IMT Institute for Advanced Studies, Lucca

Lucca, Italy

A Perceptual Learning Model to Discover the Hierarchical Latent Structure of Image Collections

PhD Program in Computer Science Engineering

XX Cycle

By

Davide Bacciu

2008

The dissertation of Davide Bacciu is approved.

Program Coordinator: Prof. Ugo Montanari, Università di Pisa

Supervisor: Prof. Antonina Starita, Università di Pisa

Supervisor: Prof. Paulo J. G. Lisboa, Liverpool John Moores University

Tutor: Prof. Antonina Starita, Università di Pisa

The dissertation of Davide Bacciu has been reviewed by:

Prof. Paul H. Lewis, University of Southampton

Dr. Alfredo Vellido, Universitat Politècnica de Catalunya

IMT Institute for Advanced Studies, Lucca

2008

to Antonina Starita
(*in memoriam*)

Acknowledgements

My first thought and my biggest acknowledgement goes to my late supervisor, Prof. Antonina Starita. I know dissertations invariably start by thanking supervisors, sometimes wholeheartedly and some other times just for political correctness. Here, I would like to acknowledge Tonina as my mentor, as well as a friend who has never missed to show her support and care even in the worse times of her disease. I know her last words for me were of reliance and appreciation; I will do my best, throughout my future works and actions, to show that I deserved such words. I would also like to thank my second supervisor, Prof. Paulo Lisboa, for taking up the burden of guiding me in the last steps of my PhD path, as well as for all the mind-challenging discussions in front of his whiteboard.

A word of acknowledgment goes to the two international reviewers, Prof. Paul Lewis and Dr. Alfredo Vellido, for their careful and deep evaluation of this verbose dissertation; their constructive criticisms helped putting mine into perspective.

The latter four years have been a long, tiring and fascinating journey that had influenced my thinking more than any other experience in life. I know that I owe this to the people who shared with me parts of this journey. Let me start by thanking my latest friends in LJMU, Ian, Terence and Inigo, for welcoming me in their Lab, showing my way in Liverpool (pubs in particular!) and for sharing countless coffee-break discussions that were worth the pain of English coffee. I would also like to thank my CI&MLG group in Pisa: Alessio for his continuous scientific support, Franco Alberto who has shared with

me the downsides of a PhD and Katuscia, for a friendship that has gone past the walls of the Lab. A very special thanks goes to my fellow friends at IMT, Alessio, Anna, Erika, Leonardo, Matteo, Michela, Mike and Veronica, my Lucca's family gathered around a table where we've shared expectations, ideas, projects and dreams (not to mention a significant number of bottles of superb tuscanian wine).

I shall thank my friends in Pisa, for taking me as I am and for being there anytime I needed, as well as I shall thank my family, for growing me with the firm belief that I'm holding my future in my hands and that their love and trust will always support my life choices. Finally, thank you Giulia for making all good for me, for a countless number of things, for sharing your life with me: if I eventually got out of the PhD tunnel, it has been for the light that I always saw at the end of it, it has been for your eyes.

Contents

List of Figures

List of Tables

List of Abbreviations

AHC ..... Agglomerative Hierarchical Clustering
AIC ..... Akaike Information Criterion
AILBG ..... Adaptive Incremental Linde Buzo Gray
ALL ..... Acute Lymphoblastic Leukemia
AML ..... Acute Myeloid Leukemia
APV ..... Asymptotic Property Vector
ART ..... Adaptive Resonance Theory
BIC ..... Bayesian Information Criterion
BMU ..... Best Matching Unit
CC_{HC} ..... Consensus Clustering with Hierarchical Clustering
CC_{SOM} ..... Consensus Clustering with Self Organizing Maps
CL ..... Competitive Learning
CMRM ..... Cross Media Relevance Model
CoRe ..... Competitive Repetition-suppression
CRM ..... Continuous Relevance Model
DSRPCL ..... Distance Sensitive Rival Penalized Competitive Learning
ECKLVQ ..... Expansive Competitive Kernel Learning Vector Quantization
ECLVQ ..... Expansive Competitive Learning Vector Quantization
EFS ..... Event Free Survival
ELBG ..... Enhanced Linde Buzo Gray
EM ..... Expectation Maximization
EMD ..... Earth Movers Distance

FCM ..... Fuzzy C-Means
FFT ..... Fast Fourier Transform
FSCL ..... Frequency Sensitive Competitive Learning
GLOH ..... Gradient Location and Orientation Histogram
GMM ..... Gaussian Mixture Model
GNG ..... Growing Neural Gas
HC ..... Hierarchical Clustering
HDP ..... Hierarchical Dirichlet Process
HRPLSA ..... Hierarchical Region-topic Probabilistic Latent Semantic Analysis
HSV ..... Hue Saturation Value
KLVQ ..... Kernel Learning Vector Quantization
KPCA ..... Kernel Principal Component Analysis
LBG ..... Linde Buzo Gray
LDA ..... Latent Dirichlet Allocation
LSI ..... Latent Semantic Indexing
LVQ ..... Learning Vector Quantization
MDL ..... Minimum Description Length
MIL ..... Multiple-instance learning
MML ..... Minimum Message Length
MSE ..... Mean Squared Error
OPTOC ..... One Prototype Take One Cluster
PAM ..... Partitioning Around Medoids
PCA ..... Principal Component Analysis
PSNR ..... Peak Signal to Noise Ratio
RCA ..... Robust Competitive Clustering
RF ..... Receptive Field
RGB ..... Red Green Blue
RPCL ..... Rival Penalized Competitive Learning
RS ..... Repetition Suppression
SCAD ..... Simultaneous Clustering and Attribute Discrimination
SCM ..... Similarity-based Clustering Method
SIFT ..... Scale Invariant Feature Transform
SOM ..... Self Organizing Maps

SSCL ..... Self-Splitting Competitive Learning
SURF ..... Speeded Up Robust Features
SVC ..... Support Vector Clustering
SVM ..... Support Vector Machine
VBG ..... Variational Bayesian Gaussian Mixture
VBS ..... Variational Bayesian Mixtures with Splitting
VQ ..... Vector Quantization
WBC ..... Wisconsin Breast Cancer
WTA ..... Winner Takes All

Vita

July 29, 1979	Born, Nuoro (NU), Italy
2003	B.Sc. Computer Science Final mark: 110/110 cum laude Università di Pisa, Pisa, Italy
2003	M.Sc. Computer Science Final mark: 110/110 cum laude Università di Pisa, Pisa, Italy
Jan. 2004 - Mar. 2005	Research Assistant Arts Lab, Scuola Superiore Sant'Anna Pisa, Italy
May 2004 - Dec. 2005	System Design and Software Development Robotech Srl Peccioli, PI, Italy
Nov. 2007 - Feb. 2008	Visiting Student Neural Computation Research Group, Liverpool John Moores University Liverpool, UK

Publications

1. D. Bacciu and A. Starita, "Expansive Competitive Learning for Kernel Vector Quantization", *Submitted to Pattern Recognition Letters*, 2008.
2. D. Bacciu and A. Starita, "Competitive Repetition Suppression (CoRe) Clustering: a Biologically inspired Learning Model with Application to Robust Clustering", *IEEE Transactions on Neural Networks*, To Appear, 2008.
3. D. Bacciu, E. Biganzoli, P.J.G. Lisboa and A. Starita, "Unsupervised Breast Cancer Class Discovery: a Comparative Study on Model-based and Neural Clustering", *Chapter in Computational Intelligence in human cancer research*, KES Rapid Research Results Series, 2008.
4. D. Bacciu, A. Bellandi, B. Furletti, V. Grossi and A. Romei, "Discovering Strategic Behaviors in Multi-Agent Scenarios by Ontology-Driven Mining", *Chapter in Advances in Robotics, Automation and Control*, pp. 171–198, To Appear, 2008.
5. D. Bacciu, E. Biganzoli, P.J.G. Lisboa and A. Starita, "Are Model-based Clustering and Neural Clustering Consistent? A Case Study from Bioinformatics", *Proceedings of the 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'08)*, LNCS, Vol. 5178, pp. 181 – 188, Springer, 2008.
6. D. Bacciu, A. Botta and L. Badia, "Fuzzy Admission Control with Similarity Evaluation for VoWLAN with QoS Support", *Proceedings of the 5th Annual Conference on Wireless On demand Network Systems and Services (WONS'08)*, pp. 57 – 64, IEEE, 2008.
7. D. Bacciu and A. Starita, "Convergence Behavior of Competitive Repetition-Suppression Clustering", *Proceedings of 14th International Conference on Neural Information Processing (ICONIP'07)*, LNCS, Vol. 4984, pp.497–506, Springer, 2008.
8. D. Bacciu, A. Botta and D. Stefanescu, "A framework for semantic querying of distributed data-graphs via information granules", *Proceedings of the 10th IASTED International Conference on Intelligent Systems and Control (ISC'07)*, Cambridge, US, 2007.
9. D. Bacciu and A. Starita, "A Robust Bio-Inspired Clustering Algorithm for the Automatic Determination of Unknown Cluster Number", *Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN'07)*, pp. 1314-1319, IEEE, 2007.

10. D. Bacciu, A. Botta and D. Stefanescu, "Augmenting the Distributed Evaluation of Path Queries via Information Granules", *Proceedings of the 5th International Workshop on Mining and Learning with Graphs (MLG'07)*, pp. 105-109, 2007.

11. D. Bacciu, A. Micheli and A. Starita, "Simultaneous Clustering and Feature Ranking by Competitive Repetition Suppression Learning with Application to Gene Data", *Proceedings of the 2007 International Conference on Computational Intelligence in Medicine and Healthcare (CIMED'07)*, Plymouth, UK, 2007.

12. D. Bacciu, L. Badia and A. Botta, "Fuzzy Agreement for Network Service Contracts", *Proceedings of the 6th International Conference on Computational Intelligence in Economics and Finance (CIEF'07)*, Salt Lake City, Utah, USA, 2007.

13. D. Bacciu, A. Botta and H. Melgratti, "A fuzzy approach for negotiating quality of services", *Proceedings of the 2006 Trustworthy Global Computing Conference (TGC'06)*, LNCS, Vol. 4661, pp. 200–217, Springer, 2006.

14. D. Bacciu and A. Starita, "Competitive Repetition-suppression (CoRe) Learning", *Proceedings of the 2006 International Conference on Artificial Neural Networks (ICANN'06)*, LNCS, Vol. 4131, pp. 130 – 139, Springer, 2006.

15. J. Cinkelj, M. Mihelj, D. Bacciu, M. Jurak, E. Guglielmelli, A. Toth, S. Mazzoleni, J. DeLafonteyne, J. Verschelde, M. Munih, "Assessment of stroke patients by whole-body isometric force-torque measurements II: software design of the ALLADIN Diagnostic Device", EMBEC 2005.

16. D. Bacciu, L. Zollo, E. Guglielmelli, F. Leoni, A. Starita, "An RLWPR Network for Learning the Internal Model of an Anthropomorphic Robot Arm", *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04)*, Vol. 1, pp. 260 - 265, 2004, IEEE.

Presentations

1. D. Bacciu, "Are Model-based Clustering and Neural Clustering Consistent? A Case Study from Bioinformatics", at *KES 2008*, Zagreb, Croatia, 2008.
2. D. Bacciu, "Convergence Behavior of Competitive-Repetition Suppression Clustering", at *ICONIP 2007*, Kitakyushu, Japan, 2007.
3. D. Bacciu, "A Robust Bio-Inspired Clustering Algorithm for the Automatic Determination of Unknown Cluster Number", at *IJCNN 2007*, Orlando, Florida, 2007.
4. D. Bacciu, "Simultaneous Clustering and Feature Ranking by Competitive Repetition Suppression Learning with Application to Gene Data", at *CIMED 2007*, Plymouth, UK, 2007.
5. D. Bacciu, "Fuzzy Agreement for Network Service Contracts", at *CIEF 2007*, Salt Lake City, Utah, 2007.
6. D. Bacciu, A. Botta "Agenti Software Intelligenti per la Generazione Automatica di Contratti", at *Research to Business (R2B'07)*, Bologna, Italy, 2007.
7. D. Bacciu, "A fuzzy approach for negotiating quality of services", at *TGC 2006*, Lucca, Italy, 2006.
8. D. Bacciu, "Competitive Repetition Suppression Learning", at *ICANN 2006*, Athens, Greece, 2006.
9. D. Bacciu, "Repetita Iuvant? Constructive and destructive effects of redundancy and repetition in art, biology and computer science", at *CSE Seminars, IMT Lucca*, Lucca, Italy, 2006.

Abstract

Biology has been an unparalleled source of inspiration for the work of researchers in several scientific and engineering fields including computer vision. The starting point of this thesis is the neurophysiological properties of the human early visual system, in particular, the cortical mechanism that mediates learning by exploiting information about stimuli repetition.

Repetition has long been considered a fundamental correlate of skill acquisition and memory formation in biological as well as computational learning models. However, recent studies have shown that biological neural networks have different ways of exploiting repetition in forming memory maps. The thesis focuses on a perceptual learning mechanism called repetition suppression, which exploits the temporal distribution of neural activations to drive an efficient neural allocation for a set of stimuli. This explores the neurophysiological hypothesis that repetition suppression serves as an unsupervised perceptual learning mechanism that can drive efficient memory formation by reducing the overall size of stimuli representation while strengthening the responses of the most selective neurons. This interpretation of repetition is different from its traditional role in computational learning models mainly to induce convergence and reach training stability, without using this information to provide focus for the neural representations of the data.

The first part of the thesis introduces a novel computational model with repetition suppression, which forms an unsupervised competitive system termed CoRe, for Competitive Repetition-suppression learning. The model is applied to general

problems in the fields of computational intelligence and machine learning. Particular emphasis is placed on validating the model as an effective tool for the unsupervised exploration of bio-medical data. In particular, it is shown that the repetition suppression mechanism efficiently addresses the issues of automatically estimating the number of clusters within the data, as well as filtering noise and irrelevant input components in highly dimensional data, e.g. gene expression levels from DNA Microarrays. The CoRe model produces relevance estimates for the each covariate which is useful, for instance, to discover the best discriminating bio-markers.

The description of the model includes a theoretical analysis using Huber's robust statistics to show that the model is robust to outliers and noise in the data. The convergence properties of the model also studied. It is shown that, besides its biological underpinning, the CoRe model has useful properties in terms of asymptotic behavior. By exploiting a kernel-based formulation for the CoRe learning error, a theoretically sound motivation is provided for the model's ability to avoid local minima of its loss function. To do this a necessary and sufficient condition for global error minimization in vector quantization is generalized by extending it to distance metrics in generic Hilbert spaces. This leads to the derivation of a family of kernel-based algorithms that address the local minima issue of unsupervised vector quantization in a principled way. The experimental results show that the algorithm can achieve a consistent performance gain compared with state-of-the-art learning vector quantizers, while retaining a lower computational complexity (linear with respect to the dataset size).

Bridging the gap between the low level representation of the visual content and the underlying high-level semantics is a major research issue of current interest. The second part of the thesis focuses on this problem by introducing a hierarchical and multi-resolution approach to visual content under-

standing. On a spatial level, CoRe learning is used to pool together the local visual patches by organizing them into perceptually meaningful intermediate structures. On the semantical level, it provides an extension of the probabilistic Latent Semantic Analysis (pLSA) model that allows discovery and organization of the visual topics into a hierarchy of aspects. The proposed hierarchical pLSA model is shown to effectively address the unsupervised discovery of relevant visual classes from pictorial collections, at the same time learning to segment the image regions containing the discovered classes. Furthermore, by drawing on a recent pLSA-based image annotation system, the hierarchical pLSA model is extended to process and represent multi-modal collections comprising textual and visual data. The results of the experimental evaluation show that the proposed model learns to attach textual labels (available only at the level of the whole image) to the discovered image regions, while increasing the precision/recall performance with respect to flat, pLSA annotation model.

Chapter 1

Introduction

1.1 Motivation

The tremendous growth of multimedia repositories has been fostering the work of researchers in multiple areas of computer science over the last decade. The heterogeneous nature of such information, encompassing textual, visual and auditive data, motivates the need for developing alternative access methods with respect to those used to store, index, process and retrieve traditional textual information.

Image data, in particular, has undergone a massive growth in size due to the progress in acquisition of visual information, encompassing satellite images, pictures of everyday life captured by digital cameras, frames from closed-circuit cameras and medical images. This considerable amount of data can reveal useful high-level semantic information which, however, needs to be extracted from the low-level pixel-based representation. For this reasons there is an increasing demand for *image mining* tools, whose fundamental challenge encompasses the extraction of implicit knowledge, image data relationships and other patterns not explicitly stored in visual data collections.

Image mining draws resources and knowledge from multiple research fields such as machine vision, image processing, image retrieval, data mining, machine learning, database and distributed systems. Moreover,

an image mining system cannot be built by simply integrating various techniques borrowed from the aforementioned areas, whereas it requires an extensive rethinking of the models in the view of the unique research issues characterizing the process of knowledge extraction from pictorial information.

The last ten years of image mining studies have been characterized by an extremely active research in the field of *Content-Based Image Retrieval* (CBIR) systems (LSDJ06). Despite the name, most of the CBIR approaches developed so far have been trying to tackle the problem of visual information mining more from a low-level, image-feature perspective (e.g. color, texture, shape, interest points), than from a semantic-oriented perspective (HNDA04). Recently, there has started to be an increasing attention towards new machine vision models that overtake such low-level approaches in favor of stratified representations (e.g. latent topic models) that can convey a richer semantics into the visual content description.

Our research effort began with the objective of developing a biologically inspired computational model addressing visual content understanding from a hierarchical perspective, where information is incrementally refined to extract higher level concepts from the bottom to the top of the hierarchy. Computer science started its long joint journey with biology and neuroscience since its early days, back in 1943, when McCulloch and Pitts published *A Logical Calculus of the Ideas Immanent in Nervous Activity* (MP43), laying foundations for artificial neural networks. Since then, the relationship between the sciences of the artificial and of the natural has evolved into a two-way association, where one serves as a means for improving the other. This dissertation drew inspiration from the study of those powerful cortical memory mechanisms that allow visual information to be processed in isolation before being incrementally merged and isolated into semantically meaningful entities. In particular, our initial research statement focused on the development of a *fully cognitive* approach to image content understanding, founding on a computational model of the biological memory mechanisms underpinning perceptual learning (DD95). However, research is a story about wins and losses and, as the doctoral study evolved, we had to abandon the hy-

pothesis that perceptual learning alone could be exploited to build an efficient model for visual content understanding. On the other hand, our studies on the neural correlates of perceptual learning led us to the development of a novel competitive learning model inspired by the *repetition suppression* phenomenon (Des96). This cortical memory mechanism, exploits the temporal distribution of neural activations to drive an efficient neural allocation for a set of stimuli. In this dissertation, we explore the neurophysiological hypothesis that repetition suppression serves as an unsupervised perceptual learning mechanism that can drive efficient memory formation by reducing the overall size of stimuli representation while strengthening the responses of the most selective neurons. This interpretation of repetition is different from its traditional role in computational learning models mainly to induce convergence and reach training stability, without using this information to provide focus for the neural representations of the data.

The first part of this dissertation is devoted to the formalization of a novel computational model inspired by repetition suppression, which forms an unsupervised competitive system termed *CoRe*, for *Competitive Repetition-suppression learning* (BS08). In particular, it is shown that the repetition suppression mechanism efficiently addresses the issues of automatically estimating the number of clusters within the data, as well as filtering noise and irrelevant input components in highly dimensional data. The description of the model includes a theoretical analysis using robust statistics (Hub81) to show that the model is robust to outliers and noise in the data. Moreover, it is also given a convergence analysis (BS07a) that shows that, besides its biological underpinning, the CoRe model has useful properties in terms of asymptotic behavior. Furthermore, the study of CoRe's convergence analysis yields, as an additional contribution in Part I, to the formalization of a family of optimized kernel-based algorithms addressing unsupervised vector quantization. The performance of the proposed models is evaluated on general problems in the fields of computational intelligence and machine learning. Particular care is devoted to evaluating CoRe's performance as an exploratory tool for biomedical datasets comprising both low dimensional samples of meaning-

ful proteomic expression levels (ABQ+06), as well as high dimensional data from DNA Microarrays.

In the second part of the work we recover the initial motivation of the thesis, that is the development of a multilayered model for visual content understanding. Due to the failure to address the discovery of high level visual semantics by relying solely on perceptual learning mechanisms, we turned our attention to a recent approach to visual content characterization, that is the latent aspects model (Hof01; BNJ03). Starting from a text-like bag-of-words representation of the images, these models allow to estimate the underlying semantic content of a picture as a latent aspect (or topic) (SRE+05; FFFPZ05). In the second part of the dissertation we propose to extend probabilistic Latent Semantic Analysis (pLSA) (Hof01) to allow the discovery and organization of the visual topics into a hierarchy of aspects. The proposed model, named Hierarchical Region-topic Latent Semantic Analysis (HRPLSA), integrates with a multi-resolution image representation obtained by exploiting CoRe learning to pool together the local visual patches, organizing them into perceptually meaningful intermediate structures. As a result, we obtain a stratified representation both on the semantic (i.e. the topic) level as well as on the spatial level. The proposed HRPLSA model is applied to the unsupervised discovery of relevant visual classes from pictorial collections, at the same time learning to segment the image regions containing the discovered classes. Furthermore, we extend the hierarchical pLSA model, by drawing on a recent pLSA-based image annotation system (MGP07), to process and represent multi-modal collections comprising textual and visual data.

In the next section, we review the thesis outline by describing the relevant issues introduced in each chapter. Finally, Section 1.3 lists the published works that relate to the contributions presented in the dissertation.

1.2 Outline of the Thesis

The dissertation is organized into two parts corresponding to the two research areas that we are willing to contribute.

Part I describes a novel computational learning model inspired by a perceptual learning mechanism of the visual cortex. This part is organized as follows

Chapter 2 introduces the neuro-physiological foundations of perceptual learning and provides motivations underlying the development of the Competitive Repetition suppression (CoRe) model. Moreover, Chapter 2 presents a comparative analysis with the state-of-the-art models in unsupervised learning , as well as a performance evaluation on benchmark datasets.

Chapter 3 presents a theoretical analysis of competitive repetition suppression learning. In particular, we show how its error formulation can be interpreted in terms of robust M-estimators (Hub81) proving, both analytically and experimentally, that CoRe learning implements a cluster estimation process that is robust to noise and data outliers. The second part of Chapter 3 studies the convergence behavior of CoRe learning; within this discussion, we introduce a kernel-based formulation of the CoRe error function, that we exploit to provide theoretical motivations for CoRe's ability in avoiding local minima of its error function.

Chapter 4 draws on the convergence analysis developed for CoRe learning, developing further the ideas underlying the global minimum condition introduced in Chapter 3. In particular, we describe a competitive neural algorithm that performs unsupervised learning vector quantization in kernel space. The performance of the proposed algorithm is confronted with that obtained by optimized learning vector quantization algorithms, showing how the proposed method can sensibly reduce the quantization error by exploiting a simple global minimum term that can be computed in linear time (with respect to the dataset size).

