumetric heat kernel signatures. In: *Workshop on 3D Object Retrieval*, pp. 39–44 (2010)
 62. Ray, K.M., Wang, H., Chu, Y., Chen, Y.F., Bert, A., Hasso, A.N., Su, M.Y.: Mild cognitive impairment: apparent diffusion coefficient in regional gray matter and white matter structures. *Radiology* **24**, 197–205 (2006)
 63. Reuter, M., Wolter, F.E., Shenton, M., Niethammer, M.: Laplace–Beltrami eigenvalues and topological features on eigenfunctions for statistical shape analysis. *Comput. Aided Des.* **41**(10), 739–755 (2009)
 64. Rovaris, M., Bozzali, M., Iannucci, G., Ghezzi, A., Caputo, D., Montanari, E., Bertolotto, A., Bergamaschi, R., Capra, R., Mancardi, G.L., Martinelli, V., Comi, G., Filippi, M.: Assessment of normal-appearing white and gray matter in patients with primary progressive multiple sclerosis—a diffusion-tensor magnetic resonance imaging study. *Arch. Neurol.* **59**, 1406–1412 (2002)
 65. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.* **40**(2), 99–121 (2000)
 66. Rujescu, D., Collier, D.A.: Dissecting the many genetic faces of schizophrenia. *Epidemiol. Psychiatr. Soc.* **18**(2), 91–95 (2009)
 67. Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **6**, 461–464 (1979)
 68. Serratos, F., Sanfeliu, A.: Signatures versus histograms: definitions, distances and algorithms. *Pattern Recognit.* **39**(5), 921–934 (2006)
 69. Shawe-Taylor, J., Cristianini, N.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge (2004)
 70. Shenton, M.E., Dickey, C.C., Frumin, M., McCarley, R.W.: A review of MRI findings in schizophrenia. *Schizophr. Res.* **49**(1–2), 1–52 (2001)
 71. Smith, N., Gales, M.: Speech recognition using SVMs. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems, NIPS'02*, vol. 14, pp. 1197–1204 (2002)
 72. Smith, N.D., Gales, M.J.F.: Using SVMs to classify variable length speech patterns. *Tech. Rep. CUED/F-INFENG/TR-412*, Cambridge University Engineering Department (2002)
 73. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: *Proceedings of the Symposium on Geometry Processing, SGP'09*, pp. 1383–1392 (2009)
 74. Swain, M.J., Ballard, D.H.: Color indexing. *Int. J. Comput. Vis.* **7**(1), 11–32 (1991)

75. Taylor, W.D., Hsu, E., Krishnan, K.R.R., MacFall, J.R.: Diffusion tensor imaging: background, potential, and utility in psychiatric research. *Biol. Psychiatry* **55**(3), 201–207 (2004)
76. Timoner, S.J., Golland, P., Kikinis, R., Shenton, M.E., Grimson, W.E.L., Wells III, W.M.: Performance issues in shape classification. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI'02, pp. 355–362 (2002)
77. Toews, M., Wells III, W., Collins, D.L., Arbel, T.: Feature-based morphometry. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI'09, pp. 109–116 (2009)
78. Tomasino, B., Bellani, M., Perlini, C., Rambaldelli, G., Cerini, R., Isola, M., Balestrieri, M., Caligrave, S., Versace, A., Mucelli, R.P., Gasparini, A., Tansella, M., Brambilla, P.: Altered microstructure integrity of the amygdala in schizophrenia: a bimodal MRI and DWI study. *Psychol. Med.* **41**(2), 301–311 (2010)
79. Tsuda, K., Kawanabe, M., Rätsch, G., Sonnenburg, S., Müller, K.R.: A new discriminative kernel from probabilistic models. *Neural Comput.* **14**, 2397–2414 (2002)
80. Ulaş, A., Castellani, U., Mirtuono, P., Bicego, M., Murino, V., Cerruti, S., Bellani, M., Atzori, M., Rambaldelli, G., Tansella, M., Brambilla, P.: Multimodal schizophrenia detection by multiclassification analysis. In: Martín, C.S., Kim, S.W. (eds.) Proceedings of the Iberoamerican Congress on Pattern Recognition, CIARP'11. Lecture Notes in Computer Science, vol. 7042, pp. 491–498. Springer, Berlin (2011)
81. Ulaş, A., Castellani, U., Murino, V., Bellani, M., Tansella, M., Brambilla, P.: Heat diffusion based dissimilarity analysis for schizophrenia classification. In: M.L. et al. (ed.) IAPR International Conference on Pattern Recognition in Bioinformatics, PRIB'11. Lecture Notes in Bioinformatics, vol. 7036, pp. 306–317. Springer, Berlin (2011)
82. Ulaş, A., Duin, R.P.W., Castellani, U., Loog, M., Mirtuono, P., Bicego, M., Murino, V., Bellani, M., Cerruti, S., Tansella, M., Brambilla, P.: Dissimilarity-based detection of schizophrenia. *Int. J. Imaging Syst. Technol.* **21**(2), 179–192 (2011)
83. Ulaş, A., Yıldız, O.T., Alpaydın, E.: Eigenclassifiers for combining correlated classifiers. *Inf. Sci.* **187**, 109–120 (2012)
84. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, New York (1998)
85. Voets, N.L., Hough, M.G., Douaud, G., Matthews, P.M., James, A., Winmill, L., Webster, P., Smith, S.: Evidence for abnormalities of cortical development in adolescent-onset schizophrenia. *NeuroImage* **43**(4), 665–675 (2008)
86. Xu, Z., Jin, R., Yang, H., King, I., Lyu, M.R.: Simple and efficient multiple kernel learning by group Lasso. In: Proceedings of the 27th International Conference on Machine Learning, ICML'10, pp. 1175–1182 (2010)
87. Yoon, U., Lee, J.M., Im, K., Shin, Y.W., Cho, B.H., Kim, I.Y., Kwon, J.S., Kim, S.I.: Pattern classification using principal components of cortical thickness and its discriminative pattern in schizophrenia. *NeuroImage* **34**(4), 1405–1415 (2007)