Chapter 5 starts by analyzing CoRe's limitations when dealing with high-dimensional data, that is typically characterized by several irrelevant and noisy components. Drawing on this consideration, we

generalize CoRe's inhibition mechanism so that it can be used to suppress irrelevant input features, while providing a means for producing incremental estimates of the relevance of the input components (i.e. feature ranking). Through a set of bio-medical case studies, we show how the *feature-wise CoRe* algorithm can be effectively applied as an exploratory tool for discovering prognostic information, such as pathology sub-types and bio-markers regulating pathological processes.

Part II shifts the focus of the dissertation to the discovery of latent structures within visual content. In particular, we embed the perceptual learning mechanism introduced in Part I, into a multi-resolution hierarchical latent topic model. The content of this part is organized as follows

Chapter 6 introduces the state-of-the art in image representation techniques and presents a comparative analysis of the most relevant contributions to unsupervised image content understanding and annotation.

Chapter 7 describes a multi-resolution image representation approach exploiting the CoRe learning model introduced in Part I. This representation is embedded into a hierarchical latent aspect model, named Hierarchical Region-topic Probabilistic Latent Semantic Analysis, that is applied to the unsupervised characterization and segmentation of the semantic content of image collections. By applying a multi-modal extension developed for the flat probabilistic Latent Semantic Analysis (pLSA), we extend the hierarchical approach to model collections of images and associated captions. The effectiveness of the proposed model is tested in unsupervised visual content discovery and automatic image annotation.

Chapter 8 concludes the dissertation by reviewing the contributions to the state-of-the-art and by discussing future developments of this doctoral research.

1.3 Published Works

Part of the contributions presented in the dissertation have been published or have been submitted for publication. In particular, the first model based on Competitive Repetition-suppression learning appeared in

D. Bacciu and A. Starita, "Competitive Repetition-suppression (CoRe) Learning", *Proceedings of the 2006 International Conference on Artificial Neural Networks (ICANN'06)*, LNCS, Vol. 4131, pp. 130 – 139, Springer, 2006

and has been later extended in

D. Bacciu and A. Starita, "A Robust Bio-Inspired Clustering Algorithm for the Automatic Determination of Unknown Cluster Number", *Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN'07)*, pp. 1314-1319, IEEE, 2007.

An overall introduction of the CoRe model, covering mostly Chapter 2 and the first part of Chapter 3, is about to be published in

D. Bacciu and A. Starita, "Competitive Repetition Suppression (CoRe) Clustering: a Biologically inspired Learning Model with Application to Robust Clustering", *IEEE Transactions on Neural Networks*, To Appear, 2008.

The convergence analysis presented in the second part of Chapter 3 has appeared in

D. Bacciu and A. Starita, "Convergence Behavior of Competitive Repetition-Suppression Clustering", *Proceedings of 14th International Conference on Neural Information Processing (ICONIP'07)*, LNCS, Vol. 4984, pp.497–506, Springer, 2008.

The feature-wise extension of the CoRe algorithm described in Chapter 5 has been first introduced in

D. Bacciu, A. Micheli and A. Starita, "Simultaneous Clustering and Feature Ranking by Competitive Repetition Suppression Learning with Application to Gene Data", *Proceedings of the 2007 International Conference on Computational Intelligence in Medicine and Healthcare (CIMED'07)*, Plymouth, UK, 2007,

while its application to prognostic classes discovery described in Section 5.3.2 has appeared in

D. Bacciu, E. Biganzoli, P.J.G. Lisboa and A. Starita, "Are Model-based Clustering and Neural Clustering Consistent? A Case Study from Bioinformatics", *Proceedings of the 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES'08)*, LNCS, Vol. 5178, pp. 130 – 139, Springer, 2008

and will be published in an extended version in

D. Bacciu, E. Biganzoli, P.J.G. Lisboa and A. Starita, "Unsupervised Breast Cancer Class Discovery: a Comparative Study on Model-based and Neural Clustering", *Chapter in Computational Intelligence in human cancer research*, KES Rapid Research Results Series, 2008.

The kernel-based learning vector quantization algorithm described in Chapter 4 is currently under consideration for publication in

D. Bacciu and A. Starita, "Expansive Competitive Learning for Kernel Vector Quantization", *Submitted to Pattern Recognition Letters*, 2008

All published contributions are presented in this thesis in revised and extended form.

Part I

A Model of Perceptual Learning in the Visual Cortex

Chapter 2

Competitive Repetition-Suppression Learning

Repetita iuvant. (Latin Proverb)

2.1 Introduction

Repetition has long been considered a fundamental correlate of learning: already in Ancient Rome, Latins had a popular proverb referring to repetition as the mother of learning (i.e. *Repetitio est mater studiorum*). This early concept of repetition as a high road for conscious acquisition of knowledge has later evolved into a well-established idea of repetition mediating skill acquisition as well as memory formation at various representational levels and in several learning mechanism of the brain. Neurophysiological studies have shown that restated exposure to a perceptual stimulus is essential for the development of its neuronal representation, while the repeated execution of a motor task induces adaptation in the brain circuitry regulating motor control, resulting in an increased performance in the task. Similarly, repetition performs a strategic function in computational learning models, where patterns' presentation is iterated

throughout the training epochs until a given task-related performance criterion is met (e.g. the minimization of an objective function).

Recent studies have shown that biological neural networks have different ways of coping with repetition and exploiting it to drive knowledge and skill acquisition: here, we are particularly interested in a memory mechanism of the visual cortex called *repetition suppression* (Des96). This mechanism exploits the temporal distribution of neuron activations as a source of training information to drive an efficient neural allocation on a set of stimuli. In this work, we explore Desimone's neurophysiological hypothesis (Des96) that the repetition suppression mechanism serves as an unsupervised mean for reducing the size of stimuli representation, that is the number of neurons coding a given stimulus, while strengthening the responses of the most selective neurons, i.e. those showing sharp responses to particular classes of input stimuli.

This interpretation of repetition is different from its traditional understanding that is adopted, for instance, by the machine learning community. In computational learning models, in fact, repetition is typically exploited as part of the pattern sampling process to induce convergence and reach training stability, but it does not convey, by itself, any information into learning. Conversely, in the repetition suppression model, the knowledge regarding pattern repetition is actively used in determining the neural representation of the inputs. Our research effort focuses on this *active* approach to repetition, with the intent of developing a novel computational learning model inspired by the repetition suppression mechanism.

We introduce the *Competitive Repetition-suppression* (CoRe) neural network, an unsupervised learning model that, by mimicking the behavior of the repetition suppression mechanism, evolves a neural population in the direction of maximum selectivity by means of a procedure that penalizes or enhances the responses of the neurons on the basis of the stimuli frequency. In order to do that, CoRe resorts to competitive learning (RZ85) and, in particular, to the soft-competitive approach (YZG92; SO03) in which each unit, or the units from a selected subset, is allowed to adapt its weights in proportion to its activation strength. CoRe learning can

be used to efficiently determine an optimal allocation of feature detector units and to train their parameters locally, without external teaching signals and a-priori knowledge concerning the inputs distribution and the problem at hand.

The identification of the optimal number of feature detectors, along with their parameters, is a central problem in several machine learning tasks: in neural network models, for instance, the determination of the number of units that are needed to solve a task is often led to heuristics and to expert knowledge concerning the problem at hand. In this chapter, our attention will be primarily devoted to robust data clustering and, more specifically, to the traditional challenge of automatically estimating the appropriate number of data partitions. In the following we presents an in-depth description of the CoRe model, introducing a soft-competitive formulation that is evaluated in the context of unsupervised clustering: in particular, we interpret CoRe neurons as cluster detectors so that the repetition suppression mechanism can be used to selectively suppress irrelevant neurons, consequently determining the unknown cluster number k. In the comparative analysis with the clustering models in the literature, we focus on the relationship between CoRe clustering and a popular divide-and-conquer multi-agent learning algorithm, that is the Rival Penalized Competitive Learning (RPCL) model (XKO93; Xu07; Xu03). In particular, we point out how CoRe extends the winner-rival competition of RPCL to include set of units characterized by generic activation functions; moreover, we show how CoRe's repetition suppression mechanism positively addresses a key open issue of RPCL, that is how to adaptively modulate the strength of the rival penalization (Che05).

Before discussing the details of the CoRe model, we introduce, in Section 2.2, the biological processes from which we drew inspiration for the development of CoRe, while in Section 2.3 we describe related competitive learning models as well as an overview of the literature with respect to the cluster number identification task.

2.2 Biological Foundations

This section introduces the biological foundations of the CoRe model: before discussing the details of the memory mechanism modeled by CoRe learning, we give an high level introduction to the biological visual system and we discuss the main cortical mechanisms of implicit visual memory.

2.2.1 The Visual System

Hubel and Wiesel, in the pioneering work (HW62) which earned them the Nobel prize, discovered the strongly hierarchical nature of the vision system. Following their lesson, this section briefly introduces the structure of the vision system from a bottom up perspective, climbing the visual pathways in the direction of the visual information flow, from the retina to the high level cognitive centers.

The *retina* is a composite structure consisting of 3 functionally distinct layers of cells: *photoreceptors*, *collectors cells* and *retinal ganglion cells*. Photoreceptors come in two different types, namely *cones* and *rods*, where the former are specialized in color detection with high spatial resolution, while the the latter are not discriminant to wavelength (no color perception) and have low spatial resolution, but have high light sensitivity, therefore are of much use for night vision. At the level of retinal ganglion cells, the image undergoes an extensive processing, which reduces the amount of information transferred to higher visual areas. One of the major results of this process is that absolute levels of illuminations are replaced by a retinotopic map of light differences. This is done by the particular structure of the ganglion receptive fields, namely *center-surround*. An on-center off-surround cell receives excitatory inputs from a small patch of its visual field and inhibitory inputs from the surrounding regions via the lateral connections.

The information leaving the retina is split into different pathways, the major of which is directed to the *Lateral Geniculate Nucleus* (LGN). LGN is six-layered structure of the thalamus, made up of two magnocellular layers and four parvocellular layers of center-surround cells. The organi-

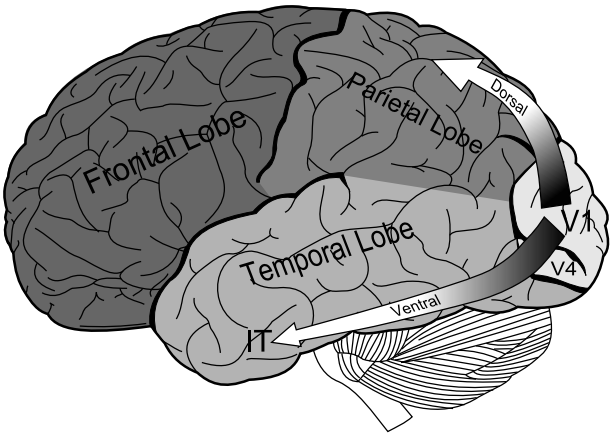


Figure 1: The primary visual cortex (V1) is the source of two cortical visual streams. The ventral stream is directed to the inferior temporal (IT) area and underlies object recognition. The dorsal stream is directed to the posterior parietal area and underlies spatial perception.

zation of the LGN cells is retinotopic, which means that a single point in the visual field is represented by cells above and below each other in the layered structure.

The majority of the innervations from the LGN projects into the primary visual cortex (PVC), also known as V1 (see Fig. 1). Within primary visual cortex the image continues to have a retinotopic organization, with more neurons allocated to the representation of the center of the visual field (*cortical magnification*). Center-surround cells are still present in the primary visual cortex but at least two other classes of neurons can be found, that are *simple cells* and *complex cells* (HW62). Simple cells respond to the presence of edges at particular locations and orientations in the visual field. Complex cells represent a more abstract type of information which is often independent of the location in the visual field.

Until this point we have analyzed the various stages which constitute the early vision. At this point the image is still represented in terms of local features such as orientation, wavelength and motion, but it is still lack-

ing a global structure and there is no indication of what are the neural representations of the *objects* contained in the image. The first steps towards such high level representation require (i) determining which pieces of the image *go-together*, i.e. belong to the same object, (ii) inferring the true object color and (ii) determining a motion pattern that is coherent with the object representation and the viewer motion. Most of these tasks are taken over by the *extrastriate cortex* (visual association cortex) surrounding the primary visual cortex. Color perception as we experience it consciously does not appear until area V4 (see Fig. 1). Following the responses of cones through the visual pathways, color information reaches area V2 and, at this point, it cannot be related with "real" color perception, which is a context dependent task, but only with wavelength perception, which can be influenced by the spectral composition of the incident light. This means that colors as we see them are the result of a process which disentangles the ambient light from the spectral reflectance of the object: area V4 is the site where this takes place.

The perception of motion seems to be mediated by two types of neurons. *Component selective* neurons, which can be found in V1 and in the middle temporal area (MT), respond to edges with a particular orientation and that are moving along in a particular direction. *Pattern selective* neurons, on the other hand, can be found only in MT and seem to pool the outputs of component selective neurons in order to detect particular patterns of movement.

Until this point it has been described how higher level features such as color and motion are detected, but nothing has been said about how this information is put together to generate the perception of an object. Proceeding along the *ventral stream* (see Fig. 1), which departs from V1 passing thorough V2 and V4, we finally reach the *inferior temporal* cortex (IT), which is considered the site of object recognition processes. As one proceeds along the ventral stream from earlier to latter areas, neuronal properties change: the size of the receptive fields increases, indicating more information integration, and the complexity of visual processing grows. While V1 cells respond to local edge orientation, neurons in IT respond selectively to object features. In Von der Malburg's hypothesis

(dM85), this distributed object representation is bound together by the synchronization of the neural activation of the feature detectors. Moreover, top-down interactions with higher cognitive areas such as the prefrontal cortex, i.e the site of working memory, play a key role in the process of object recognition (DD95). Parallel to the ventral stream, the dorsal stream (see Fig. 1) departs from the primary visual cortex and reaches the posterior parietal cortex. Its role seems to be more connected with spatial perception and visuomotor control. As in IT cortex, receptive fields of parietal neurons are very large, denoting high complexity of visual processing. Moreover, also parietal cortex appears to interact tightly with the prefrontal cortex.

2.2.2 Cortical Mechanisms of Visual Memory

The ability of humans and animals to learn from experience is recognized to be supported by multiple memory systems with different functional characteristics and diverse neural bases (Squ92). A fundamental distinction between memory systems differentiates between declarative and non-declarative (implicit) models: the former supports the explicit recollection of objects, places or events and grounds primarily on the structures of the medial temporal lobe (Des96). Implicit memory, on the other hand, does not require conscious awareness and relates primarily to skill acquisition (i.e. motor, perceptual and cognitive skills), habit formation, priming, emotional learning and, in general, all the knowledge that is more related to performance than to explicit recollection (Squ92). In this section, we briefly review the non-declarative memory models that relate to the acquisition of visual skills and perceptual knowledge, focusing primarily on the cortical machinery that represents the neural correlate of this from of implicit memory.

In biological vision systems, the neural mechanisms for learning and memory are continually modulating and altering the representation of perceptual stimuli in the cortex. *Perceptual learning* is a key behavioral phenomenon of implicit visual memory that is concerned with the improvement in one's ability on a variety of simple sensory tasks following

practice (TG04a). Perceptual learning is a process that continuously, and unconsciously, adapts the neural representation of the visual stimuli, allowing the visual cortex to assimilate knowledge about specific familiar patterns (TG04a). It operates on neural representation at various level of the visual information processing pathway, undertaking a wide spectrum of perceptual tasks, ranging from the identification of primitive visual attributes, such as position, orientation, texture and shapes, to the detection of complex geometric shapes and alphanumeric characters.

Priming is a behavioral phenomenon that is closely related to perceptual learning. This non-conscious form of memory improves the identification of an object or word by experience with that object or word. In particular, people are faster and more accurate at naming or reading repeated stimuli with respect to new stimuli. Perceptual priming is not affected by small variations in orientation, texture, color and size (WM98). For instance, a fragmented image of an object, which appears meaningless at first sight, can be easily identified after the presentation of an unfragmented picture of the same object. Priming seems to have long lasting results both on normal and amnesic subjects (WM98) and its effects, similarly to perceptual learning, appear to be modulated by the number of stimulus repetitions. Further, evidence suggests that the degree of attention devoted to a task does not affect the magnitude of priming (Des96; WM98).

The two behavioral phenomena described above have been recently correlated with three commonly observed neuronal effects in memory demanding tasks, that are *repetition suppression*, *enhancement* and *delay activity* (Des96). In Repetition Suppression (RS), the repeated presentation of a visual stimulus induces both a short-term and a long-term suppression of the neuronal responses in subpopulations of visual neurons. Repetition suppression is not dependent on the behavioral significance of the stimuli and is completely unconscious. Brain imaging studies have shown that, under conditions which lead to priming, the repetition of visually presented objects induces reduced activation in the cortical areas. In (Des96) it is hypothesized that a smaller population of activated neurons connected with a better task performance suggests that a smaller

representation of the stimulus, associated with a sharpening of the neuronal selectivity, is a better representation.

Enhancement works in an opposite fashion, in that neuronal responses are enhanced for objects with learned behavioral relevance. For instance, in a delayed-matching-to-sample (DMS) experiment, a subject is asked which visual stimulus (or stimuli combination) matches the target among those that are presented. Monitoring the IT cortex in a DMS task shows that the cortical neurons can distinguish between matching and non-matching samples (Des96). Clearly these results cannot be explained in terms of the repetition suppression mechanism, since RS does not distinguish between behaviorally relevant and irrelevant stimuli. In particular, the DMS experiment shows that neurons that normally respond to a given stimulus A have a stronger response when A is the target stimulus, whereas all the other cells experience suppression. Moreover, the neuronal responses are enhanced only when the stimulus matches the sample, while they are not influenced by the non-matching (i.e. behaviorally insignificant) repetitions of the stimulus.

Delay activity (DA) is another cortical mechanism which has a fundamental role in the DMS task, since it maintains the memory of the behaviorally relevant samples. The delay activity at the level of the IT cortex has only limited temporal scope, since it reduces to the association of coupled stimuli. For instance, a group of cells responsive to B in a task in which is requested to associate A and B will show elevated activity in the delay between A and B. These results suggest that DA in IT cortex may be associated with the representation of what is behaviorally important at a given moment of time. In other words, it may account for contextual information within a limited temporal horizon.

Enhancement and delay activity seems to depend strongly on feedback from the prefrontal cortex to the temporal cortex and are thought to be important for active visual working memory. Conversely, repetition suppression appears to be an intrinsic property of visual cortical areas such as inferior temporal cortex and is thought to be a key neural correlate for perceptual learning and priming (Des96). Together, these cortical mechanisms bias the competitive interactions that take place between

stimuli representation in the cortex, possibly influencing visual attention. In the next section we delve more into the details of repetition suppression, analyzing three hypotheses that have been presented in the literature to account for the RS phenomenon and providing motivations for the development of a learning model inspired by this cortical memory mechanism.

2.2.3 The Repetition Suppression Mechanism

The finding that perceptual learning and priming constitute separate memory mechanisms, that operate through different rules with respect to episodic memory, has motivated a whole body of research aimed at revealing the underlying mechanisms regulating its activity. A particularly interesting finding concerning perceptual learning is that while people show improved perception and recognition abilities for repeated stimuli, the overall neural activity is reduced as a result of the repeated pattern presentation. This phenomenon is known as *repetition suppression* (RS) and appears to be fundamental for mediating perceptual learning and priming (Des96; WM98). Repetition suppression induces long-lasting changes to the visual cortex, decreasing the neural activity as a consequence of the repeated presentation of similar stimuli. Neurophysiological evidence has shown that its effects can be observed also when the repeated stimulus is presented at different retinal locations and in case of variations to the visual stimulus geometry. Like priming and perceptual learning, repetition suppression does not depend on the behavioral significance of the stimuli, i.e. it is not specifically linked to the active maintenance of a sample in memory, nor does it require any form of response/reward signal. Furthermore, RS is a process that operates unconsciously, since its effects can be recorded also in anesthetized subjects (Des96).

In the literature are reported three models that have been proposed to account for the repetition suppression phenomenon (see Fig. 2)

1. *Fatigue.* In this model all the neurons initially responsive to a stimulus experience a proportionally equivalent suppression. This change affects the mean population firing strength but preserves the pattern

of relative responses across the neurons (MD94).

2. *Sharpening.* According to this model only some neurons experience repetition suppression. In particular, neurons coding irrelevant features with respect to the particular stimulus at hand experience suppression. This process is fundamentally a learning mechanism that enhances the sharpness and distributivity of the stimuli representation (Des96; WM98).

3. *Facilitation.* This model accounts for the faster processing time of repeated stimuli, by causing shorter latencies and durations of neural firing (SR96; JG05).

We explore the sharpening model hypothesis, since in our opinion it has the most suitable characteristics to be used as a learning process. Moreover, from the neurophysiological point of view, the sharpening model is the one that best explains the behavior of repetition priming at the level of IT cortex and V4 (GSHM06). In this model, the repetition suppression phenomenon induces a sharpening of the neural representation of items by means of an overall reduction of the number of active neurons which is counterbalanced by the steepening of the response of the most item-selective neurons. This process seems to be aimed at the selection of neurons that act as detectors of the most informative features. Moreover it appears that such a process may facilitate novelty detection, since more familiar stimuli experience more suppression than unfrequent items.

We suggest that this form of repetition suppression provides interesting hints for the development of innovative learning schemes and mechanisms. The literature in this field offers few works that exploits the neurophysiological issues described so far. The *activation sharpening* model by French (Fre92), for instance, extends the standard backpropagation with an extra step in which the response of the most active hidden nodes is increased slightly for each pattern, while the other nodes activations are decreased. French's learning rule, however, does not achieve sharpening by exploiting repetition. Norman and O'Reilly (NO03), on the other hand, devise a competitive network to simulate the repetition-driven sharpening of neural representations within an hippocampal model. Initially

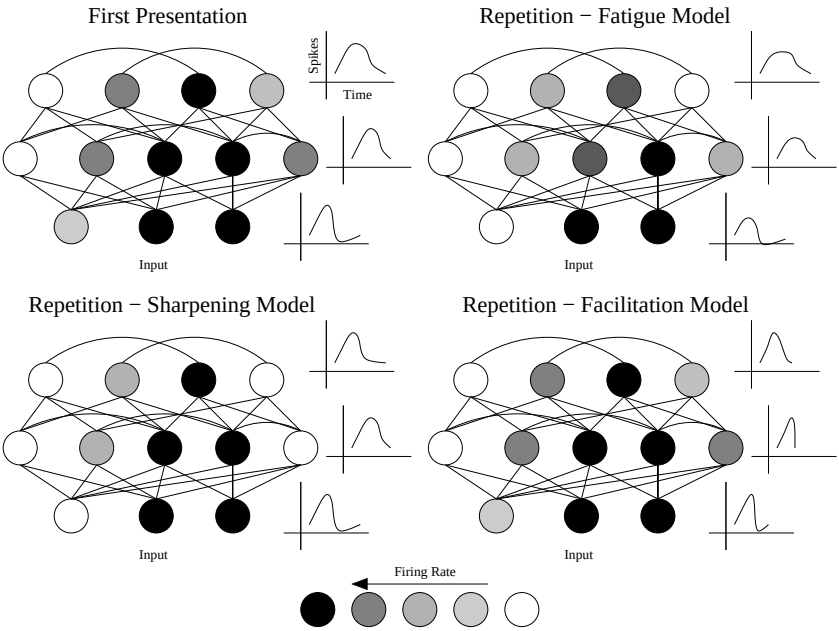


Figure 2: Graphical exemplification of the three repetition suppression hypotheses: the fatigue model (top-right, lower firing strengths), the sharpening model (bottom-left, fewer neurons responding) and the facilitation model (bottom-right, shorter duration of neural processing). The particular connection topology of the exemplar neural network is not relevant for the description of the RS mechanism. The networks depict the activation of the neurons subject the repeated presentation of an input x, under the three RS hypothesis. The neural activity (i.e. the output) is represented by the grey levels in the circles: darker gray levels identify more active neurons. The graphs, on the other hand, depict neural spikes with respect to time. Adapted from (GSHM06).

many neurons respond weakly to a distributed input pattern representing the stimulus. Through learning, a fixed percentage of neurons is selected to be reinforced while the remainder is inhibited by using lateral connections. This model focuses more on familiarity detection than on feature selection and applies only to a very specific network model.

The learning scheme we propose in the following is inspired by the repetition suppression phenomenon and focuses in particular on neuron and feature selection. Our idea is to devise a broad scope model that can be applied to a wide range of neural network models and learning systems on real world tasks. For this reason we do not model explicitly the novelty detection aspects of repetition suppression, whereas we devise a flexible learning scheme that can be applied to several neural models and that can, eventually, be instantiated to reproduce the novelty detection behavior.

2.3 Background

Before delving into the details of the proposed model, we review key related work and those contributions in the literature from which our model draws inspiration. In particular, we describe hard and soft competitive learning models as well as related clustering algorithms dealing with the cluster number identification problem.

2.3.1 Competitive Learning

The general term *Competitive learning* identifies a broad class of online neural network models that generate data partitions by means of an unsupervised learning strategy based on some kind of competition between the activation levels of the units composing the network. The typical competitive learning scenario compromises of a set on units $U = \{u_1, \ldots, u_i, \ldots, u_N\}$ characterized by an activation function that determines the response of the neurons to the patterns in the data set $\chi = \{x_1, \ldots, x_k, \ldots, x_K\}$. Each unit has an associated weight vector c_i, known also as *prototype*, that is iteratively modified by the learning process. Associated with a weight

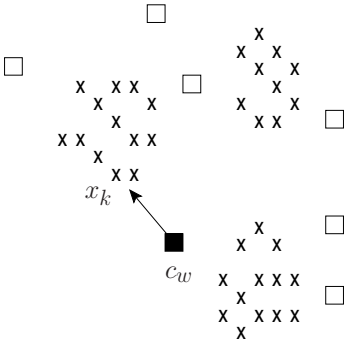


Figure 3: Competitive learning example: the weight vector c_w of the BMU (■) is moved towards the current pattern x_k.

vector c_i there is a *Vororonoi region* C_i that is the set of vectors in the reference feature space $\mathcal{F}$ (usually $\mathcal{F} = \mathbb{R}^d$, but $\mathcal{F}$ can be, e.g., every kernel induced space (SMB+99)) for which c_i is the nearest neighbor; in other words,

$$C_i = \left\{ x \in \mathcal{F} | i = \arg\min_j \|c_j - x\|^2 \right\}. \tag{2.1}$$

The earliest competitive approaches adopted a Winner Takes All (WTA) learning strategy, where a single neuron, called the Best Matching Unit (BMU), is allowed to update its weight vector for each pattern presentation. The basic WTA competitive learning model determines the BMU by seeking the neuron u_i whose weight vector c_i is the closest to the input x_k with respect to the Euclidean norm (see Fig. 3). More formally, the algorithm can be summarized by the following steps

0 - Sample At each learning step randomly pickup a sample x_k from the dataset.

1 - Activation For each unit u_i calculate the activation as

$$h_i = \begin{cases} 1 & \text{if } i = w \text{ s.t.} \quad w = \arg\min_j \|x_k - c_j\| \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

2 - Update For each unit u_i, update the prototype as

$$c_i^t = \begin{cases} c_i^{t-1} + \alpha(t)h_i(x_k - c_i^{t-1}) & \text{if } i = w \\ c_i^{t-1} & \text{otherwise} \end{cases} . \tag{2.3}$$

where $\alpha(t)$ is the time-dependent learning rate regulating the magnitude of the weight update and is assumed to decay to zero as t increases as to ensure convergence of the algorithm (YZG92). The procedure described above essentially yields to an online analog of the popular k-means, also known as the Linde-Buzo-Gray (LBG) algorithm (LBG80). In fact, it can be easily show (AKCM90) that it implements a steepest descent optimization of the LBG quantization error

$$J = \frac{1}{K} \sum_{i=1}^{N} \sum_{x_k \in C_i} \|x_k - v_i\|^2 \tag{2.4}$$

where C_i identifies the set of patterns x_k for which u_i is the BMU, i.e. the *Voronoi cell* induced by the weight vector c_i.

The earlier WTA models are prone to produce under-utilized neurons, that is the so called *dead-unit* problem (RZ86). Figure 4 portrays a typical scenario showing neuron under-utilization: the weight vector c_1 is a winner for both the top-most clusters and starts oscillating between the two, while c_2 never wins for any sample and is thus a dead-unit. As a consequence, the two clusters are not correctly identified and the algorithm produces a sub-optimal solution. To address this problem, (DeS88) proposed the use of a *conscience* mechanism for making frequently winning neurons less likely to win in the future. Frequency Sensitive Competitive Learning (FSCL) (AKCM90), for instance, solves the dead-unit problem by modifying (2.2) to compute the best matching unit u_w as

$$w = \arg\min_j \frac{n_j}{\sum_i n_i} \|x_k - c_j\| \tag{2.5}$$

where n_j is the number of patterns for which unit u_j was the BMU. While FSCL can deal effectively with the dead-unit problem, it cannot address the issue of estimating the appropriate number of clusters from the data. In particular, if FSCL is initialized with a network size that is larger than

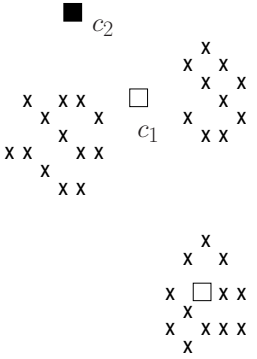


Figure 4: The neuron under-utilization problem: weight c_2 (■) is frozen in its position (dead-unit) since c_1 is always the BMU for the samples in the 2 top-most clusters.

the *actual* number of clusters, then it will distribute the unit in excess on the data introducing a distortion in the estimation of the clusters' centroids. Several competitive learning algorithms have been proposed to address this issue. One of the earliest and most effective approaches is Rival Penalized Competitive Learning (RPCL) (XKO93). The key idea of RPCL is that, for each pattern, not only the winner is updated so to approach the input pattern, but also the second winner (i.e the *rival*) is deflected from it, with a step size that is proportional to the Euclidean distance between the rival's prototype and the input pattern (see Fig. 5). More formally, the RPCL algorithm consists of the following steps

0 - Sample At each learning step randomly pickup a sample x_k from the dataset.

1 - Activation For each unit u_i calculate the activation as

$$h_i = \begin{cases} 1 & \text{if } i = w \text{ s.t. } \quad w = \arg\min_j \gamma_j \|x_k - c_j\| \\ -1 & \text{if } i = r \text{ s.t. } \quad r = \arg\min_{j \neq w} \gamma_j \|x_k - c_j\| \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

where c_i is the prototype of the i-th cluster, u_w and u_r are the winner and the rival, respectively. The term $\gamma_j = n_j / \sum_i n_i$ is the con-

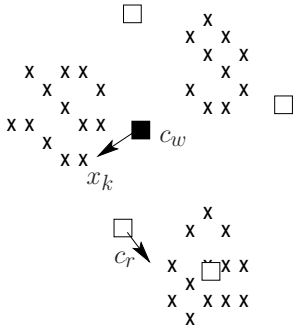


Figure 5: Rival Penalized competitive learning example: the BMU (■) is moved towards the current pattern x_k, while the weight of the second winner (i.e. c_r) is deflected from x_k.

science parameter used to reduce the likelihood that a frequent winner is the best matching unit.

2 - Update For each unit u_i, update the prototype as

$$c_i^t = \begin{cases} c_i^{t-1} + \alpha_w(t) h_i (x_k - c_i^{t-1}) & \text{if } i = w \\ c_i^{t-1} + \alpha_r(t) h_i (x_k - c_i^{t-1}) & \text{if } i = r \\ c_i^{t-1} & \text{otherwise} \end{cases} \tag{2.7}$$

where $\alpha_w(t)$ and $\alpha_r(t)$ are the learning and de-learning rate, with $\alpha_w \gg \alpha_r$.

RPCL uses the rival de-learning strategy to make sure that each cluster is assigned to exactly one prototype, while exceeding units are driven away from the data. Due to its reduced computational complexity and the ability to incrementally determine the cluster number from the data, RPCL has been applied extensively to various cluster number identification problems encompassing gene expression data analysis (NZF+03), object detection (Xu07), software component classification (NS04) and environmental data clustering (ACFV03).

Self Organizing Maps (SOM) (Koh82) introduced the idea of *soft competitive learning* by modifying the competition strategy to allow multiple

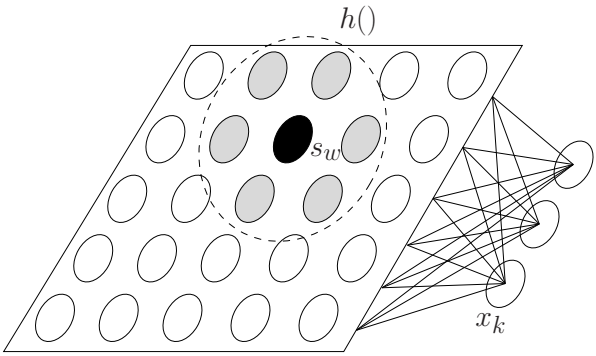


Figure 6: Self Organizing Map. Neurons are arranged in a grid of fixed size and their position on the grid, i.e. s_i is used to determine the co-activated neurons (shown in gray) with respect to the best matching unit in s_w (shown in black).

winners for a given input. A SOM is neural network whose units are organized on a grid with fixed topology (see Fig. 6): in addition to the weight vector c_i the neurons are characterized by a fixed position on the grid s_i. Given an input pattern, the SOM identifies the BMU as in standard competitive learning, e.g. the unit whose vector is the closest to the pattern in Euclidean sense. The BMU activation is then propagated along the grid based on a neighborhood function and both the BMU and its neighbors are updated. In particular, the BMU receives full credit for adaptation, while its neighbors are allowed to learn proportionally to their spatial contiguity with the best matching unit. This process constrains unit learning in such a way that, at convergence, grid-neighbors respond to similar input patterns, thus realizing a topologically ordered map of the input distribution.

Summarizing, the SOM algorithm is the following:

0 - Sample At each learning step randomly pickup a sample x_k from the dataset.

1 - Activation Determine the BMU u_w using $w = \arg\min_j \|x_k - c_j\|$ or the version with conscience in (2.5). For each unit u_i, compute the

activation using the neighbor function

$$h(d_{w,i}) = \exp\left(\frac{d_{w,i}}{\sigma(t)}\right) \tag{2.8}$$

where $d_{w,i}$ measures the grid-distance between the BMU u_w and the unit u_i; $\sigma(t)$ is decreasing function of t ensuring that the size of the neighborhood reduces with time.

2 - Update For each unit u_i, update the prototype as

$$c_i^t = c_i^{t-1} + \alpha(t)h(d_{w,i})(x_k - c_i^{t-1}). \tag{2.9}$$

Based on the SOM, several other topology preserving, soft competitive, models have been proposed. Of particular interest are the *Neural Gas* (NG) (MS91) and *Growing Neural Gas* (GNG) (Fri94; Fri95) networks: both models release the constraint that units are organized in a grid. Rather, they determine the BMU neighborhood and the adaptation strength based on distance ranks: each time a pattern x_k is presented to the network, all the units u_i are ranked in order of increasing distance between their weight vector c_i and x_k. Given r_i as the rank of the i-th unit, the update rule is given by

$$c_i^t = c_i^{t-1} + \alpha(t)h(r_i)(x_k - c_i^{t-1}). \tag{2.10}$$

where $h(r_i)$ plays the role of SOM's neighborhood, being a decreasing function of r_i, e.g. $h(r_i) = \exp(-r_i/\lambda(t))$ with $\lambda(t)$ decreasing with time. The key difference between NG and GNG is that while the former network has a fixed size that has to be specified a priori, the latter is capable of estimating the network size during learning. In particular GNG starts with two neurons and iteratively adds (i.e. *grows*) units based on local error measures gathered during the adaptation process (see (Fri95) for further details).

All the models described so-far implements soft competition by means of neighborhood function that determines how much learning is propagated from the BMU to its neighbors. The first full *soft competition scheme*

(SCS) has been formalized in (YZG92). Instead of updating only neighboring neurons, SCM defines an activation function that distributes the learning credit to all the units in the network proportionally to their local activation levels. In other words, the update rule for c_i is given by

$$c_i^t = c_i^{t-1} + \alpha(t)h(x_k, c_i^{t-1})(x_k - c_i^{t-1}). \tag{2.11}$$

where the credit assignment function $h(x_k, c_i^{t-1})$ is defined as

$$h(x_k, c_i^{t-1}) = \frac{\exp -\beta(t)\|x_k - c_i^{t-1}\|^2}{\sum_j \exp -\beta(t)\|x_k - c_j^{t-1}\|^2}. \tag{2.12}$$

and

$$\lim_{t\to\infty} \beta(t) = \infty. \tag{2.13}$$

Hence, upon each pattern presentation, all the weight vectors are simultaneously updated such that c_i is shifted a fraction $h(x_k, c_i^{t-1})$ towards x_k. This is a soft competition scheme in the sense that there is no ultimate winner; rather, each prototype is updated towards the data with a step size that is proportional to its probability of winning. The annealing scheme of $\beta(t)$ ensures that at the beginning of learning no neuron is attracted by a particular partition: as training develops the scheme converges towards a WTA scheme where each unit tends to win for a single partition.

The SCS model has inspired the development of several soft competitive algorithms. For instance, Xu (Xu07; Xu03) has recently presented a divide-and-conquer framework for multi-agents that extends the RPCL model to soft competitive learning strategies. In particular, the generalized RPCL model provides a sharing mechanism that distributes learning and penalization to groups of agents proportionally to their activation strengths. In other words, RPCL is extended to a soft-competitive mechanism comprising multiple winners and rivals so to minimize the following global error criteria

$$\epsilon(\theta) = \sum_{k=1}^{K}\sum_{i=1}^{N} p_{k,i}\epsilon_k(\theta_i) \tag{2.14}$$

where $\epsilon_k(\theta_i)$ is the individual criteria that evaluates the performance of the i-th agent (having parameters $\theta_i = \{c_i, \dots\}$) on the observed sample x_k. The term $p_{k,i} = q_{k,i} - \alpha\pi_{k,i}$ provides the sharing mechanism by balancing the amount of positive learning q_{k_i} and penalization $\pi_{k,i}$. In the general model, both q_{k_i} and $\pi_{k,i}$ are weighted sums of the exponentiation of ϵ_k (for more details refer to section 3.1.2 in (Xu07)). The agent error can be specified either as the standard Euclidean distance $\epsilon_k(\theta_i) = \|x_k - c_i\|$ or, in a probabilistic form, as $\epsilon_k(\theta_i) = -ln(p(x_k|\theta_i))$, where $p(\cdot)$ is, for instance, given by a Gaussian centered in c_i.

A competitive neural network based on a radically different learning strategy is the *Adaptive Resonance Theory* (ART) model (CG88). This family of algorithms has been developed specifically to address the *stability-plasticity dilemma*, that is how to design a learning system so that it remains plastic, i.e. capable of learning, in response to exogenous stimuli, while maintaining a stable neural representation of the significant events experienced in the past, i.e. avoiding to forget previous knowledge. As in other competitive learning models, an ART network consists of a set of weight vectors that store the neural representation of the input stimuli. The ART family comprises various models, often with substantially different network architectures, sharing the common belief that adaptation is only possible within a state of *resonance*. In particular, if an input vector matches closely an existing ART prototype, it achieves resonance and the prototype is updated to match the particular input to an higher degree. The activation of the resonant state is regulated by a *vigilance parameter* that determines how closely the input vector has to match an existing prototype. In a sense, the vigilance parameter determines the sizes of the detected perceptual classes: a low vigilance value produces few large classes, while a larger number of finer classes is obtained as the network vigilance increases. If the ART network does not enter the resonant state because no prototype can pass the vigilance test, then it triggers a neuron recruiting strategy. This procedure dynamically activates silent neurons from a pool of *uncommitted* units and uses them to represent the novel input (i.e. a stimulus that hasn't a close-enough representation in the network). In essence, an ART network is initialized with all uncommitted

neurons and dynamically activates them whenever a novel, surprising stimulus is encountered. This neuron enabling process continues until there are no more committable neurons: in this case, a novel input produces no response. ART leaning strategy addresses the stability-plasticity dilemma, since it allows a certain degree of neural adaptivity while controlling the degree of forgetting of the network through the vigilance parameter.

2.3.2 Cluster Number Identification

Clustering is a popular task that in the last 30 years has received much attention from the scientific community. Due to the large number of works in the literature it is particularly arduous, as well as beyond the scope of this thesis, to give a comprehensive overview of the works in the area. In this chapter, we focus primarily on a central issue that rises in clustering, that is how to estimate the appropriate number of data partitions. Therefore, in the remainder of this section, we will center our review on those clustering models that have been devised to explicitly address the cluster number identification problem. For a broad discussion on clustering models we refer the reader to survey papers such as (XW05) and (JMF99).

Taking an high level look at the approaches in the literature, we can roughly classify the algorithms in two classes depending on whether they perform cluster number identification as a second stage with respect to parameter learning or jointly with the model fitting process. Approaches from the former class, often generate a number of candidate models for different values of the cluster number; then, they select the best candidate model based on some kind of validity measure. This approach is particularly common in probabilistic mixture models: here data is assumed to be generated by a number of random sources and the mixture components are fitted to the data so to locate such sources. Finding the number of data sources, in this context, is equivalent to fitting the mixtures by optimizing an information criterion. Usually, the *Expectation Maximization* (EM) (DLR77) algorithm is used to estimate the model parameters for a fixed number of components: then the fitted model which optimizes the infor-

mation criterion is selected as the final candidate. Among the most popular model-selection indices is the Bayesian Information Criterion (BIC) (Sch78), that is

$$BIC = -2\ln(L) + N_p \ln(K) \tag{2.15}$$

where K is the sample size, L is the maximum likelihood estimated for the model and N_p is the number of parameters in the mixture. In this case, lower BIC values imply either fewer model parameters, better likelihood, or both: hence, the winning model is that minimizing the BIC. Fraley and Raftery (FR98) describe an application of BIC model selection to a Gaussian mixture. The Akaike Information Criterion (AIC) (Aka74)

$$AIC = N_p - 2\ln(L), \tag{2.16}$$

has been proposed earlier than BIC to describe the tradeoff between bias and variance: its formulation is quite similar to that in (2.15), but AIC penalizes free parameters less strongly than BIC does. Reference (WC92) describes an algorithm for estimating the number of mixtures using the AIC index. Other model selection criteria exist, such as the Minimum Description Length (MDL) (Ris96) and Minimum Message Length (MML) (OBW96): for a comparison and a comprehensive overview refer to (MP00). An interesting approach founding on the information bottleneck model is presented in (SB04): this approach exploits information curves, defined in terms of mutual information, to determine the maximum number of resolvable clusters in the data; the distortion introduced by the sampling noise is taken into account directly in the formulation of the objective function, thus bounding the number of clusters that can be resolved from the data without overfitting.

Apart from these information-theoretic approaches, cluster number determination can be performed by means of model selection criteria tailored to the particular unsupervised learning model at hand. In kernel-based algorithms, for instance, the cluster number is often estimated based on some form of factorization of the *kernel matrix*. Each element k_{lm} of the kernel matrix (known also as *Gram matrix* (SSM98)) identifies the dot-product distance in the kernel-defined feature space between the samples x_l and x_m. The general intuition is that the kernel matrix will show a

block diagonal structure whenever there are groupings within the data set. Girolami (Gir02) proposed a kernel clustering algorithm where cluster number is estimated based on the eigenvalue decomposition of the kernel matrix K, that is $K = U\Lambda U^T$, where the columns of the matrix U are the individual eigenvectors u_l of K and the diagonal matrix Λ contains the associated eigenvalues λ_l. Within this decomposition there will be a certain number of dominant eigenvectors that can be used as an estimate of the number of distinct clusters in the data set.

The class of clustering algorithms that performs joint parameter estimation and cluster number identification is wider than that of the two-stage models and further differentiates between several diverse approaches. Some of them take on a divisive strategy that begins the training phase with a single weight vector and incrementally adds new cluster detectors as learning proceeds. This is the case, for instance, of *Self-Splitting Competitive Learning* (SSCL) (ZL02), that generates an estimate of the number of clusters based on a one-prototype-take-one-cluster (OPTOC) paradigm and a self-splitting validity measure. The OPTOC paradigm is an hard competitive learning strategy that uses biasing to ensure that each unit focuses on a single cluster while ignoring the rest of the data. To achieve this, OPTOC defines an *Asymptotic Property Vector* (APV) that guides the learning of the unit prototype. In practice, the APV A_i and the weight vector c_i together determine a circular learning neighborhood in input space, that ascertains a set of "preferred" stimuli for the i-th neuron. Initially, this neighborhood is large, implying that the weight vector and the APV are far. As learning progresses, this size reduces to zero, guaranteing convergence of the prototype to the center of the cluster. To achieve this, the OPTOC update rule ensures that patterns outside of the dynamic neighborhood contributes less to learning as compared to the inside patterns. The neighborhood size is dynamically adjusted by moving both the APV and weight vector of the BMU towards the current input vector x_k.

The OPTOC process ensures that the single units do not start oscillating between different clusters; however, alone OPTOC cannot be used to estimate the appropriate number of clusters. To achieve this, SSCL

iteratively chooses one of the existing prototypes (initially, the only prototype) to split it into two weight vectors based on a split validity measure. This criterion measures the distortion between the OPTOC solution and a purely WTA solution (e.g. obtained by k-means): if this distortion is large, then there are data samples that have been ignored by OPTOC. SSCL thus selects the prototype with higher distortion and splits it so that the new weight vector can, eventually, take care of the ignored samples. This self-splitting behavior terminates if no prototype is suitable for further splitting.