Index

A

Apparent diffusion coefficient, 251, 252, 257
Approximate set coding, 52
Approximation capacity, 55, 58, 61
Artificial cognitive system, 1

B

Boltzmann weights, 49–51
Brain parcellation, 255

C

Channel capacity, 53, 60
Characteristic polynomial, 145
Classification, 268
 multi-classification, 268
Classifier, 26
 Fisher's linear discriminant, 28
 logistic classifier, 28
 nearest neighbor classifier, 27
 normal density Bayes classifier, 27
 Parzen density classifier, 27
 SVM classifier, 29
Cluster, 181, 182
 ESS-cluster, 186
Cluster process
 Gauss-Dirichlet cluster process, 164
 Wishart-Dirichlet cluster process, 164, 165
Clustering, 4, 8, 86, 87, 89, 93, 159, 180
 clustering ensemble, 86, 89, 96, 97
 clustering game, 182
 consensus clustering, 87, 93, 97
 evidence accumulation clustering, 7, 86, 96
 k-means clustering, 158
 pairwise clustering, 158, 159
 proximity-based clustering, 159
Co-association matrix, 7, 86, 90, 91, 97

Coding, 52, 53

Communication protocol, 53, 54

D

Determinant, 143
Diffusion-weighted imaging, 252
Dimensionality reduction, 94, 96
Dissimilarity, 4, 14, 16, 17, 19, 20, 25
 (non-)Euclidean dissimilarity, 6, 16–18, 29, 32
 (non-)intrinsic dissimilarity, 17, 18
Distance, 14, 16, 17, 21, 23, 270
 χ^2 distance, 262
 Bhattacharyya distance, 262
 diffusion distance, 261
 EMD distance, 262
Divergence
 Jensen-Shannon divergence, 75
 Jensen-Tsallis q -divergence, 75
 KL divergence, 5, 262
 posterior divergence, 73
Dominant set, 129, 187, 188

Dynamics

 infection and immunization dynamics, 192, 193
 payoff-monotonic dynamics, 191
 replicator dynamics, 182, 190

E

Embedding, 6–8, 15, 20, 23–25, 30, 123
 constant shift embedding, 158, 161
 curvature dependent embedding, 124
 generative embedding, 264
 geometric manifold embedding, 123
 graph embedding, 129
 kernel embedding, 132
 spectral embedding, 8, 123