Other approaches adopt a progressive clustering model, where the number of possible partitions exceeds the actual data distribution and exceeding clusters are merged as learning proceeds. The *Competitive Agglomeration* (CA) clustering (FK97), for instance, produces a sequence of partitions with a decreasing number of clusters by means of an error function that creates a competitive environment in which units compete for the samples and only *significant* neurons survive. The loss function of competitive agglomeration clustering is defined as

$$J_{ca}(\chi) = \sum_{i=1}^{N}\sum_{k=1}^{K}(u_{ki})^2 d_{ki}^2 - \alpha \sum_{i=1}^{N}\left[\sum_{k=1}^{K} u_{ki}\right]^2 \tag{2.17}$$

subject to the constraints

$$u_{ki} \in [0,1], \quad 1 < \textstyle\sum_{k=1}^{K} u_{ki} < K, \quad \displaystyle\sum_{i=1}^{N} u_{ki} = 1 \tag{2.18}$$

and where the term d_{ki} measures the distance of the feature point x_k from the prototype c_i. Together, equations (2.17) and (2.18) recall a typical fuzzy clustering scenario, where the matrix $U = [u_{ki}]$ determines a fuzzy partition of the samples x_k in N clusters, with fuzzy membership given by u_{ki}. However, (2.17) differs from a typical fuzzy c-means (Bez81) loss function for the second term in the subtraction: while the first term controls the shape and size of the clusters and encourages partitions with many clusters, the second one penalizes solutions with a large number of partitions and encourages the agglomeration of clusters. As a consequence of the combined action of the two terms (mediated by the weight

α) CA produces results that minimize the sum of intra-cluster distances, while partitioning the data into the smallest possible number of clusters. The rules for prototype and membership update are obtained by constrained minimization of (2.17) using Lagrange multipliers (see (FK97) for the details of the derivation steps).

The definition of the competitive agglomeration algorithm is flexible enough to incorporate different distance measures in the loss function. For instance, the same authors of CA, proposed the *Robust Competitive Clustering* (RCA) (FK99) algorithm. RCA estimates the cluster number by means of a competitive agglomeration process, while achieving immunity to noise and outliers by incorporating robust loss functions in the definition of the RCA error. In (FN04), on the other hand, is presented a competitive agglomeration strategy that achieves simultaneous clustering and attribute discrimination (SCAD): in other words, SCAD determines the optimal prototypes c_i along with a relevance measure, or feature weight, of its covariates c_{il}.

The *Similarity-based Clustering Method* (SCM) (YW04) takes an alternative approach with respect to progressive clustering: instead of exploiting competitive agglomeration to get rid of overestimated clusters, SCM exploits an Agglomerative Hierarchical Clustering (AHC) procedure to estimate the local optimal cluster number as well as their volumes. The SCM learning scheme is set up by maximizing the total similarity measure J_{scm}, that is

$$J_{scm}(\chi) = \sum_{i=1}^{N} \sum_{x_k \in \chi} \exp\left(-\frac{\|x_k - c_i\|}{\beta}\right)^{\gamma} \tag{2.19}$$

where the power parameter γ is used to take over the effect of the β parameter so that it can be initialized as the sample variance. First, SCM finds an approximated density shape for the data set by estimating γ such that it optimizes J_{scm}; then, it seeks the optimal value for the prototypes c_i, that correspond to the peaks of the objective function J_{scm}. Finally, SCM uses AHC to build a hierarchical clustering tree of the peaks c_i that is used to select the final cluster number. The agglomerative hierarchical clustering method is quite general, so it has been used to determine

the optimal cluster number in mode seeking algorithms other than SCM, including, for instance, the popular *Mean Shift* clustering (WY07).

Previously, it has been pointed out how finite mixture models can be used together with model-selection criteria to identify the number of random sources generating the data samples. Figueiredo and Jain (FJ02) proposed an alternative approach to mixture models clustering by exploiting the intuition that a component with a null mixing weight is essentially indistinguishable from a non-existing component. Therefore they devised a *Gaussian Mixture Model* (GMM) that uses a variant of the *Expectation Maximization* (EM) algorithm to learn to annihilate the mixing components of irrelevant mixtures, while positioning the remaining mixtures on the data clusters. To achieve this they modify the log-likelihood by adding a term derived from a Minimum Description Length criterion: the update equation resulting from the EM optimization are shown to be able to identify the data sources, starting from an overestimation of the actual cluster number.

So far, we have described models that adopt either the divisive or the agglomerative approach. However, in the literature there are models using an hybrid strategy that mixes both approaches. *Circular K-means* (CK-means) (Cha05), for instance, determines the number of clusters by means of a sequence of split and merge operations that are taken based on a modified version of the Variance Ratio Criterion (VRC) index. Essentially VRC measures the ratio among the *between clusters distance* and the *within clusters distance* and the number of clusters is determined as the configuration that maximizes the VRC. As a consequence, an existing cluster is split into a number of new clusters as long as its VRC keeps increasing, whereas, at each iteration, the two closest prototypes are merged provided that the joining operation increases the VRC. *Variational Bayesian Splitting* (VBS) (CL07) applies a similar merge-split approach to Gaussian Mixture Models trained by a variational optimization procedure. With respect to the mixture model in (FJ02), VBS treats the model selection problem locally by restricting the number of components that are updated at each learning step. In particular, for each input pattern, only those components falling in the same region of the sample are allowed to update

their parameters, i.e. they are *free*, while the rest is kept fixed. During the optimization process, some of the *free* coefficients converge to zero and are canceled from the mixture similarly to (FJ02). However, differently from the GMM model, VBS is initialized with a single component and incrementally adds new components to the mixture by means of a splitting procedure. In particular, at each step, one component is selected and replaced by two subcomponents: at this stage only the two subcomponents are considered free while the rest is held fixed. Both components are tested against the data in their region and, if the test suggests the existence of more than one cluster, then both subcomponents are retained and the size of the mixture increases. If this test fails, then one of them is eliminated and the initial component is recovered. This process is iterated for all the components in the mixture and terminates solely when all the components fail the splitting test. In the case where a successful split is encountered, a new round of splitting tests is initialized for all the components.

As a concluding remark, we can observe that, in general, two-stage algorithms have clearly higher computational impact with respect to single-stage models, due to the fact that they often fit several models before choosing the cluster size. On the other hand, the calculation of the regularization terms and the split-merge process is often a key factor determining the computational complexity of the of single stage algorithms. Our work focuses on a progressive clustering model that targets at estimating the unknown cluster number from the data by means of a bio-inspired penalization scheme characterized by a lower computational complexity with respect to popular clustering approaches.

2.4 Competitive Repetition-suppression (CoRe) Learning

CoRe is a soft-competitive learning model that has been inspired by the repetition suppression memory mechanism described in 2.2.3. In this model, only a subset of the most active units is allowed to learn in proportion to their activation strength, while the least active units are penalized by

reducing their response to the pattern that has produced the low firing strength. In this section we describe a general formulation of CoRe as an unsupervised neural network model, while in Section 2.5 we shift our focus on the application of CoRe learning to clustering and cluster number determination.

As in every competitive learning model, prototypes are fundamental in the definition of the CoRe model: given a unit $u_i \in U$, the prototype c_i determines the neuron preferred stimulus, that is the pattern that produces the highest neural activation. In addition, units are characterized by an activation function $\varphi_i(x_k|\lambda_i)$, defined in terms of the unit parameters λ_i, that determines the firing strength of the neuron in response to the presentation of an input $x_k \in \chi$. The exact form of the activation function is left unspecified at this stage, but we assume it to have a bounded maximum: typically, the activation function φ_i serves as an estimate of the similarity of input pattern x_k with the prototype c_i.

The CoRe competition is engaged between two sets of units: at each step the most active units are selected to form the *winners pool*, while the remainder is inserted into the *losers pool*. More formally, we define the winners pool for the input x_k as the set of units u_i that fires more than θ_{win}, that is

$$win_k = \{i \mid \varphi_i(x_k|\lambda_i) \geq \theta_{win},\ u_i \in U\} \cup \{i \mid i = \arg\max_{j \in U} \varphi_j(x_k|\lambda_j)\}, \tag{2.20}$$

where the second term of the union ensures that the winner set isn't empty if no unit fires more than θ_{win}. Conversely, the losers pool corresponding to x_k is

$$lose_k = \{i \mid \varphi_i(x_k|\lambda_i) < \theta_{win},\ u_i \in U\} \setminus \{i \mid i = \arg\max_{j \in U} \varphi_j(x_k, |\lambda_j)\}, \tag{2.21}$$

where the second term of the set difference discounts the most active unit from the loser pool in order to ensure that $\{win_k, lose_k\}$ is a partition of the units set U. Hence, for each incoming stimulus x_k, neurons are partitioned into the two sets win_k and $lose_k$ depending on their activation level. The *contrastive* soft-competition mechanism of CoRe allows all the units in the winners pool to be *positively* adapted by strengthening their

response to x_k; at the same time, neurons in the losers pool are subject to a *negative* reinforcement that makes them less likely to be activated by the next presentation of the stimulus x_k. Such partitioning is determined by the firing threshold θ_{win}, that can be interpreted as the minimal firing strength that a neuron has to achieve to be considered selective for a pattern x_k. In Section 2.6.1 we will discuss how the choice of this parameter influences the behavior of the algorithm. Figure 7 shows a graphical interpretation of a general CoRe neuron: the prototype (depicted as a black circle) identifies the preferred stimulus and each neuron is characterized by an area of its receptive field where only attractive learning can happen (i.e. the gray areas enclosing the prototypes in Fig. 7). The shape and size of this area is determined by the choice of the winners threshold as well as by the properties of the activation function. Neurons engage a competition for each stimuli falling inside their receptive fields but outside the positive learning area: the lightly shaded portion of the space in Fig. 7 indicates the area where repetition suppression will surely be applied to the loser neurons.

Modeling the stimulus repetition is a key issue for implementing a learning scheme based on repetition suppression. In our model we define a parameter, named *stimulus predominance*, that represents an approximate measure of the pattern frequency. The stimulus predominance at the time t is defined as

$$\nu_i^t = \frac{1}{|\chi_t|} \sum_{x_k \in \chi_t} \eta_i(x_k), \tag{2.22}$$

where χ_t is the set of the input patterns presented to the network up to time t and $\eta_i(x_k)$ is the relative activation of the i-th unit upon the presentation of input x_k, that is

$$\eta_i(x_k) = \frac{\varphi_i(x_k|\lambda_i)}{z_U^k}, \tag{2.23}$$

where z_U^k acts as a normalization factor. For instance, the term z_U^k can be chosen as the output of the maximally active unit, from the set U, on the pattern x_k, i.e. $z_U(x_k) = max_{u_j \in U}\{\varphi_j(x_k|\lambda_j)\}$. Alternatively, z_U^k can be

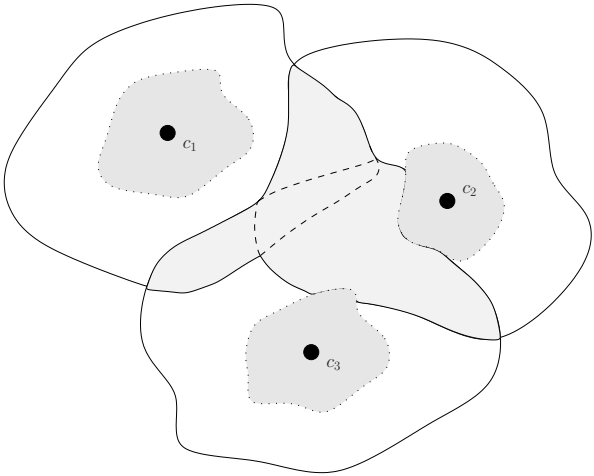


Figure 7: Receptive fields for a generic CoRe neuron: the solid lines determine the boundaries of the receptive fields, while the dark-gray areas identify the regions of the space where each neuron will surely be a winner (depending on the value of θ_{win}). The light shaded area shows the region where neurons compete for activation and losers are suppressed.

calculated as a soft max (see (BS06)) or as the sum of the activations of the units $u_j \in U$.

The general objective of the definition in (2.22), is to measure the frequency of the patterns that are preferred by the unit with prototype c_i, scaled by the relative activation strength with respect to the contribution of the maximally similar prototype z_U^k. The stimulus predominance is used to regulate the amount of penalization that is applied to the losers, defined as

$$RS_k^t = \frac{1}{M|win_k|} \sum_{i \in win_k} \nu_i^t \varphi_i(x_k|\lambda_i), \tag{2.24}$$

that is the repetition suppression for the pattern x_k calculated at time t, where M is the maximum of the activation function φ_i.

The expression in (2.24) is used to calculate the penalization for the losers neuron. For instance, we can use it to define a pseudo-target activation for units in the losers pool as $\hat{\varphi}_i^t(x_k) = \varphi_i(x_k|\lambda_i)(1 - RS_k^t)$. This reference signal forces the losers to reduce their activation proportionally to the amount of repetition suppression they receive. The error of the i-th loser can thus be written as

$$\underline{E}_{i,k}^t = \frac{1}{2}(\hat{\varphi}_i^t(x_k) - \varphi_i(x_k|\lambda_i))^2 = \frac{1}{2}(-\varphi_i(x_k|\lambda_i) RS_k^t)^2. \tag{2.25}$$

Conversely, in order to strengthen the activation of the winner units, we set the target activation for the neurons $u_i \in win_k$ to M. The error, in this case, can be written as

$$\overline{E}_{i,k}^t = (M - \varphi_i(x_k|\lambda_i)). \tag{2.26}$$

The unit parameters λ_i can be adapted by an optimization procedure that minimizes the error functions defined in (2.25) and (2.26). Notice that although CoRe learning may resort to supervised learning for training the λ_i, it remains an unsupervised algorithm since all the reference signals it uses are self-generated on the basis of the input pattern distribution over space and time.

The ideal outcome of CoRe learning would be to generate pools of highly selective neurons, hence, it is important to define a metric for iden-

tifying the most significant units which have been produced by the learning process. We define the *relevance factor* for the unit u_i as

$$\hat{\nu}_i^t = \frac{1}{\nu_i^t |\chi^t|} \sum_{x_k \in win_{u_i}^t} \frac{\varphi_i(x_k|\lambda_i)}{z_{win_k}^k}, \tag{2.27}$$

where $z_{win_k}^k$ follows the definition given above and $win_{u_i}^t$ is the set of patterns $x_k \in \chi_t$ for which unit u_i was in the winners pool, i.e.

$$win_{u_i}^t = \{x_k \mid i \in win_k,\ x_k \in \chi_t\}. \tag{2.28}$$

In other words, the relevance factor defines a soft-measure of the frequency with which u_i was the most active unit in the winners pool. This measure can be used to determine which units are less significant for the representation of the input patterns and can be used, for instance, for deciding whether a neuron has to be retained or discarded by the network. How exactly this is done depends on the network implementation: for instance, in this thesis, we focus on a pruning approach where neurons with a low relevance are pruned permanently from the network. Here the low relevance is determined by a threshold θ_{prune} that has to be provided by the user: more details about this strategy, and about the effects of the choice of the metaparameter θ_{prune}, will be given in the next section. We would like to point out that pruning is not the only strategy available for treating irrelevant neurons: for instance, it can be implemented a neuron commitment-uncommitment mechanism that behaves specularly with respect to ART networks (see Section 2.3). In other words, CoRe can be started with all the neurons committed so that, as learning proceeds, irrelevant units are *silenced* by uncommitting them as soon as their relevance falls below a given threshold $\theta_{silence}$. Uncommitted neurons can later be re-activated if a surprising stimulus is encountered, that is to say, if none of the committed neurons fires more than a given threshold θ_{min}. By means of this strategy, we can realize incremental learning within the CoRe framework.

2.5 Competitive Repetition-suppression (CoRe) Clustering

2.5.1 The Algorithm

The general Core model introduced in the previous section can be readily instantiated in a clustering context with application to the cluster number determination problem. Each CoRe unit is interpreted as a cluster detector, where the prototype c_i defines the cluster representative and the activation function $\varphi_i(x_k)$ determines whether the pattern x_k belongs to the i-th cluster. Here, we take a soft-clustering approach, where each pattern can be assigned to multiple clusters with different membership weights. In particular, we chose the activation function to be a bounded measure of similarity between the pattern x_k and the prototypes c_i. A suitable choice for φ_i is the gaussian function centered in c_i with spread σ_i, i.e.

$$\varphi_i(x_k|\lambda_i) = e^{-\frac{\|x_k - c_i\|^2}{2\sigma_i^2}} \tag{2.29}$$

where λ_i includes all the center components c_i and the spread σ_i (in the remainder of the section λ_i will be omitted to ease the notation of φ_i). Hence, the winners set contains those units that are selectively tuned on the current pattern x_k while the losers set contains those neurons that have a dissimilar prototype c_i or that have insufficiently sharp responses. Notice that in order to simplify the index notation we assumed in (2.29) (and in the remainder of the chapter) to work with scalar σ_i's (i.e. purely spherical Gaussians). However, all the results described hereafter hold for the general case of symmetric covariance matrices Σ_i and elliptic Gaussians, i.e. $\varphi_i(x_k|\lambda_i) = \exp\{-(x_k - c_i)^T \Sigma^{-1} (x_k - c_i)\}$.

Following the indications in Section 2.4, we obtain the incremental learning rules for prototype adjustment and spread tuning by applying *gradient descent* to the error measure in (2.25) and (2.26). The parameter increments for the units $u_i \in lose_k$ can be obtained by differentiating (2.25) with respect to the parameters c_i and σ_i. Therefore the prototype

update at time t can be calculated as

$$\triangle \underline{c}_{i,k}^{t} = \frac{\partial \underline{E}_{i,k}^{t}}{\partial c_i} = \varphi_i(x_k) RS_k^t \frac{\partial(\varphi_i(x_k) RS_k^t)}{\partial c_i} \tag{2.30}$$

where the differentiation on the right side can be expanded by chain rule as

$$\begin{aligned} \frac{\partial(\varphi_i(x_k) RS_k^t)}{\partial c_i} &= \frac{\partial(\varphi_i(x_k) RS_k^t)}{\partial \varphi_i(x_k)} \frac{\partial(\varphi_i(x_k))}{\partial c_i} \\ &= \left(RS_k^t + \varphi_i(x_k) \frac{\partial(RS_k^t)}{\partial \varphi_i(x_k)} \right) \cdot \frac{\partial(\varphi_i(x_k))}{\partial c_i} \\ &= RS_k^t \varphi_i(x_k) \frac{(x_k - c_i)}{\sigma_i^2} \end{aligned} \tag{2.31}$$

in which we have used $\frac{\partial(RS_k^t)}{\partial \varphi_i(x_k)} = 0$ if $u_i \in lose_k$ at time t (follows from the definition of RS in (2.24)). Substituting the results of (2.31) in (2.30) we obtain

$$\triangle \underline{c}_{i,k}^{t} = \left(\frac{\varphi_i(x_k) RS_k^t}{\sigma_i} \right)^2 (x_k - c_i). \tag{2.32}$$

Similarly, the spread update at time t can be calculated as

$$\begin{aligned} \triangle \underline{\sigma}_{i,k}^{t} &= \frac{\partial \underline{E}_{i,k}^{t}}{\partial \sigma_i} = \varphi_i(x_k) RS_k^t \frac{\partial(\varphi_i(x_k) RS_k^t)}{\partial \sigma_i} \\ &= (\varphi_i(x_k) RS_k^t)^2 \frac{\|x_k - c_i\|^2}{\sigma_i^3} \end{aligned} \tag{2.33}$$

while the parameter increments for the units $u_i \in win_k$ can be written as follows

$$\triangle \overline{c}_{i,k}^{t} = \frac{\partial \overline{E}_{i,k}^{t}}{\partial c_i} = -\varphi_i(x_k) \frac{(x_k - c_i)}{\sigma_i^2} \tag{2.34}$$

$$\triangle \overline{\sigma}_{i,k}^{t} = \frac{\partial \overline{E}_{i,k}^{t}}{\partial \sigma_i} = -\varphi_i(x_k) \frac{\|x_k - c_i\|^2}{\sigma_i^3}. \tag{2.35}$$

The update rule for the unit parameters $\lambda_i^t = (c_i^t, \sigma_i^t)$ is

$$\lambda_i^t = \begin{cases} \lambda_i^{t-1} - \alpha_\lambda^{lose} \triangle \underline{\lambda}_{i,k}^{t} & \text{for} \quad i \in lose_k \\ \lambda_i^{t-1} - \alpha_\lambda^{win} \triangle \overline{\lambda}_{i,k}^{t} & \text{for} \quad i \in win_k \end{cases} \tag{2.36}$$

where α_λ^{win} and α_λ^{lose} are the learning and de-learning rate, respectively, with $\alpha_\lambda^{win} \gg \alpha_\lambda^{lose}$.

Analyzing equations (2.32) to (2.36) gives an interesting insight on the algorithm behavior. As one would expect, winner units have their prototype moved towards the current pattern x_k (see (2.34)), but at the same time their selectivity is enhanced by reducing the spread of their activation function (in (2.35)). Conversely, loser units are moved further from the current prototype (see (2.32)), while the spread of their activation function is enlarged (in (2.33)). On the one hand, this compensates for the displacement with respect to the unit preferred stimuli while, on the other hand, it reduces the unit selectivity, thus penalizing it for not having sharp responses and making it more prone to be discarded by a relevance/selectivity based pruning. This spread adjustment process has also a self-regulating effect on frequent winner neurons. For instance, if a unit u_i wins for many samples, then its spread σ_i^2 will reduce up to a point where u_i stops prevailing for certain patterns x_k, that will hence exit from its winner set $win_{u_i}^t$. On the other hand, weakly winning units maintain wider spreads, hence they are able to compete again on those x_k that have exited $win_{u_i}^t$. Therefore, CoRe clustering provides an adaptive conscience mechanism that prevents the low-utilized center problem (AKCM90).

In order to analyze in detail the behavior of the proposed CoRe clustering model, a procedural description of its learning steps is given in Algorithm 1. Notice that the algorithm ensures that, for each pattern x_k, there exist at least one winner unit that, in the worst-case scenario, is the most active neuron (see the win_k^t empty condition in Algorithm 1). Such a winner search policy can be refined by envisaging more complex winner recruitment policies, such as those described at the end of Section 2.4. In particular, a constraint can be added to the minimum activation that a winner neuron has to produce. For instance, if the firing strength of the maximally active unit does not exceed a predefined minimum threshold θ_{min}, we can consider reallocating the unit with the smallest *relevance factor*, whenever this value does not exceed a threshold θ_{rel}. If such minimum relevance condition is not satisfied, then there are no uncommitted

Algorithm 1 CoRe Clustering

Given: the dataset $\chi = \{x_1, \ldots, x_K\}$, the winners threshold θ_{win}, the pruning threshold θ_{prune}, the spread initialization constant α and the initial cluster number N

Set $t = 0$, $\theta_{decay} = 1 - \frac{1}{K}$, $\nu_i^t = 1$ and $\hat{\nu}_i^t = 1$;
Randomly initialize the N prototypes c_i^t;
Initialize the Gaussian spreads $(\sigma_i^0)^2$ proportionally to the data covariance (see Eq. 2.38);
repeat
  Generate a random presentation order for the patterns in χ;
  for all $x_k \in \chi$ **do**
    $t = t + 1$
    for $i = 1$ to N **do**
      Calculate unit activation $\varphi_i(x_k)^t$;
      if $\varphi_i^t(x_k) \geq \theta_{win}$ **then** Add i to win_k and x_k to $win_{u_i}^t$;
      else Add i to $lose_k$;
      end if
    end for
    if win_k is empty **then**
      Set $win_k = \left\{i | i = \arg\max_j \{\varphi_j^t(x_k)\}\right\}$ and remove i from $lose_k$;
    end if
    for $i = 1$ to N **do**
      Update ν_i^t using (2.22);
      if $i \in win_k$ **then** Update $\hat{\nu}_i^t$ using (2.27);
      else Apply the relevance factor decay $\hat{\nu}_i^t = \theta_{decay} \cdot \hat{\nu}_i^{t-1}$;
      end if
    end for
    Compute the repetition suppression RS_k^t using (2.24);
  end for
  for $i = 1$ to N **do**
    if $i \in win_k$ **then** Update c_i^t and σ_i^t using the increments in (2.34) and (2.35);
    else Update c_i^t and σ_i^t using the increments in (2.32) and (2.33);
    end if
    if $\hat{\nu}_i^t \leq \theta_{prune}$ **then**
      Prune the unit u_i;
      $N = N - 1$;
    end if
  end for
until convergence

units available and all neurons are considered to be too relevant for being pruned: hence, a new unit can be allocated and set to be the winner.

In this chapter we limit our analysis to the basic winner search policy described in Algorithm 1. Notice that the unit pruning decision in Algorithm 1 is taken based on the value of the relevance factor. This parameter is adjusted at every pattern presentation: the neurons in the winners pool are updated based on the expression in (2.27), while an *exponential decay* weight is applied to the units in the losers pool. As a result, those units that are frequent winners and that are often the best matching unit are preserved, while those not satisfying this condition are bound to be pruned. The choice of the pruning threshold θ_{prune} and of the exponential decay weight has to ensure that the pruning decision is taken based on a sufficient data support: Section 2.6.1 discusses the details of the CoRe exponential decay process and the influence of the pruning threshold and the decay weight on CoRe learning behavior.

2.5.2 Related Work

Before delving into the analysis of CoRe clustering performance we briefly review common traits and key differences between CoRe and the related work in the literature. As noted in the previous sections, the CoRe algorithm works essentially by evolving a small set of highly selective units out of an initially large neural population: therefore, the algorithm needs to be initialized with a *sufficiently* large number of units. In this sense, CoRe resembles progressive clustering algorithms such as RCA (FK99), where an initial overestimation of the cluster number leads to the final data partition by iteratively merging irrelevant units into similar clusters. Moreover, CoRe relates to another general model, that is the *similarity-based clustering* (SCM) (YW04) described in Section 2.3. SCM performs clustering on the basis of a maximization procedure on a total objective function defined in terms of the similarity of the patterns to the unit prototypes with respect to a given measure (in particular (YW04) uses a gaussian function). Like CoRe and RCA, SCM starts with an overestimation of the prototype number, merging them only at the end of the clustering

procedure by means of an Agglomerative Hierarchical Clustering (AHC) procedure. Such an approach is computationally expensive since cluster pruning is deferred to the end of the prototype positioning phase as well as because the AHC method itself requires a consistent computational load, being quadratic with respect to the dataset size. Conversely CoRe, as it will be formally discussed in Section 2.6.3, requires a consistently lower computational effort given the fact that cluster selection is performed incrementally during pattern presentation and that the repetitions suppression penalization term can be computed in linear time.

CoRe clustering is, indeed, closely related to the family of the unsupervised competitive learning models. As noted previously, Self Organizing Maps (SOM) (Koh82) introduced the idea of a competition with multiple winners, allowing a subset of the network units to learn proportionally to their spatial contiguity with the BMU, producing maps of topographically ordered neurons. Likewise SOM, CoRe allows a subset of the units to win the competition, although it does not produce topographically ordered neural maps: on the other hand, CoRe allows loser neurons to adapt their prototype with a negative reinforcement. In other words, each CoRe unit behaves like a class detector, assigning positive class information to those patterns falling inside the excitatory region of its receptive field and negative class labels to those patterns in the inhibitory region. This learning dynamics relates closely to the behavior of the Rival Penalized Competitive Learning (RPCL) algorithm. However, CoRe generalizes the RPCL model by allowing winner and rival competitors to refer to set of units (instead of single neurons), implementing a soft-competitive rival penalization scheme. Moreover, CoRe's repetition suppression mechanism, besides having a strong biological interpretation, addresses a key open issue of the RPCL model, that is how to modulate the strength of penalization (Che05). In particular, the RS mechanism offers a means for adaptively controlling rival penalization, so as to produce a sufficient level of prototype repulsion without generating instability. In the experimental evaluation part, we will show how this issue degrades strongly the performance of the RPCL algorithm whenever the number of initial units is not a close guess with respect to the true cluster

number.

CoRe formulation is general enough to be applied to neurons with a broad selection of activation functions (e.g. Gaussian, sigmoid, etc.), in contrast with the RPCL model that is restricted to hard competitive Euclidean units. In our intention, CoRe can serve as an unsupervised learning algorithm in complex hierarchical neural models comprising multiple hidden layers, such as those developed by the machine vision community (e.g. Neocognitron (Fuk03), the HMAX model (SWB$^+$07) or convolutional networks (LHB04)). In particular, CoRe can be used to develop compact neural representations of the information flowing between the most internal layers of such hierarchical models. Finally, the CoRe model defines a measure of neuron significance, i.e. the relevance factor, that can be used to selectively silence (or prune) irrelevant neurons. As it has been noticed earlier, this particular mechanism can be considered as the counterpart of the neuron commitment mechanism of ART networks (CG88), where the relevance threshold θ_{prune} plays the role of a vigilance parameter.

The recent evolution of RPCL into a divide-and-conquer framework for multi-agents systems (see Section 2.3) has extended it to a more general competitive learning strategies, comprising multiple winners and rivals as well as a wider choice of the agents error functions. Although this model represents an interesting development of the RPCL framework, it retains two key open issues of the earlier model, that are the penalization modulation and the robustness to noise. The former problem has already been described above, whereas the latter relates to the influence that noise and outliers have on the quality of the clustering result and will be studied in detail in Chapter 3. CoRe learning addresses these two issues by providing a means for adaptively controlling the penalization strength, that is the repetition suppression mechanism, and, as it will be shown in Section 3.2.1, by means of a robust cluster identification strategy.

In the remainder of the chapter, we investigate the performance of the CoRe model by comparing it with several state-of-the-art clustering algorithms. In particular, we analyze the behavior of CoRe learning and RPCL with respect to the open issues instantiated above. Such compara-

tive analysis is carried out on the original RPCL formulation, since these issues are common to both the original and the generalized model, which, however, still misses a reference experimental and simulative analysis (Xu07).

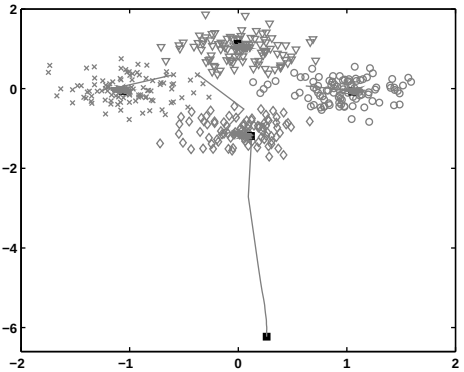
2.6 Experimental Evaluation

In this section we present the results of several experiments assessing CoRe performance on various popular clustering tasks, comparing its performance with that of several state-of-the-art clustering algorithms. In particular, we focus on a comparative analysis between CoRe clustering and RPCL. Before describing the outcome of the experimental evaluation, we analyze how the choice of the CoRe parameters influences its behavior. Finally, we discuss the computational complexity of the CoRe algorithm with respect to the models used in the experimental evaluation.

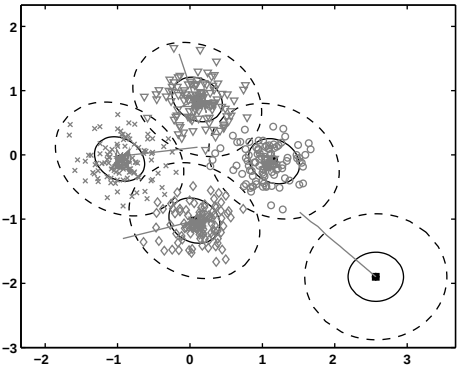
2.6.1 Fitting the model: Meta-Parameters Selection

The formal description of CoRe model given in Algorithm 1 highlights the parameters that might influence CoRe performance, that are: the winners threshold θ_{win}, the spread initialization σ_i^0, the decay weight θ_{decay} and the pruning threshold θ_{prune}. To describe their effects on CoRe learning dynamics we consider an exemplar four classes dataset consisting of 400 datapoints generated by four gaussian sources with mean (1,0),(-1,0),(0,1),(0,-1) and variance 0.1.

Figure 8.a and 8.b show a clustering result obtained with RPCL and CoRe starting with an initial population of 5 seed points: both models are able to identify the correct number of clusters and the corresponding centroids with 98% of correctly clustered data points. The solid gray lines in Fig. 8 trace the prototype trajectories produced during the centroid identification phase. Figure. 8.b visually explains the behavior of CoRe neurons. The solid ellipses identify the *support* of the neuron, that is the hypersphere enclosing the samples that produce positive learning or, in other words, the portion of the input space where the neuron will surely



(a) RPCL Clusters



(b) CoRe Clusters

Figure 8: Clustering on four gaussian distributions with mean (1,0),(-1,0),(0,1),(0,-1) and variance 0.1. The identified clusters are marked as $\times, \circ, \diamond, \triangledown$) and the corresponding centroids are identified by ■. The solid gray lines in (a) and (b) depicts the prototype trajectories during training. The solid ellipsoids in (b) identify the support of the CoRe neurons, while the dashed ones identify the corresponding receptive fields. Notice that CoRe's results have been obtained with pruning disabled to allow a straightforward comparison with RPCL.

be a winner. The dashed ellipses, on the other hand, identify the significative area of the receptive fields (RF). In particular, the portion of the input space where dashed RFs overlap determines the stimuli that will have a suppressive effect on the non winning neurons. In other words, the inputs producing repetition suppression will tend to deflect the loser neurons; such a displacement will be contrasted by the patterns belonging to the unit support, that will produce an *anchoring* effect on the respective prototypes. As a consequence, if a neuron has a sufficient support, it won't be deflected by the repetition suppression. The learning dynamics of CoRe neurons has a nice biological interpretation, since it resembles a naive on-center/off-surround strategy. For instance, neurons responds positively to stimuli falling inside the center of their receptive fields, while they might respond negatively to patterns falling in the surround area. The naive part is related to the fact that a unit might respond positively to stimuli in the surround if no other CoRe neuron is sufficiently activated.

The winners threshold θ_{win} determines the relative size of the neuron support with respect to the receptive field size. Using the definition of winners pool in (2.20) and the Gaussian activation in (2.29), we can determine the hypersphere enclosing the portion of the space where the i-th neuron will surely be a winner as

$$Supp_i = \{x \mid \|x - c_i\|^2 \leq -2\log(\theta_{win})\sigma_i^2\}, \tag{2.37}$$

that corresponds to the solid ellipses in Fig. 8.b. Clearly, if we take

$$\lim_{\theta_{win} \to 0} Supp_i$$

we obtain a support that covers the full extension of the neuron's receptive field, resulting in a pure soft-competitive scheme (YZG92). For instance, given an input pattern x, every unit with a non-zero activation will be a winner, learning to approach the input vector proportionally to its activation strength. At the same time, the gaussian spread will tend to reduce as a consequence of learning (see Eq. (2.35)) in accordance with the soft-competitive spread annealing process proposed by Gersho et al. in (YZG92). The winners threshold θ_{win} regulates the tradeoff between

positive learning, i.e. related to winners, and negative learning, that is related to loser units. If θ_{win} approaches to zero, then there will be only a slight amount of penalization applied to the neurons that will, then, be prone to overfit the data. On the other hand, setting an high threshold might produce an excessive amount of penalization, thus preventing the correct identification of the clusters. Figure 9 shows CoRe performance on the four Gaussians dataset for different values of θ_{win} and for a different number of seed points. The solid line in Fig. 9 describes CoRe performance when the number of units is a close approximation of the true cluster number, i.e. 5 seed points; the dashed line shows the performance when using a neat over-approximation of the cluster number, i.e. 20 units. The best solution is clearly obtained for $\theta_{win} = 0.9$: under this value the performance of the 20 units case starts to deteriorate quickly, since the amount of penalization produced is too small to deflect all the units in excess, thus producing overfitting of the data. On the other hand, the 5-units network start to deteriorate performance for $\theta_{win} \leq 0.7$: the peak around 0.5 is generated by the complete superposition of the receptive fields of two neurons, that produces a good clustering result (notice that in a vector quantization problem only the *first* maximally responding unit is chosen to represent each pattern) but an incorrect prototype allocation. Similarly, the performance drop for the 20-units network reduces for $\theta_{win} \leq 0.4$, since the algorithm start to behave like a pure soft-clustering model: the little penalization effect allows neurons to superimpose almost perfectly the respective receptive fields. Again, in this simple example this may result in a good clustering performance but with a wrong prototype allocation.

The choice of the initial variances σ_i^0 is another important factor influencing the performance of CoRe learning. In particular, the spreads have to be selected wide enough to allow competition between the CoRe units, but not so large to produce too much de-learning, possibly deflecting the units from the data and delaying the convergence of the algorithm. We initialized the spread to be proportional to the sample variance σ_χ^2, that is

$$\sigma_\chi^2 = \frac{\sum_{k=1}^{|\chi|} \|x_k - \overline{x}\|^2}{|\chi|} \quad \text{where} \quad \overline{x} = \frac{\sum_{k=1}^{|\chi|} x_k}{|\chi|}. \tag{2.38}$$

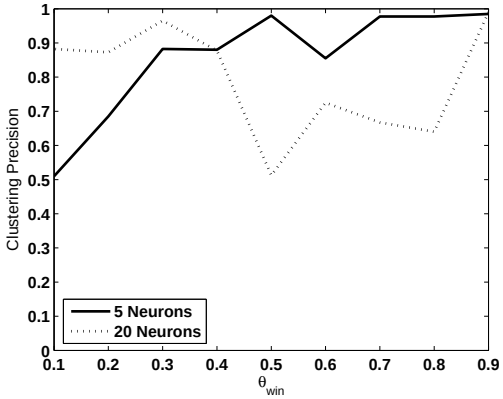


Figure 9: Performance of the CoRe clustering algorithm as a function of the winner threshold θ_{win}.

Using the expression in (2.38), we can write the initial spread as $\sigma_i^2 = \alpha \cdot \sigma_\chi^2$, where α is a proportionality constant. Figure 10 illustrates the performance of a 5 neuron CoRe network on the four Gaussians dataset for different values of α and diverse prototype learning rates. The solid line is related to the CoRe network using a learning rate of 0.005 for winner and loser prototypes: the best results are obtained for $\alpha \in [0.15, 0.4]$, while larger α values produce an excessive unit repulsion, preventing CoRe from identifying correctly the cluster number. The dashed line in Fig. 10 shows the results obtained for a lower prototype de-learning rate (i.e. 0.001): the clustering error for large α values is smaller due to the reduced repulsive strength, but, in general, the algorithm is not capable of correctly identifying the true cluster number, since the unit in excess cannot be driven away from the data.

All the results presented so-far have been produced by a CoRe algorithm based only on prototype repulsion, without using any pruning strategy. Here, we study the behavior of the CoRe algorithm subject to the relevance pruning described in Section 2.5 and Algorithm 1. First, the expression in (2.27) describes the relevance of a neuron in terms of its selectivity, measuring how often a neuron is the best matching unit

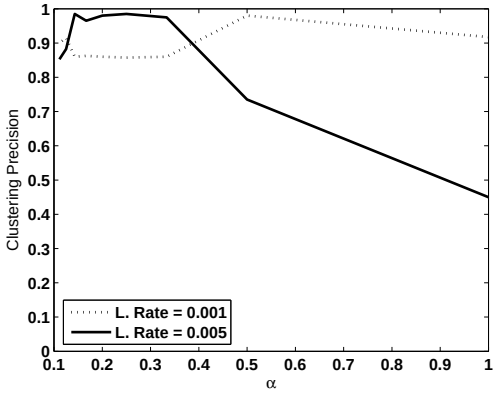


Figure 10: Performance of the CoRe clustering algorithm as a function of the proportionality constant α regulating the initial spread size.

for the prototypes in its winners set. This measures tends to one for the highly selective neurons that fire only for patterns for which they have been tuned to respond. On the other hand, the relevance factor drift to zero for units that fire spuriously for many patterns that are already represented by other neurons. However, this process of relevance zeroing may take long or can be stopped if, for instance, a unit has few patterns left into its receptive field. Hence, for each pattern presentation we apply a constant exponential decay $\hat{\nu}_i^t = \theta_{decay} \cdot \hat{\nu}_i^{t-1}$ to the loser neurons relevance and we set a pruning threshold θ_{prune} that determines which units have to be removed from the network (see Algorithm 1). The rationale behind this choice is to save computational load and to speed up convergence by removing early those units that are considered irrelevant. The choice of the decay weight θ_{decay} and of the pruning threshold θ_{prune} is central in determining the performance of the pruning process. Here, we choose a decay weight $\theta_{decay} = 1 - (\frac{1}{K})$, that depends on the size of the dataset $K = |\chi|$. The dotted line in Fig. 11 shows the behavior of the relevance factor of a neuron under the effect of the exponential decay alone: the vertical line determines the time instant when the relevance factor of the decay-only curve reaches the level $\theta_{prune} = 0.005$. In this ex-

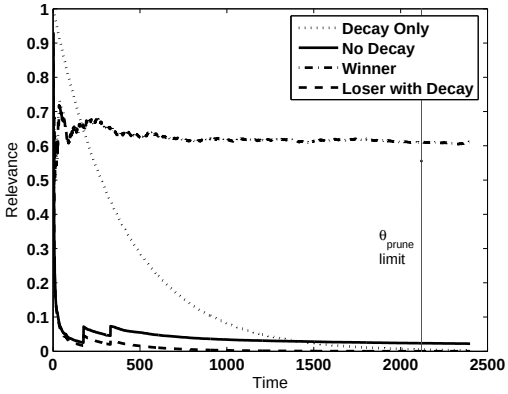


Figure 11: Behavior of the relevance factor $\hat{\nu}$ under different conditions.

ample the dataset size m is equal to 400, hence the decay weight reaches the pruning limit after more than 5 learning epochs. The solid line, on the other hand, shows the relevance of a loser neuron on the 4 Gaussian example: the relevance reduction is steeper in early phases of learning but stops before the neuron can be pruned. Combining such *natural* relevance reduction with the exponential decay produces the dashed curve in Fig. 11: the relevance reduction effect continues after the initial epoch so that the pruning decision can be taken early. Finally, the dashed-dotted curve in Fig.11 describes the relevance of a winner unit: after an initial tuning phase, the neuron relevance stabilizes to a level that prevents it from being pruned. The behavior of the relevance curves in Fig. 11 shows that a neuron that fires un-selectively for patterns that are assigned to other units is most likely to be pruned with respect to a neuron that never wins and that is then subject to the exponential decay alone. This strategy is consistent with the biological foundations of the repetition suppression mechanism, where the most un-selective neurons experience more suppression. Moreover, this behavior determines a fast pruning of those units that may interfere with the selective neurons, e.g. by deflecting them from their cluster centroids by means of the RS mechanism.

The determination of the convergence criteria is another key-issue influencing the clustering outcome. Typical convergence criteria establish a measure of clustering stability, for instance by setting a threshold on the prototype change between epochs, that is $\max_i \|c_i^{t+1} - c_i^t\| \leq \epsilon$. Alternatively, it may be required that the cluster assignment of the patterns remains stable for a number E of epochs. To determine convergence we take into consideration, together with the clustering stability criterion, a measure of the residual competition that is ongoing between the units. In particular, we define a measure of the amount of RS that is generated during an epoch e, that is the Mean Applied Repetition Suppression (MARS)

$$\frac{\sum_{k=1}^{|\chi|} \sum_{i \in lose_k} RS_k^{(e|\chi|+k)} \cdot \varphi_i(x_k)^2}{|\chi|}. \tag{2.39}$$

The MARS formula in (2.39) essentially measures the mean repetition suppression that has been applied to the loser neurons during a given learning epoch. MARS therefore assesses the strength of the competition that is ongoing between the units: when its value becomes negligible, then little repulsion is applied and neurons are mostly acting based on a purely attractive dynamics. Hence, the cluster number identification phase can be considered completed and convergence can be determined by a clustering stability criterion. In the experiments presented hereafter, we determined convergence for MARS values under 0.05 and cluster assignment stability for 10 epochs. As regards the simulative setup, following the results of the analysis discussed above, we initialized $\theta_{prune} = 0.005$ and $\theta_{win} = 0.9$, while the initial variance was determined based on (2.38) with proportionality constant $\alpha = 0.2$. As regards the cluster initialization, all the simulations were run repeatedly with randomly initialized prototypes, without experiencing consistent fluctuations in the clustering performance. Similarly, the datapoints' presentation order was permuted randomly at each training epoch, in order to untie performance from specific clustering pattern sequences. The RPCL algorithm has been implemented based on the model in (XKO93) and tested to reproduce the results in the literature: RPCL parameters have been optimized to maximize its performance in the single experimental trials.

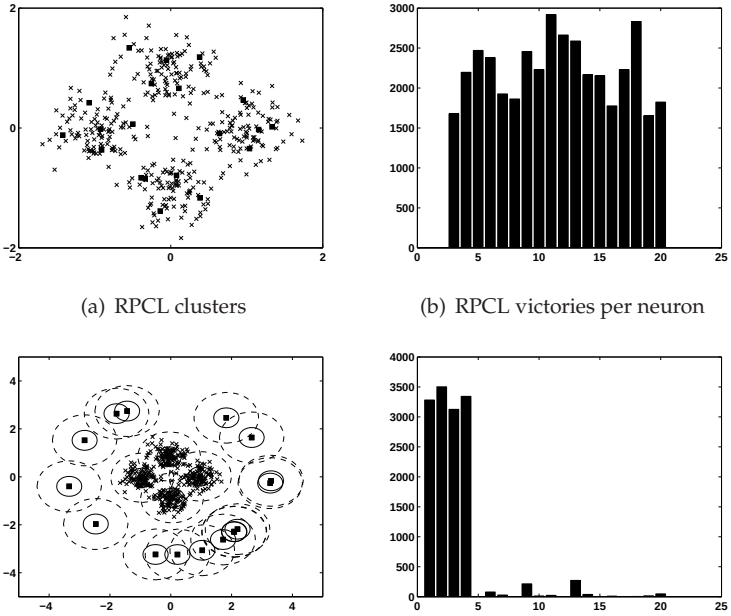
2.6.2 Results and Discussion

Example 1. The first example is based on the four Gaussian dataset described in the previous section. Here, CoRe clustering is compared to RPCL when the initial guess about the number of seed units is not accurate enough. For instance, at this stage we run RPCL and CoRe with a very rough approximation of the cluster number (i.e. 20 units): the two types of graphs in 12 depict the prototype displacement with respect to the datapoints and the number of victories per units, where the latter graph is intended to show the sharpness of the neural representation.