- Embedding (*cont.*)
 - spherical embedding, 8, 132, 135
 - structure-preserving embedding, 123
- Embedding methods, 94
 - curvilinear component analysis, 96
 - curvilinear distance analysis, 96
 - Isomap, 95
 - Laplacian eigenmap, 95
 - linear methods, 96
 - locality preserving projections, 96
 - locally linear embedding, 94
 - neighborhood preserving projections, 96
- Empirical risk, 48
- Equilibrium, 8, 184, 191
 - ESS equilibrium, 185, 188
 - Nash equilibrium, 184, 188, 192
- Essentialism, 2
- Evidence accumulation, 90
 - evidence accumulation clustering, 86, 96
 - probabilistic evidence accumulation, 87, 97
- Exponential map, 134
- F**
- Fisher score, 68, 72
- G**
- Game, 183
 - clustering game, 182, 185
 - matching game, 199
- Game theory, 8
- Generative embedding, 7, 68, 69, 236
 - latent-variable-based embedding, 72
 - parameter-based generative embedding, 72
 - pLSA-based generative embedding, 72
- Gibbs sampling, 168
- Global heat kernel signature, 259
- Graph, 4, 126
 - graph characteristics, 125
 - graph Laplacian, 126
 - graph representation, 128
 - graph spectrum, 126
 - oriented line graph, 144
- H**
- Histogram
 - histogram intersection, 261
 - intensity histogram, 256
- Hypergraph, 8, 140
 - hypergraph Laplacian, 140
 - hypergraph representations, 130
- Hypothesis class, 6, 46, 48, 49
- I**
- Image feature, 230, 232
- Information theory, 51, 52
- Invariants, 18
- K**
- Kernel, 68
 - Fisher kernel, 70
 - information-theoretic kernel, 7, 69, 73, 237
 - Jensen-Shannon kernel, 76
 - Jensen-Tsallis kernel, 76
 - kernel methods, 3
 - positive definite kernel, 73
- L**
- Learning, 4
 - active learning, 240
 - kernel learning, 231
 - multiple kernel learning, 234, 269
 - relaxation labelling, 4
 - tree learning, 225
- M**
- Magnetic resonance brain imaging, 6
- Magnetic resonance imaging, 248
- Matching
 - matching game, 199
 - point-pattern matching, 199
- Model selection, 58, 61, 62
- Multi-dimensional scaling, 166
- N**
- Nucleus
 - nucleus classification, 230, 236
 - nucleus detection, 225
 - nucleus segmentation, 228
- O**
- Object detection, 226
- P**
- Pairwise stability, 92
- Partition process, 164
 - exchangeable partition process, 164
- Pattern analysis, 1, 46, 47, 50, 61
 - similarity-based pattern analysis, 5
- Payoff, 183–185
- Probabilistic latent semantic analysis, 265
- Property
 - accidental property, 2
 - essential property, 2
- Q**
- Quality control, 171, 172
- R**
- Region of interest, 250

- Renal cell carcinoma, [9](#), [69](#), [172](#), [230](#), [236](#), [238](#), [242](#)
- Representation
 - feature-vector representation, [3](#)
- S**
- Segmentation
 - graph cut, [228](#)
 - nucleus segmentation, [228](#)
 - WM-GM-CSF segmentation, [254](#)
- Shape descriptor, [259](#)
 - geometric shape descriptor, [257](#)
 - spectral shape descriptor, [258](#)
- Shift-invariance, [161](#)
- Space
 - dissimilarity space, [20](#), [21](#), [23](#), [25](#), [29](#), [30](#), [260](#)
 - Euclidean space, [3](#)
 - indefinite space, [132](#)
 - pseudo-Euclidean space, [20](#)
 - score space, [266](#)
 - spherical space, [133](#)
 - vector space, [13](#), [14](#), [20](#), [21](#), [29](#)
- Standard quadratic program, [188](#)
- Statistical learning, [47](#), [48](#)
- Strategy
 - infective strategy, [193](#)
 - mixed strategy, [190](#)
 - pure strategy, [183](#)
 - strategy profile, [183](#)
- Structure-preserving embedding, [8](#)
- Superpixels, [229](#)
- Surface alignment, [183](#), [202](#), [206](#)
- Survival analysis, [240](#)
- Suyari's entropy, [74](#)
- T**
- Tissue micro array, [6](#)
- Tissue microarray, [69](#), [77](#), [171](#), [220](#), [236](#)
 - TMA processing pipeline, [222](#)
 - TMARKER, [223](#)
- Typical instance, [51](#)
- Z**
- Zeta function, [125](#), [127](#)
 - Ihara zeta function, [8](#), [127](#), [132](#), [140](#), [143](#)