The plots for RPCL (i.e. Fig. 12.a and 12.b) show that the rival penalization mechanism is not sufficient to deflect the excess units from the data: as a result all the seed points are allocated to cover a portion of the input vectors. This behavior is clearer from the bar graph in Fig. 12.b, where we notice that all the neurons tend to converge to a common mean firing frequency. In contrast, the CoRe repetition suppression mechanism has a strong de-learning effect on those neurons that do not show enough selectivity, moving them outside the area where the input data lies (see Fig. 12.c). Notice that at this stage, we have disabled CoRe pruning to allow a fair comparison between the two algorithms.

The bar graphs in Fig. 12.d show a different behavior with respect to RPCL: as CoRe learning proceeds, the overall neural activation decreases and few neurons become highly frequent winners. As a result, CoRe can correctly detect the 4 clusters, deflecting the units in excess from the data and obtaining a result that is comparable with that obtained with pruning enabled (see Fig. 13.a).

The RS mechanism has a self-regulating effect on the magnitude of de-learning: hence the prototype repulsion learning rate can be chosen equal to the attractive learning rate, obtaining a consistent deflection of the excess prototypes that does not impair the convergence of the winner prototypes to the centroid of the clusters. In fact, the RS mechanism slowly decreases the amount of penalization as the neural representation gets sharper, thus automatically regulating the rate of de-learning. In contrast, RPCL performance is highly sensitive to the selection of the de-learning



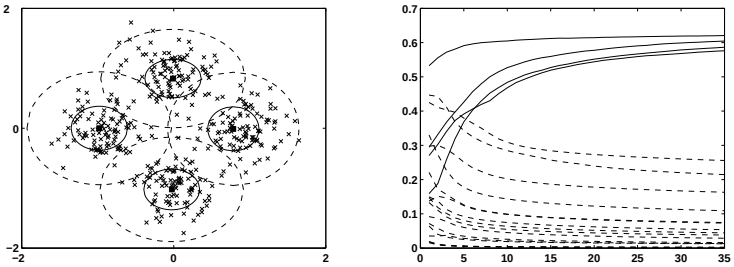
(a) RPCL clusters

(b) RPCL victories per neuron

(c) CoRe clusters

(d) CoRe victories per neuron

Figure 12: RPCL and CoRe performance on the four gaussian dataset: CoRe results have been obtained with pruning disabled to show a fair comparison in terms of neural selectivity. (a) RPCL cannot drive away the excess units when the initial number of neurons overestimates too much the true data distribution: as a consequence, the number of victories gets distributed almost evenly among the neurons (b). (c) CoRe with disabled pruning deflects all the excessive neurons from the data; (b) as CoRe learning proceeds it produces a sharper neural representation with reducing number of active neurons, in compliance with the repetition suppression model.

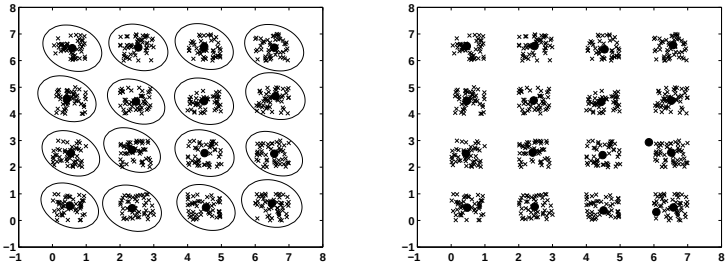


(a) CoRe with pruning (b) Relevance factor

Figure 13: CoRe performance on the four gaussian dataset with pruning enabled: all the excessive neurons are pruned (a). The curves in (b) show the behavior of CoRe's relevance factor as learning proceeds.

rate, since, in general, it needs to be re-selected appropriately not only for different clustering problems, but also for different initial positions of the seed points (Che05). In this particular example, values of the de-learning rate superior to 0.005 prevent the convergence of the RPCL algorithm, while smaller values are not able to deflect the units in excess from the data. Figure 13.b shows the behavior of the relevance factor during the learning epochs: the relevance of the 4 neurons that identified the Gaussian clusters stabilizes to high values, while that of the loser units slowly drifts towards zero. Again, it can be noted that after an initial phase of strong relevance reduction, there follows a phase in which the loser units *freeze*, never firing for any of the input patterns: hence the relevance reduction becomes dependent only on the exponential decay strategy.

Example 2. The second example (see Fig. 14) is based on a 16 group dataset, where each cluster is generated by a uniform probability on the rectangle. This is a difficult clustering task, since the performance of most of the clustering algorithms will depend on the prototypes' initialization (YW04). In particular, if the seed points are not initialized in each of the clusters, we cannot ensure that a good data partition will be found. CoRe on the other hand is not sensitive to the initialization of the unit prototypes: as it can be seen in Fig. 14.a, it is able to correctly partition the data



(a) CoRe Clusters (b) RPCL Clusters

Figure 14: Clustering results for a 16 group dataset: the datapoints in each cluster are generated with uniform probability on the rectangle. (a) Clusters generated by CoRe from an initial population of 50 neurons. Full circles identify clusters center while ellipsoids define the clusters shape estimate given by the gaussian activation function. (b) Clusters generated by RPCL from an initial population of 20 neurons: 2 seed points (not shown) have been driven away from the data; 2 extra-prototypes are retained.

into 16 groups, identifying the center of each cluster as well as an approximation of the cluster shape. CoRe was run with an initial population of 50 units, with prototypes randomly chosen with uniform probability in the $[0, 7] \times [0, 7]$ interval. The ellipsoids in Fig. 14.a represent the activation functions spread of the cluster detector units learned by CoRe: these results are comparable with that of the SCM algorithm (YW04), although they are obtained with a reduced initial population and without resorting to hierarchical clustering. Fig. 14.b shows the results of RPCL clustering on the 16 group dataset. Here we choose an initial cluster number that is a close guess of the true data partition, i.e. 20 units. The results show that RPCL is able to drive away from the data two units (not shown in the plots). However, it can not partition correctly the data, since two extra-units are retained at the end of the learning epochs (Fig. 14.b).

Example 3. In the third example we evaluated the performance of CoRe clustering on popular benchmark data sets from the UCI Machine Learning Repository[1], that are Iris (Fis36), Wine and Wisconsin Breast

[1]http://www.ics.uci.edu/ mlearn/MLRepository.html

Cancer. All the datasets have been preprocessed in order to normalize the data in the $[0, 1]$ interval and all the trials have been repeated 10 times and the results averaged.

The Iris dataset contains 150 samples, partitioned into three iris categories, that are iris setosa, iris versicolor and iris virginica, where the latter two are not linearly separable. Each class consists of 50 patterns with four continuous features detailing sepal length, sepal width, petal length and petal width. CoRe clustering, with an initial population of 50 random seeds, identified the correct number of clusters and achieved a final error of 6 misclassified patterns. RPCL, on the other hand, was run from an initial population of 4 neurons since, as noted in the previous example, it cannot detect the correct cluster number when initialized with 50 seed points. The RPCL method converged after 100 epochs, having identified 3 clusters and producing 9 data misclassifications. Table 1 (adapted from (XW05)) summarizes the performance of various clustering methods on the IRIS dataset: CoRe outperforms most of prototype-based clustering algorithms and achieves an error performance that's comparable, although inferior, to that of kernel-based models such as Mercer Clustering (Gir02) and SVC (BHHSV02). These two algorithms can separate almost perfectly the two non-linearly-separable clusters because they can build complex separating hyperplanes by exploiting the high dimensional mapping of the input data into the kernel space. In particular, the SVC approach builds class borders passing through the support vector of the identified cluster. However this model does not generate cluster centroids, hence it is useful only if we are not seeking explicitly the cluster representatives. On the other hand, the Mercer Kernel approach clusters the data based on the trace of the within-group scatter matrix of the inputs transformed in an high dimensional feature space induced by a Mercer Kernel. This algorithm, in opposition to SVC, retains the possibility of generating cluster representatives in input space.

A second set of simulations was run on the 13-dimensional Wine data set, in order to study further the performance of CoRe with respect to Mercer clustering, This dataset can be partitioned into three classes, representing different types of wines. Table 2 shows the simulation results:

Table 1: Clustering Results for the IRIS Dataset

Algorithm	Error Number	Error %
GVLQ (PBEK93)	17	11.3%
FCM (RB01)	16	10.6%
GFMM (GB00)	$0-7$	$0-4.7\%$
Mercer Kernel Clustering (Gir02)	3	2%
SVC (BHHSV02)	4	2.7%
CDL (EF98)	6	4%
HC (MV00)	$13-17$	$8.7-11.3\%$
RHC (MV00)	$5-6$	$3.3-4\%$
Fuzzy ART Clustering (BA02)	$6.7-46.4$	$4.5-30.9\%$
SOM Clustering (WC03)	6	4%
K-Means	16	10.6%
RPCL	9	6%
CoRe Clustering	6	4%

(GLVQ) General Learning Vector Quantization - (FCM) Fuzzy c-Means - (GFMM) General Fuzzy Min-Max network - (SVC) Support Vector Clustering - (CDL) Cluster Detection and Labeling network - (HC) Hierarchical Clustering - (RHC) Relative Hierarchical Clustering

again Mercer kernel clustering obtained the best result, identifying three clusters with four misclassifications. On the other hand, CoRe correctly identified the true cluster number (starting from an initial population of 50 neurons), obtaining a performance that is close to that of Mercer clustering and presenting a significant improvement with respect to classical prototype-based algorithms. In general, SVC and Mercer Kernel obtain a better clustering accuracy at the expenses of a non-linear computational complexity. For instance, SVC is $O(K^2)$, where K is the dataset size, while Mercer kernel clustering is $O(K^3)$, since it needs to perform a factorization of the kernel matrix in order to determine the cluster number estimate. On the other hand, the CoRe algorithm is $O(K)$, showing a fair

Table 2: Clustering Results for the WINE Dataset

Algorithm	Error Number	Error %
Mercer Kernel Clustering (Gir02)	4	2.3%
CoRe Clustering	5	2.8%
FCM	9	5%
K-Means	8	4.5%
RPCL	8	4.5%

tradeoff between clustering precision and computational complexity. A wider discussion on the computational issues of CoRe learning is left to Section 2.6.3.

The third set of simulations was run on the 9-dimensional Wisconsin Breast Cancer (WBC) data set collecting 699 cases of diagnostic samples. The dataset contains 16 records with missing values: since CoRe algorithm does not deal with missing values, we deleted the 16 records from the data set and used the remaining 683 records, belonging to two different classes, benign and malignant tumors. Table 3 shows the results of the simulation: again CoRe correctly identified the true cluster number, achieving the lowest clustering error. In particular, CoRe obtained better results than two state-of-the-art kernel clustering approaches, that are the Camastra-Verri Algorithm (CV05) and Ng-Jordan spectral clustering (NJW01).

2.6.3 Computational Issues

CoRe clustering has a computational complexity that is comparable to that of the fuzzy c-means (FCM) (Bez81) family of clustering algorithm (e.g. FCM (Bez81), RCA (FK99),...). Analyzing Algorithm 1 it is clear that CoRe epoch complexity is $O(cKN)$, where K is the number of input samples and N is the number of units. The constant c is a small number that depends on the number of loops that are performed on the K units for each datapoint. In our implementation $c = 3$, that is equal the number

Table 3: Clustering Results for the WISCONSIN BREAST CANCER Dataset

Algorithm	Error Number	Error %
Camastra-Verri Kernel Clustering (CV05)	20.5	3%
Spectral Clustering (NJW01)	31	4.5%
CoRe Clustering	19	2.6%
Neural Gas (CV05)	26.5	3.9%
K-Means	26.5	3.9%
SOM (CV05)	22.5	3.3%
RPCL	31	4.5%

of for-loops that are required to compute the units' activation (first loop in Algorithm 1), the ν_i^t and $\hat{\nu}_i^t$ factors (second loop) and the units' update (third loop). Since initialization is bounded by $O(K)$, the final complexity of the CoRe algorithm is $O(3KN)$: observe that in general, due to the pruning strategy, N is not a constant and decreases from N_{max}, i.e. the initial unit number, to N_{min}, i.e. the number of identified clusters.

Compared with other progressive clustering algorithms (see Section 2.3), CoRe retains a low computational complexity. For instance, the robust RCA algorithm is $O(K \log K + KN)$, where the $K \log K$ term is determined by the penalization factor that is used to prevent overfitting the data set with too many prototypes. SCM (YW04) and, in general, hierarchical clustering algorithms (ESI98) are $O(NK^2)$, where the quadratic complexity term is, again, determined by the cluster number identification process. As noted earlier in Section 2.6.2, kernel and spectral clustering algorithms (e.g. see (Gir02; BHHSV02; CV05; NJW01)) have computational complexities that are at least $O(NK^2)$ since they usually require a factorization of the kernel/laplacian matrix in order to estimate the cluster number. As a general comment, the RS penalization terms is an efficient mean for determining the unknown cluster number from the data while retaining a linear computational complexity that is comparable with that of simple non-agglomerative clustering models such as FCM, k-means and RCPL.

The computational load of CoRe clustering for a single iteration on

Table 4: Computational load for CoRe and RPCL clustering

Dataset	CoRe		RPCL	
	$N = 5$ (Ep)	$N = 10$ (Ep)	$N = 5$ (Ep)	$N = 10$ (Ep)
4 Gaussians	9.23 (0.62)	11.92 (1.02)	4.36 (0.14)	14.68 (0.14)
Iris	5.72 (0.26)	9.75 (0.39)	1.85 (0.05)	9.66 (0.05)
Wine	7.49 (0.34)	15.92 (0.53)	73.9 (0.07)	36.79 (0.07)

the dataset is, certainly, superior to that of RCPL, that is an $O(K)$ algorithm. In order to compare the computational load of the two algorithms and their respective convergence times we report in Table 4 the simulation times (in seconds) obtained for three data sets used in the previous section. The results have been obtained on a 1.7-GHz Centrino machine with 1064-MB memory running Matlab and for two different initial network sizes, i.e. $N = 5$ and $N = 10$. As expected, RPCL is shown to be faster than CoRe on the single iteration through the dataset (see the number between brackets in Table 4). Such load increase is due to the fact that CoRe updates all the units for each datapoint presentation. However, if we look at the total converge time we see that CoRe reduces the computational gap with RPCL. This is due, on the one hand, to the pruning strategy that eliminates units as learning proceeds and, on the other hand, to CoRe's faster convergence rate.

In general, RPCL convergence speed seems to degrade strongly as the number of initial units N increases, probably because the network size growth produces an increase in the number of local minima of the error function, thus delaying its convergence. Notice the different convergence behavior of RPCL on the Wine dataset: repeated simulation with $N = 5$ always produce slower convergence, since the rival penalization mechanism cannot easily get rid of the fourth excess unit. CoRe, on the other hand, seems to be less sensitive to variations in the initial cluster number: chapter 3 will explain this behavior by presenting theoretical motivations for CoRe's faster convergence rate and, in particular, for its ability to avoid local minima of the loss function.

2.7 Conclusion

This chapter introduced the general CoRe model, an unsupervised learning scheme inspired by a memory mechanism of the visual-cortex called repetition suppression. CoRe learning produces sharp feature detectors and compact neural representations of the input stimuli through a mechanism of neural suppression and strengthening that is dependent on the frequency of the stimuli. In particular, we showed how the general CoRe formulation can be used to derive a clustering algorithm that can automatically estimate the cluster number without a-priori information.

In particular, we highlighted CoRe similarities with rival penalized learning, proposing the idea that CoRe clustering represents a soft competitive generalization of RPCL, extending the rival-penalized competition to multiple winner and loser units. Experimental results show how the CoRe soft-rival competitive model overcomes RPCL limitation concerning clusters initialization. RPCL performance, in fact, reduces consistently when the initial seed points number is too distant from the actual cluster number. CoRe, on the other hand, converges to the optimal result also when the initial cluster number is a rough over-approximation of the actual distribution, due to the higher level of competition that is produced between the sets of winner and rival units. Moreover, the repetition suppression effect modulates adaptively unit de-learning. This, on the one hand, makes CoRe more robust with respect to the choice of the learning rates. On the other hand, it produces a parsimonious unit allocation in those regions of the feature space that are more densely populated, in contrast with RPCL, that tends to allocate more units to the densely populated regions of the input space (BS07b).

Experimental results have further shown that CoRe clustering achieves performance results that are comparable with that of state-of-the-art kernel based models, while retaining a linear computational complexity. Moreover, CoRe convergence showed robustness with respect to the choice of the initial cluster number. In particular, CoRe seems resistant to the increase in the number of local minima of the error function produced by a larger initial network size, while RPCL convergence speed degrades

strongly as the number of units increases. In the next chapter we will tackle this aspect from a theoretical point of view, motivating CoRe's ability in avoiding local minima.

In our opinion, competitive repetition-suppression learning can be successfully used as a training mechanism within hierarchical neural network models characterized by articulated architectures, comprising neurons with various activation functions. This is the case, for instance, of several neural models within the machine vision community, such as the classical Neocognitron (Fuk03) and the recent HMAX hierarchical object recognition model by Serre et al (SWB$^+$07). On the one hand, CoRe offers a common formalism and training ground for expressing competition between neurons with different activation functions and characterized by different parameters. On the other hand, the repetition suppression mechanism along with the relevance ranking can be successfully exploited to estimate the number of neurons comprising the single layers of the hierarchical network. Furthermore, all CoRe computations are performed locally to the single layers, hence contributing to maintain the learning complexity low even for strongly stratified structures.

Chapter 3

Theoretical Properties of CoRe Clustering

3.1 Introduction

Chapter 2 has introduced the CoRe learning formulation showing how it can effectively be applied to clustering and cluster number determination. In particular we have focused our analysis on evaluating its performance both in terms of clustering precision and in terms of computational load. In this Chapter, we shift our attention to a theoretically oriented analysis of the CoRe clustering, studying both its properties in terms of robustness to noise and outliers in the data, as well as its convergence behavior. With respect to the former property, we provide theoretical motivations for the robustness to noise and outliers of CoRe's prototype estimation process; additionally, we provide experimental evidence supporting the robustness of CoRe's cluster number identification process when dealing with noisy data.

Previously, it has been highlighted the close relationship between CoRe and Rival Penalized Competitive Learning: in the first part of this chapter, we take a closer look at this association by studying the issue of noise robustness of both algorithms. We will give formal arguments that CoRe can be considered as a robust version of the RPCL algorithm where rival

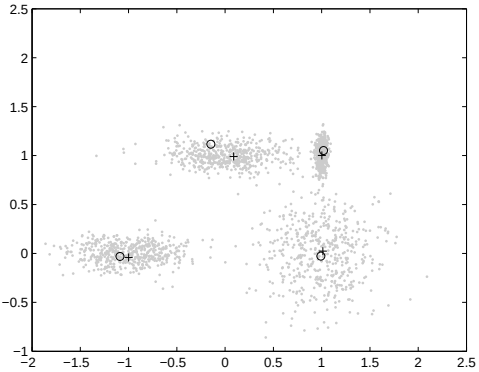


Figure 15: Results for a dataset with 4 ellipsoidal clusters: RPCL prototypes are represented as ∘ and CoRe centroids as $+$.

penalization is modulated by sample repetition. The capital importance of relying on a robust estimation process when dealing with the cluster number identification problem can be well understood by a simple example. Consider, for instance, the scenario in Fig. 15 that shows a dataset comprising four ellipsoidal clusters characterized by different sizes and orientations. Figure 15 shows the cluster centroids identified by CoRe and RPCL for the original data: the two algorithms have been initialized with 20 and 6 neurons, respectively. Such a different initialization is motivated by placing RPCL in the most advantageous scenario since, as we have pointed out in Section 2.6.2, RPCL cannot correctly estimate the cluster number when the initial guess is too large with respect to the ground truth. Figure 15 confirms that both algorithms are able to identify the four cluster centroids, although CoRe seem to be able to cope better with cluster of elliptical shape, achieving a more precise centroid identification. This can be explained by the fact that CoRe's activation function uses a full covariance matrix which naturally models clusters of elliptical shape. Besides the distortion introduced by elliptical clusters, here we are more interested in evaluating the effect of noise on the cluster number estimation task. For this reason, we corrupted the original data by

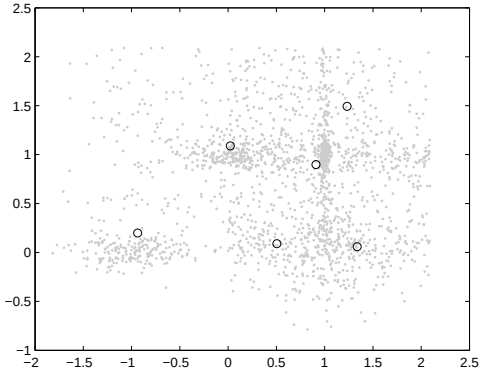
40% multiplicative uniform noise (see Fig. 16). Figure 16.a shows that RPCL now fails to identify the actual cluster number, while the CoRe estimation procedure still identifies the four clusters (Fig. 16.b), with only a slight distortion in the position of the bottom-right centroid, thus showing some kind of robustness with respect to noise and data outliers.

Section 3.2 studies the robustness properties of CoRe clustering from a theoretical point of view, comparing its error function formulation with that of RPCL as well as with other robust clustering models such as RCA (FK99). Section 3.3 presents further experimental results supporting the robustness properties of CoRe clustering.

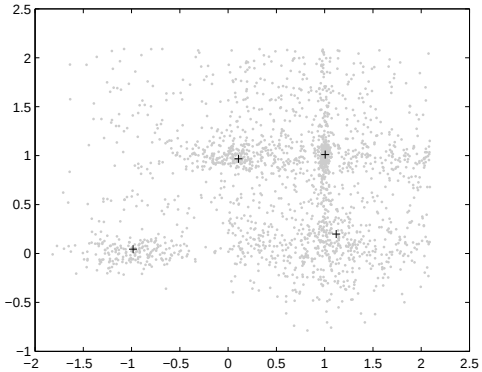
Aside from robustness, another fundamental theoretical issue that needs to be addressed when modeling a novel computational learning system is determining its convergence behavior. In the second part of this chapter, in Section 3.5, we will study the CoRe error formulation to describe its convergence properties with respect to the framework introduced in (MW06) for studying the behavior of the distance-sensitive RPCL (DSRPCL) algorithm. Section 3.4 introduces a kernel-based (STC04) formulation of the CoRe error that is exploited to give theoretically-sound motivations for CoRe's ability of avoiding local minima of the loss function. In particular, we use the kernel error formulation to generalize the results given in (MPGRLRGB02) for hard vector quantization, to kernel-based algorithms. These results will be further expanded in Chapter 4, where we propose a novel kernel vector quantization model that avoids local minima.

3.2 Robust Clustering by CoRe Learning

We derive a batch CoRe error formulation, showing how, besides biological soundness, it posses interesting properties in terms of statistical robustness. Most prototype-based approaches use the Euclidean norm to determine cluster membership, but they suffer from the interference caused by noise and outliers in the data. This interference can be catastrophic in unsupervised clustering, where the actual cluster number has to be estimated from the data. In the following we discuss how CoRe clus-



(a) RPCL



(b) CoRe

Figure 16: Results for the 4 ellipsoidal clusters dataset corrupted by 40% multiplicative uniform noise: (a) RPCL and (b) CoRe clusters.

tering combines the biological inspiration with the robustness properties of the *M-estimators* (Hub81), hence being tolerant to outliers and noise in the data.

3.2.1 Robust Error Function

The incremental formulation of CoRe clustering, given in Section 2.5, reflects the formalization of the general CoRe learning framework and serves to highlight differences and similarities with the RPCL model. In this section, we focus on an alternative, non incremental, characterization which allows to better describe the robustness properties of CoRe clustering and to emphasize its relationships with state-of-the-art robust clustering models.

First, we give an alternative formulation for the residuals in (2.25) and (2.26) by writing a general CoRe error expression for the unit $u_i \in U$ over all the dataset $\chi = (x_1, \ldots, x_k, \ldots, x_K)$ as

$$\mathcal{J}_i(\chi) = \sum_{k=1}^{K} \delta_{ik}(1 - \varphi_i(x_k)) + \sum_{k=1}^{K} (1 - \delta_{ik})(\varphi_i(x_k) RS_k)^2 \qquad (3.1)$$

where we assume that the activation function is the gaussian kernel in (2.29), while the time index t has been omitted from RS_k for notational simplicity. The term δ_{ik} is the indicator function for the set win_k, that is

$$\delta_{ik} = \begin{cases} 1 & \text{if} \quad i \in win_k \\ 0 & \text{if} \quad i \in lose_k. \end{cases} \qquad (3.2)$$

The first term in the objective function in (3.1) tries to minimize the residuals of the winner points, that are those points that are similar enough to the cluster prototype with respect to the given similarity function φ_i and the threshold θ_{win}. Therefore the first term tries to maximize the similarity of the cluster prototype with as many datapoints as possible. The second term, on the other hand, penalizes the cluster for those datapoints that are not similar enough, counteracting the effect of the first term. In addition, the RS_k factor ensures that the the penalization is higher for those patterns that have been assigned to another cluster.

Since the units can be treated as independent, we can generalize the objective function in (3.1) by summing up over all CoRe units $U = (u_1, \ldots, u_i, \ldots, u_N)$, obtaining

$$\mathcal{J}(\chi, U) = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}(1-\varphi_i(x_k)) + \sum_{i=1}^{N}\sum_{k=1}^{K}(1-\delta_{ik})(\varphi_i(x_k)RS_k)^2. \quad (3.3)$$

Following the procedure in (FK99), we can rewrite the CoRe error $\mathcal{J}$ in terms of *loss functions* ρ_i and in terms of *typicality measures* $\hat{w}_{ik}$, yielding to

$$\mathcal{J}_R(\chi, U) = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\rho_i(d_{ik}^2) + \sum_{i=1}^{N}\sum_{k=1}^{K}(1-\delta_{ik})\hat{w}_{ik}^2 RS_k^2. \quad (3.4)$$

In (3.4), $\rho_i(d_{ik}^2)$ is the loss function associated to the i-th cluster, while $\hat{w}_{ik}$ is a typicality term proportional to ρ_i's derivative with respect to d_{ik}^2, that is

$$\hat{w}_{ik} = \alpha_i \cdot \frac{\partial \rho_i(d_{ik}^2)}{\partial d_{ij}^2}.$$

Both $\rho_i()$ and $\hat{w}_{ik}$ depends on the the term d_{ik}^2, that is a measure of the distance between the input vector x_k and the cluster prototype c_i. In the following we show that, under the assumption that the CoRe activation function is the gaussian kernel in (2.29), the term $\rho_i()$ in the CoRe error formulation (3.4) can be interpreted as the loss function of a robust M-estimator (see definition and properties in (Hub81), Section 3.2, pp. 43–55).

The general idea of robust estimation can be well described in terms of two desirable properties identified by Huber (Hub81). In his idea, a statistical procedure for fitting a non parametric model, such as in centroid estimation, should be robust in the sense that

1. small deviations from model assumptions should impair the performance only slightly;

2. large deviations from the model should not cause catastrophic interference.

In our scenario, noise and outliers play the role of the deviations from the original model, which is characterized by a number of random data sources that we are willing to discover by finding an appropriate clustering of the observation data.

Maximum likelihood estimators, or, in short, *M-estimators*, are generated as solution of a minimum problem of the form

$$\min_{\theta} \sum_{k=1}^{K} \rho(x_k; \theta), \tag{3.5}$$

where ρ is an arbitrary function that measures the loss of x_k and θ. In particular, we are often interested in finding a location estimate $\hat{\theta}$ minimizing

$$\sum_{k=1}^{k} \rho(x_k - \theta). \tag{3.6}$$

The corresponding M-estimator can be generated by solving the analogous problem

$$\sum_{k=1}^{K} w_j(x_k - \theta) = 0, \tag{3.7}$$

where

$$w_k = (x_k - \theta)^{-1} \frac{\partial \rho(x_k - \theta)}{\partial \theta} \tag{3.8}$$

is called the weight function or W-estimator, since the resulting M-estimator $\hat{\theta}$ can be written as the weighted mean

$$\hat{\theta} = \sum_{k=1}^{K} \frac{w_k}{\sum_{l=1} w_l} x_k, \tag{3.9}$$

where sample contributions are *weighted* by the W-estimator.

Following this definition, the CoRe error formulation in (3.3) and (3.4) can be restated in terms of K M-estimators for the prototypes c_i given by the following loss functions

$$\rho_i(x_k - c_i) = (1 - \varphi_i(x_k)) = (1 - e^{-\frac{\|x_j - c_i\|^2}{2\sigma_i^2}}), \tag{3.10}$$

where the corresponding W-estimators can be written as

$$w_{ik} = -\sigma_i^2 \exp(-\frac{\|x_k - c_i\|^2}{2\sigma_i^2}). \tag{3.11}$$

In the step from (3.4) to (3.10), we have implicitly used the assumption $d_{ik} = \|x_k - c_i\|$. By applying this substitution to the definition of $\hat{w}_{ik}$, we obtain

$$\hat{w}_{ik} = \alpha_i \frac{\partial \rho_i(d_{ik}^2)}{\partial d_{ij}^2} = \alpha_i w_{ik}, \tag{3.12}$$

where $\alpha_i = \sigma_i^{-2}$ discounts the contribution of the variance magnitude from the weight function. Roughly, from the formulation of the objective function in (3.3) and (3.4), we can say that CoRe clustering corresponds to a combination of K M-estimators with loss ρ_i, whose estimates are perturbed by a penalization that is proportional to the corresponding W-estimators.

In order to assess the statistical robustness of the CoRe M-estimator we need to study the behavior of the its *influence curve*. This function has been proposed by Hampel (Ham74) to measure the relative influence of individual observations on the value of the estimate. In (Hub81) is shown that the influence function of an M-estimator is proportional to the derivative of its loss function $\psi(x - \theta) = \partial\rho(x - \theta)/\partial\theta$. In the context of the location estimate in (3.6), we can thus write the influence function as

$$IC(x, F, \theta) = \frac{\psi(x - \theta)}{\int \psi'(x - \theta) dF_X(x)} \tag{3.13}$$

where $F_X(x)$ is the distribution over X. If the influence function is unbounded, then an outlier can have a large relative influence on the estimate, resulting in a strong deviation from the asymptotic solution. An interesting statistics based on the influence functions is its maximum absolute value

$$\gamma^* = \sup_x |IC(x, F, \theta)| \tag{3.14}$$

that is called *gross error sensitivity* and measures the worst approximate influence that the addition of an infinitesimal point mass can have on the value of the estimator (YW04). Since the influence function in (3.13)

is proportional to the derivative $\psi(x-\theta)$, we can study its behavior by analyzing ψ following the procedure in (WY02). From the definition in (3.10), we obtain that CoRe's ψ function is given by

$$\psi(x-c) = \frac{-2\sigma^{-2}(x-c)}{\exp(\frac{\|x-c\|^2}{2\sigma^2})} \tag{3.15}$$

where the unit index i has been omitted to ease the notation. To describe the asymptotic behavior of (3.15) we need to study the limit of $\psi(x-c)$ for $x \to \infty$ and $x \to -\infty$. A straightforward application of the L'Hospital's rule yields

$$\lim_{x\to\infty} \psi(x-c) = 0 \quad \text{and} \quad \lim_{x\to-\infty} \psi(x-c) = 0, \tag{3.16}$$

thus the influence function approaches zero whenever x tend to positive or negative infinity. We can also obtain maximum and minimum values of ψ by applying the necessary condition

$$\frac{\partial\psi(x-c)}{\partial x} = 0. \tag{3.17}$$

Given these results we obtain that the M-estimators defined by (3.10) have a bounded and continuous influence function and a finite gross error sensitivity: therefore the estimators based on (3.10) are robust with respect to Huber's definition (Hub81). According to these results, CoRe's objective function in (3.3) combines a loss function of a robust M-estimator with a penalization term proportional to the corresponding W-estimator. The first term reduces the effect of noise and outliers on the estimate, while the second term prevents the cluster at hand from attracting significant points that have already been assigned to another cluster. Therefore CoRe implements a robust cluster estimation process that is resistant to noise and outliers in the data.

Apart from proving robustness properties, the error formulation in (3.3) can be used to derive a batch version of the CoRe clustering algorithm, namely the Batch Competitive Repetition-suppression (BCoRe). In order to derive the batch learning rules of BCoRe, we apply the necessary condition for the minimization of $\mathcal{J}(\chi, U)$, differentiating (3.3) with

respect to all c_i, that is

$$\frac{\partial \mathcal{J}(\chi, U)}{\partial c_i} = \sum_{k=1}^{K} \delta_{ik} \varphi_i(x_k) \frac{(x_k - c_i)}{\sigma_i^2} + - \sum_{k=1}^{K} (1 - \delta_{ik})(\varphi_i(x_k))^2 R S_k^2 \frac{(x_k - c_i)}{\sigma_i^2} = 0. \quad (3.18)$$

Hence, the necessary condition that minimizes $\mathcal{J}(\chi, U)$ becomes

$$c_i = \frac{\sum_{k=1}^{K} S_{ik} x_k}{\sum_{l=1}^{K} S_{il}}, \quad (3.19)$$

where the S_{ik} values are calculated as follows:

$$S_{ik} = S(c_i, x_k) = \delta_{ik} \varphi_i(x_k) - (1 - \delta_{ik})(\varphi_i(x_k))^2 R S_k^2. \quad (3.20)$$

The expression in (3.19) estimates the cluster prototypes as the weighted mean of the sample vectors, where the mixing weights S_{ik} are modulated by the robust W-estimator $\varphi_i(x_k)$ and a penalization term. The spread estimate is determined similarly as the weighted variance with respect to the current mean estimate c_i and the input vectors x_k, that is

$$\sigma_i^2 = \frac{\sum_{k=1}^{K} S_{ik} \|x_k - c_i\|^2}{\sum_{l=1}^{K} S_{il}}. \quad (3.21)$$

Moreover, in the general case of a symmetric covariance matrix Σ_i (FK99), (3.21) can be reformulated as

$$\Sigma_i = \frac{\sum_{k=1}^{K} S_{ik} (x_k - c_i)(x_k - c_i)^T}{\sum_{l=1}^{K} S_{il}}. \quad (3.22)$$

The estimates in (3.19) and (3.21) cannot be solved directly, but can be approximated by means of the *fixed point iterative method*. Let τ be the epoch counter, we take the left hand side of (3.19) and (3.21) as the center and spread solutions at epoch $\tau + 1$. Their estimate is calculated by evaluating the right hand side of the equations at the epoch τ. This process is iterated until a certain stability criterion is met. Summarizing, the BCoRe algorithm can be described as follows:

Initialize c_i^0, σ_i^0; set iteration counter $\tau = 0$.

Do Repeat

Step 1 Estimate $S_{ik}^{\tau+1}$ by (3.20);

Step 2 Estimate $c_i^{\tau+1}$ by (3.19);

Step 3 Estimate $\sigma_i^{\tau+1}$ by (3.21)

Until $max_i \|c_i^{\tau+1} - c_i^{\tau}\| < \epsilon$; Increment τ.

Here, the iterative procedure is stopped whenever the prototypes estimate at time $\tau + 1$ is very close to the solution generated at the previous epoch.

3.2.2 Relationship with Other Models

The error formulation in (3.4) can be used to compare CoRe with relevant robust clustering models in the literature. Consider, for instance, the Robust Competitive Agglomeration (RCA) (FK99) algorithm described in Section 2.3: its objective function can be written as

$$\mathcal{J}_{rca}(B, V, \chi) = \sum_{i=1}^{N}\sum_{k=1}^{K}(v_{ik})^2\rho_i(d_{ik}^2) - \alpha\sum_{i=1}^{N}\left(\sum_{k=1}^{K}w_{ik}v_{ik}\right)^2. \qquad (3.23)$$

The term v_{ik} represents the degree to which the point x_k belongs to the cluster identified by the prototype β_i and is subject to the constraint

$$\sum_{i=1}^{N} v_{ik} = 1.$$

The expression $V = [v_{ik}]$ identifies the constrained fuzzy partition matrix, while B is the matrix of the cluster prototypes β_i. The first term in the RCA objective function is essentially the sum of the fuzzy intra-cluster distances mediated by a robust loss function ρ_i. This term induces overfitting, since it is minimized when the number of clusters is equal to the number of the datapoins. The second term prevents overfitting by imposing a penalization that is proportional to the cardinality of the clusters

and that is mediated by w_{ik}, that is the W-estimator generated by the loss function ρ_i. This term discounts outliers when computing the clusters cardinality, hence making the estimation more robust.

The RCA algorithm determines the cluster number by aggregating an initially over-specified set of prototypes. This agglomeration depends on the balance between the first and the second term in (3.23), that is controlled by the mixing parameter α. In (FK99), α is chosen to be dependent on the iteration step. In the first phase, it is chosen to be low in order to encourage the formation of small clusters. Then, α is increased gradually to promote agglomeration and, finally, it is decreased again to allow the convergence of the algorithm. RCA agglomeration resembles the CoRe neuron suppression process controlled by the RS_k factor. Repetition suppression is initially low, allowing neurons to identify their reference stimulus, and increases as soon as the units start to be tuned on specific patterns, finally decreasing to zero when there is little overlapping in the temporal activation of the units. More formally, if we compare the objective functions in (3.4) and (3.23), we notice that RS_k plays a similar role to α in determining the balance between the robust loss ρ_i and the penalization in the second term.

The error formulation in (3.3), highlights CoRe similarity with another robust clustering method, that is the *alternative hard c-means* (AHCM) algorithm (WY02). The AHCM error function is

$$\mathcal{J}_{ahcm}(B,\chi) = \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}^{1}\left(1-\exp(-\delta\|x_k-\beta_i\|^2)\right) = \sum_{i=1}^{C}\sum_{k=1}^{K}\delta_{ik}^{1}\rho_i(d_{ik}^2) \tag{3.24}$$

where δ_{ik}^1 is non-zero only when i is the index of the best matching unit for the pattern x_k. The formulation in (3.24) depends solely on the first term in (3.3), i.e. the robust loss determined by the gaussian kernel, and does not include any penalization term. Consequently, AHCM cannot determine automatically the number of clusters from the data. The same authors proposed, in (YW04), an alternative formulation based on the maximization of a gaussian similarity metric, namely the *similarity-based robust clustering method* (SCM). Again, SCM does not use any penalization term

in its objective function, but resorts to agglomerative hierarchical clustering in order to determine the number of clusters once the prototype positioning phase has converged.

The original RPCL model does not define an error function for the rival penalization learning. Recently, Ma and Wang (MW06) published a generalization of the RPCL algorithm, named distance-sensitive RPCL (DSRPCL), defined in terms of the following cost function

$$\mathcal{J}_{dsrpcl}(B,\chi) = \frac{1}{2}\sum_{k=1}^{K}\|x_k - \beta_{c(k)}\|^2 + \frac{2}{P}\sum_{k=1}^{K}\sum_{i=1,i\neq c(k)}^{N}\|x_k - \beta_i\|^{-P}. \quad (3.25)$$

In (3.25), β_i identify the cluster prototypes, while $c(k)$ denotes the index of the winner unit in the WTA competition and P is a positive number.

The first term of the DSRPCL cost function is proportional to the distance between the current sample x_k and the winner weight $\beta_{c(k)}$, playing the same role as the first term in the CoRe error formulation in (3.4). In addition, the CoRe term corresponds to a robust version of the DSRPCL estimate obtained by means of a Gaussian weight function. On the other hand, the second term of the DSRPCL cost is inversely proportional to the distance between the sample point and the loser weights, serving to counteract overfitting such as in the second term of the CoRe error in (3.4). The CoRe penalization is, in fact, inversely proportional to the dissimilarity (measured by the gaussian kernel φ_i) between the current sample and the loser neurons. The repetition suppression factor RS_k serves to modulate the penalization proportionally to the relative density of the samples constituting a cluster, i.e. more frequent patterns experience more suppression. The DSRCPL penalization, on the other hand, depends solely on the distance between the loser weights and the samples.

These robustness issues continue to be present in the recent generalization of the RPCL framework (Xu07) described in Section 2.3. In fact, the reformulated RPCL error criterion in (2.14) mixes robust Gaussian functions with non-robust Euclidean metrics, producing estimators that fail to meet the robustness conditions detailed above, that are the boundedness of the influence function and the finiteness of gross error sensitivity. Summarizing, CoRe can be considered as a soft and robust version

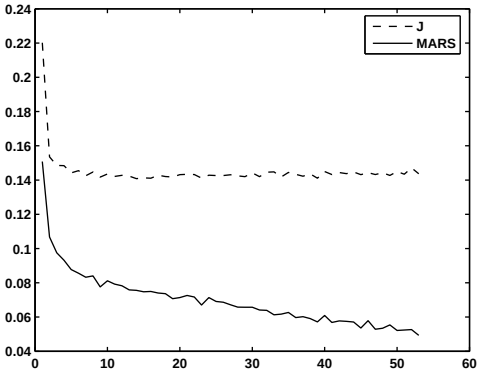


Figure 17: CoRe error (dashed) and MARS score (solid) for the four Gaussians task (Example 1). Core error has been averaged over the data set size to report it on the interval of the MARS score. The convergence criteria was $MARS < 0.05$: notice that convergence happens when the CoRe error has already reached a stable state.

of the DSRPCL (and RPCL) algorithm where rival penalization is modulated by sample repetition. Figure 17 shows an example of the behavior of the CoRe error $\mathcal{J}$, as defined in (3.3), with respect to the MARS score (see (2.39)) during training on the four gaussians dataset (see Section 2.6.1).

3.3 Experimental Evaluation

In this section we analyze the robustness properties of CoRe learning through a set of experiments on the unsupervised segmentation of noisy images. In particular, we are interested in understanding the role of robust estimation in tasks where the optimal cluster number is unknown and to study the effects of the overestimation of the cluster number in noisy environments.

In the first experiment, we applied CoRe and RPCL to the segmentation of a 64×64 synthetic image consisting of four gray-level classes of intensity 0, 80, 160 and 240. First, we tested the two algorithms with

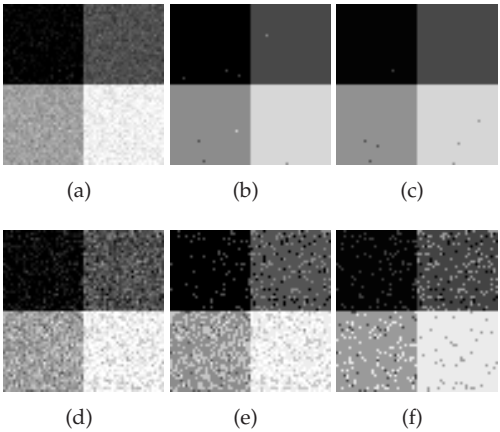


Figure 18: Comparison of segmentation results on a synthetic four class image under varying levels of noise: (a) Original image perturbed by 5% noise, (b) RPCL result, (c) CoRe result; (d) Original image perturbed by 11% noise, (e) RPCL result, (f) CoRe result.

the original image corrupted by 5% gaussian noise (Fig. 18.a) and an initial population of 10 seed points. Figure 18.b and 18.c show the results of CoRe and RPCL segmentation, respectively. A quick visual inspection of the results shows that both algorithms identified correctly 4 clusters, converging nearly to the same segmentation accuracy (i.e. the percentage of correctly segmented pixels in the reconstructed image). However, if we corrupt the same image with 11% noise (Fig. 18.d), we obtain two different behaviors for RPCL and CoRe. The former algorithm retains 3 spurious clusters, resulting in a strong degradation of the segmentation performance that drops to 70.6% (see Fig. 18.e). CoRe, on the other hand, converges to the optimal cluster number, achieving a segmentation performance of 88.8% (see Fig. 18.f). It appears that a low level of noise does not interfere significatively with the cluster identification process, while the robustness of CoRe estimation seems to be fundamental when dealing with considerable noise levels.

The second experiment compares the segmentation results of RPCL, FCM (Bez81) and CoRe clustering applied to a T1-weighted MR phan-

tom image from the *BrainWeb Simulated Brain Database*[1] (SBD). The SBD contains a set of realistic MRI data volumes produced by an MRI simulator. For our experiment we used a high-resolution 181×217 T1-weighted phantom image with slice thickness of 1mm, 3% noise and no intensity inhomogeneity. The SDB provides a discrete anatomical model of the image compromising the cerebro-spinal fluid (CSF), gray matter (GM), white matter (WM) and background. This model has been used to calculate the segmentation accuracy and to compare the results obtained by CoRe and RPCL. Both models have been initialized with 10 seed points and trained pixel by pixel on the noisy 181×217 image. At the end of the training phase both algorithms have selected four clusters: one each for CSF, GM, WM and one for the background . As detailed in Table 5, CoRe and RPCL have, on this task, a similar performance in terms of *comparison score*, that is

$$s_c = \frac{A_c \cap A_c^{ref}}{A_c \cup A_c^{ref}} \tag{3.26}$$

where A_c represents the set of pixels assigned to the class c by the learning algorithm and A_c^{ref} is the reference set of pixels for class c. This result is not surprising since the level of noise in the image is still low, so that RPCL, although not based on a robust M-estimator, achieves good results. In order to compare the performance of the two algorithms under higher levels of noise, we ran the simulation on the same image distorted by a 7% gaussian noise. Fig. 19 shows the original noisy image and the RPCL and CoRe segmentation. The results in Fig. 19 show that CoRe and RPCL have a similar segmentation performance on the white matter and gray matter, while RPCL have a consistently worse performance on the CSF segmentation. This visual impression is confirmed by the comparison scores in Table 5: RPCL experiences a consistent drop in the segmentation performance due to the presence of noise. CoRe, on the other hand, maintains an acceptable segmentation performance, in particular with respect to the CSF, since the robust similarity function used for calculating the units' activation is less affected by noise than the Euclidean measure used by RPCL.

[1] http://www.bic.mni.mcgill.ca/brainweb/

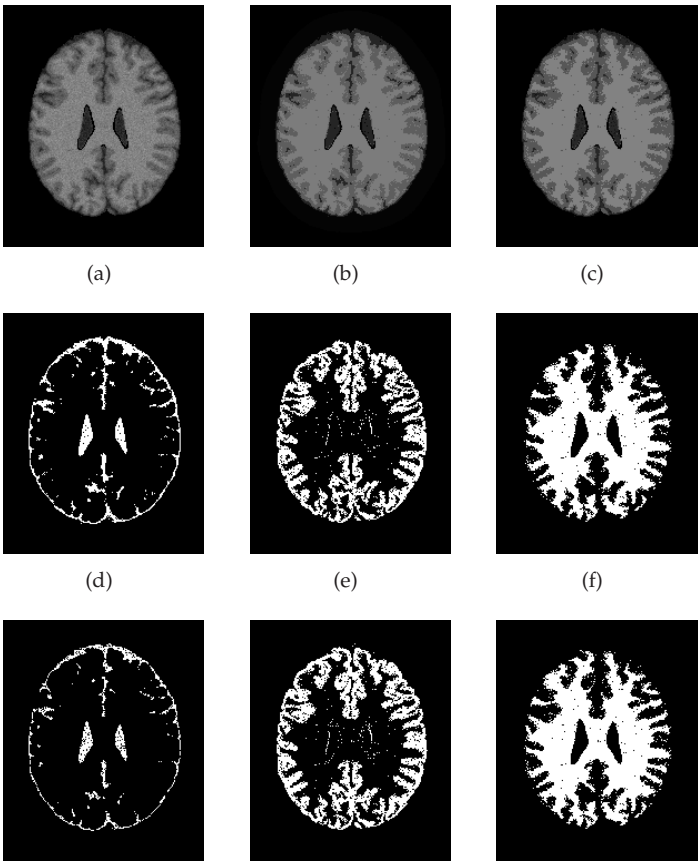


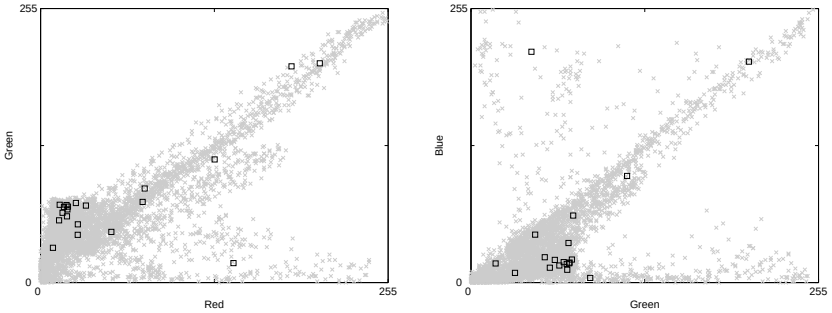
Figure 19: (a) Original image (100th slice) distorted by 7% gaussian noise. Noisy image reconstructed by CoRe (b)and RPCL (c) segmentation. CoRe (d) and RPCL (g) cerebro-spinal fluid segmentation. CoRe (e) and RPCL (h) gray matter segmentation, CoRe (f) and RPCL (i) white matter segmentation.

Table 5: Comparison Scores for the Simulated MRI

Algorithm (Noise)	WM	GM	CSF
CoRe (3%)	95.9%	90.7%	89.7%
RPCL (3%)	96.2%	90.7%	87.5%
CoRe (7%)	88.8%	78.4%	82.3%
RPCL (7%)	88.8%	77.9%	70.8%

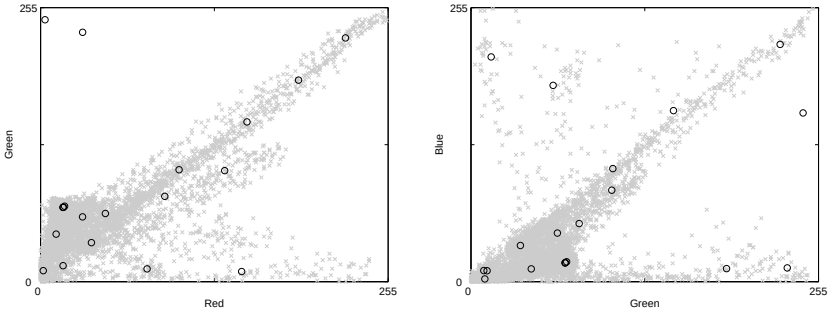
The particularly poor CSF segmentation obtained by RPCL can be explained by its incorrect background cluster allocation. In fact, RPCL assigned 3 different units to the background pixels, that are those with the gray-level that is most similar to the CSF, leading to a gapy segmentation of the cerebro-spinal fluid (see Fig. 19.g). Conversely, CoRe converged to the optimal cluster number, allocating a single unit to the background pixels. The results of this experiment exemplify the difference existing between the two algorithms: RPCL has the tendency to allocate more units to the densely populated regions of the input space (MW06). CoRe, on the other hand, tends to reduce the overall units activation in correspondence to the more frequent patterns, therefore producing a parsimonious unit allocation in those regions of the feature space that are more densely populated. In other words, taking into consideration our MR segmentation example, we see that the biggest portion of the feature space is related to the image background: as a consequence, RPCL tends to over-allocate clusters in that area, while CoRe selects only a single background feature detector.

Figure 20 illustrates this phenomenon with respect to the segmentation of the 128×128 pool image (PHK$^+$00). RPCL and CoRe clustering were run with 20 initial seed points, with CoRe unit pruning disabled to allow a fair comparison between the two cluster allocation strategies. The results in Fig. 20.a and 20.b show that the majority of RPCL prototypes is displaced in the area of the color component space that has the highest density of data points and that corresponds to the table and the background colors in Fig. 21.c and 21.d. Conversely, in the same area, CoRe allocates only 2 units (see the components plot in Fig. 20.c and



(a) RPCL - R and G Components

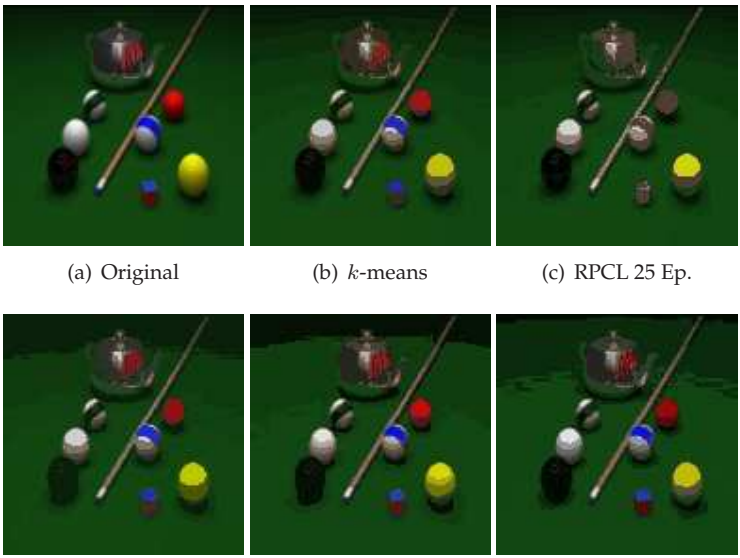
(b) RPCL - G and B Components



(c) CoRe - R and G Components

(d) CoRe - G and B Components

Figure 20: RGB Color distribution ($\star$) and cluster centers for the pool image: (a-b) RPCL prototypes ($\square$) and (c-d) CoRe prototypes ($\circ$) w.r.t. the RG and GB components.



(a) Original (b) k-means (c) RPCL 25 Ep.

(d) RPCL 100 Ep. (e) CoRe 10 Ep. (f) CoRe 23 Ep.

Figure 21: Segmentation of the pool image: (a) Original image, (b) reference k-means segmentation for $k = 16$, RPCL results after 25 (c) and 100 (d) epochs, CoRe segmentation after 10 (e) and 23 (f) epochs.

20.d). CoRe realizes a more uniform color quantization, identifying the 2 low density clusters at the top-right in Fig. 20.c that differentiate the blue-stripe of the ball from the blue-powder on the dice and on the tip of the pool stick (Fig. 21.e and 21.f). In Fig. 21.b it is shown a reference segmentation realized using the k-means algorithm (with $k = 16$): again, the majority of the prototypes concentrate on the denser area of the color space, leaving few seed points quantizing colors different from green (e.g. see the wrong dice color in Fig 21.b).

Figure 22 depicts the segmentation performance of four methods on the MR image distorted by increasing levels of noise. RPCL and CoRe have been initialized with 5 seed points, while fuzzy c-means was tested with 4 (FCM4) and 5 (FCM5) initial clusters. The results in Fig.22 underline the key role of a robust cluster identification process when dealing with noisy data. In particular, the presence of an extra cluster, influences negatively the performance of FCM5 and RPCL as the level of noise increase. Initially both methods are able to cope with a small amount of noise, achieving a segmentation performance that is comparable with that of FCM4. Soon FCM5 performance starts to deteriorate, while RPCL shows an higher robustness due to the rival penalization, and its performance drops consistently only above 9% noise. Conversely, a small overestimation of the initial cluster number does not affect CoRe performance. In particular, when noise is under 9%, CoRe has a classification score that it's comparable with that of FCM4, while, when the noise level exceeds 9%, CoRe shows a superior performance thanks to the use of a robust M-estimator in its error function.

3.4 A Kernel Based Loss Function for CoRe Clustering

Previously, it has been shown how the CoRe loss formulation in (3.3) can be used to study the robustness properties of CoRe clustering. Besides this central theoretical issue, the loss formulation allows to analyze also the convergence properties of a CoRe learning process. In order to proceed with this analysis we need to introduce an alternative formulation

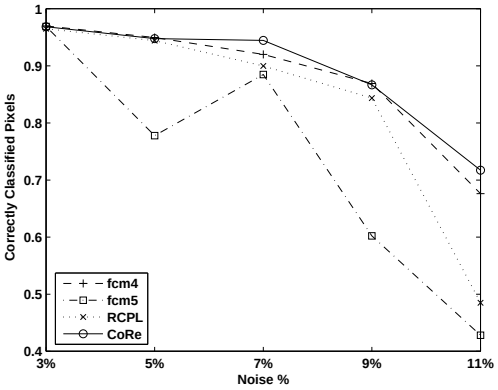


Figure 22: Comparison of classification scores on a noisy MR image for five methods: fuzzy c-means with 4 seed points (fcm4), fuzzy c-means with 5 seed points (fcm5), RPCL and CoRe with 5 seed points.

for the CoRe error in (3.3) based on *kernel induced distance metrics* (STC04). Before giving the details of such alternative formulation we briefly introduce the fundamentals of kernel methods.

3.4.1 Introduction to Kernel Methods

Kernel methods are algorithms that exploit a convenient mapping of the data onto an implicit higher dimensional space, i.e. the *kernel* or *feature space*, to perform linear operations in the mapped space that result in non-linear operations in the input space.

A *kernel* is a symmetric positive semi-definite function $\kappa : (\chi \times \chi) \longrightarrow \mathbb{R}$ that, given a set χ represents its elements as a pairwise comparison matrix $K_{ij} = \kappa(x_i, x_j)$, where $x_i, x_j \in \chi$. A fundamental theorem by Aronszajn (Aro50) states that κ is a kernel on the set χ if and only if there exist an Hilbert space $\mathcal{F}_\kappa$ and a mapping

$$\Phi : \chi \rightarrow \mathcal{F}_\kappa \tag{3.27}$$

such that, for all $x_i, x_j \in \chi$

$$\kappa(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle. \tag{3.28}$$

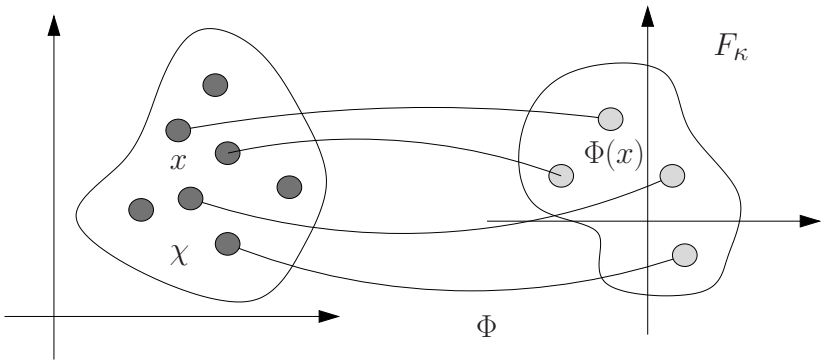


Figure 23: Every kernel κ on the space χ determine a nonlinear map Φ to an implicit feature space $\mathcal{F}_\kappa$.

The embedding function $\Phi : \chi \rightarrow \mathcal{F}_\kappa$ determines a nonlinear map from the *input space* χ to the *feature space* $\mathcal{F}_\kappa$, equipped with an inner product $\langle \cdot, \cdot \rangle : \mathcal{F}_\kappa \times \mathcal{F}_\kappa \longrightarrow \mathbb{R}$. Figure 23 graphically depicts the embedding from the input space to the kernel space via the mapping function Φ.

In general, neither the nonlinear map Φ nor the embedding space $\mathcal{F}_\kappa$ needs to be explicitly defined, since operations in $\mathcal{F}_\kappa$ can be expressed in terms of the inner product $\langle \Phi(x_1), \Phi(x_2) \rangle$. Hence, they can be computed by means of the kernel function alone, without explicitly knowing or using the mapping Φ. This approach allows to retain the power of the high dimensional mapping without the computational burden of performing operations with high dimensional vectors. Amongst the most commonly used kernels in the literature are

- Gaussian kernel

$$\kappa(x_1, x_2) = e^{\left(\frac{-\|x_1 - x_2\|^2}{\sigma^2} \right)} \tag{3.29}$$

- Polynomial kernel

$$\kappa(x_1, x_2) = (\ 1 + \langle x_1, x_2 \rangle)^d \tag{3.30}$$

- Sigmoid kernel

$$\kappa(x_1, x_2) = \tanh(\alpha \langle x_1, x_2 \rangle + \beta). \tag{3.31}$$

These kernels are all defined on vectorial data comprising real valued components: however, provided that we know how to define a suitable positive semi-definite symmetric function, we can apply kernels to objects of various nature, including strings, sequences as well as more complex, structured data, such as graphs.

Given that every kernel κ defines an Hilbert space $\mathcal{F}_\kappa$ characterized by an inner product metric $\|\cdot\|_{\mathcal{F}_\kappa}$, we can define a metric on $\mathcal{F}_\kappa$ as $\|\Phi(x)\|^2_{\mathcal{F}_\kappa} = \langle \Phi(x), \Phi(x) \rangle$. Based on this representation, every algorithm defined in terms of the Euclidean norm can be restated as a kernel based model: such an approach is known in the literature as the *kernel trick* (SSM98). By this means, the nonlinear map Φ induced by κ can be used to embed the data onto an implicit high dimensional feature space $\mathcal{F}_\kappa$, such that linear operations in $\mathcal{F}_\kappa$ result in nonlinear operations in the input space.

Without loss of generality, in the remainder of the chapter we will consider only *normalized kernels* ((GB01)). Such kernels implement a normalization in the embedding space so that mapped vectors lie on a portion of the unit hyper-sphere in the feature space. There is no loss of generality in this approach, since normalized kernels can be built out of any kernel function $\kappa(x_1, x_2)$ by

$$\hat{\kappa}(x_1, x_2) = \frac{\kappa(x_1, x_2)}{\sqrt{\kappa(x_1, x_1)\kappa(x_2, x_2)}}. \tag{3.32}$$

Normalized kernels can effectively be considered as similarity metrics. Consider, for instance, a generic kernel κ and the associated Hilbert space norm $\|\cdot\|_{\mathcal{F}_\kappa}$. Using the inner product definition of the norm we can write the feature space distance between two samples as

$$\begin{aligned} d(\Phi(x_1), \Phi(x_2)) &= \|\Phi(x_1) - \Phi(x_2)\|^2_{\mathcal{F}_\kappa} = \|\Phi(x_1)\|^2_{\mathcal{F}_\kappa} + \|\Phi(x_2)\|^2_{\mathcal{F}_\kappa} \\ &\quad -2\langle \Phi(x_1), \Phi(x_2) \rangle. \end{aligned} \tag{3.33}$$

Since we are dealing with normalized kernels it holds that $\|\Phi(x)\|^2_{\mathcal{F}_\kappa} = 1$ for every $x \in \chi$; then, (3.33) can be rewritten as

$$d(\Phi(x_1), \Phi(x_2)) = \|\Phi(x_1) - \Phi(x_2)\|^2_{\mathcal{F}_\kappa} = 2 \cdot (1 - \kappa(x_1, x_2)) \tag{3.34}$$

Equation (3.34) means that a normalized kernel is a decreasing measure of the feature-space distance between x_1 and x_2: hence, κ is a similarity

measure. The equality in (3.34) will be fundamental, in the next section, to derive the kernel based loss function for CoRe clustering and to analyze its convergence behavior in Section 3.5.

There exist a vast literature of kernel-based learning models, such as Support Vector Machines (SVM) or Kernel Principal Component Analysis (KPCA). In the latter years, several classical clustering algorithms have been extended to the kernel approach by exploiting the kernel trick. In (SSM98) it was first introduced the idea of a *kernel k-means* algorithm, that exploit the kernel trick to produce a k-means clustering where the vectors distances are measured in the feature space. This approach is able to identify correctly nonlinearly separable clusters, by projecting them on the kernel space and identifying the linearly separable clusters in the high dimensional space. In (DGK04) this model has been generalized by defining the *weighted kernel k-means* model, that is shown to include *spectral clustering* (NJW01) as a special case. Also the Rival Penalized Competitive Learning (RPCL) algorithm (XKO93) has been extended the kernel case (ZC04b). In (ZC04a), it is presented kernel based version of the *fuzzy c-means* algorithm that exploits the kernel trick in a different way with respect to the aforementioned models: instead of defining the cluster prototypes as a weighted combination of the vectors $\Phi(x_i)$, the kernel c-means estimates the prototype vectors in the input space, thus retaining a clear geometrical interpretation of the learning results in the input space.

3.4.2 Kernel Loss

The error formulation introduced in (3.3) can be restated by exploiting the kernel trick to express the CoRe loss in terms of differences in a kernel-induced feature space $\mathcal{F}_\kappa$. First of all, we need to restrict CoRe activation functions to be normalized kernels κ_i: the Gaussians used in CoRe clustering are themselves normalized kernels, so we can straightforwardly apply the kernel trick to (3.3).

Recall that, given a normalized kernel κ, we can write the feature space distance between two samples x_1 and x_2 as

$$d_{F_\kappa} = \|\Phi(x_1) - \Phi(x_2)\|^2_{\mathcal{F}_\kappa} = 2 - 2\kappa(x_1, x_2).$$

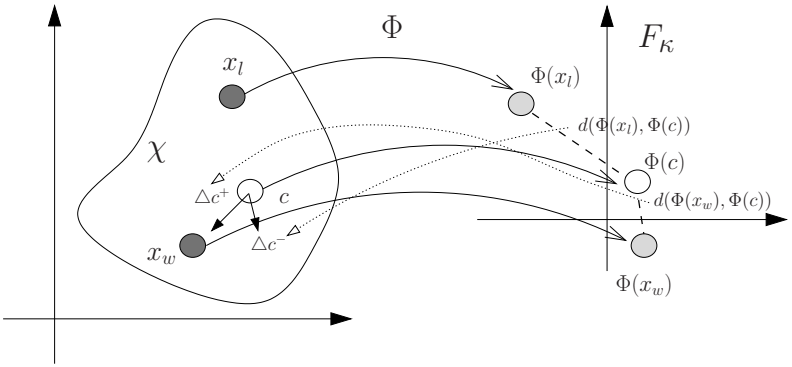


Figure 24: A kernel-based interpretation for CoRe attractive and repulsive learning. The prototype c as well as the samples x_w and x_l are mapped to the feature space $\mathcal{F}_\kappa$: winners and losers are determined based on distances in kernel space. The resulting attractive and repulsive updates, respectively $\triangle c^+$ and $\triangle c^-$ are modulated by the kernel defined similarity.

Now, if we take x_1 to be an element of the input dataset, e.g. $x_k \in \chi$, and x_2 to be the prototype c_i of the i-th CoRe unit, we can rewrite d_{F_κ} in such a way to depend on the activation function φ_i. Therefore, by applying the substitution $\kappa(x_k, c_i) = \varphi_i(x_k, \{c_i, \sigma_i\})$, we obtain $\varphi_i(x_k, \{c_i, \sigma_i\}) = 1 - \frac{1}{2}\|\Phi(x_k) - \Phi(c_i)\|^2_{\mathcal{F}_\kappa}$. Now, if we substitute this result in the formulation of the CoRe loss in (3.3) yields

$$\begin{aligned}\mathcal{J}_e(\chi, U) =& \frac{1}{2}\sum_{i=1}^{I}\sum_{k=1}^{K}\delta_{ik}\|\Phi(x_k) - \Phi(c_i)\|^2_{\mathcal{F}_\kappa} + \\ &+ \frac{1}{2}\sum_{i=1}^{I}\sum_{k=1}^{K}(1-\delta_{ik})\left[RS_k^{(e|\chi|+k)}\left(1 - \frac{1}{2}\|\Phi(x_k) - \Phi(c_i)\|^2_{\mathcal{F}_\kappa}\right)\right]^2.\end{aligned} \tag{3.35}$$

Equation (3.35) states that CoRe minimizes the feature space distance between the prototype c_i and those x_k that are close in the kernel space induced by the activation functions φ_i, while it maximizes the feature space distance between the prototypes and those x_k that are far from c_i in the kernel space (see the example in Fig. 24).

3.5 Convergence Behavior of CoRe Clustering

3.5.1 Overview of the Proof

The intended aim of this section is to show that CoRe clustering is not only characterized by strong biological foundations, whereas it posses interesting properties in terms of asymptotic behavior. To achieve this, we base our analysis on the error formulation introduced in (3.3) as well as on the guidelines provided in (MW06) for studying the convergence behavior of *Distance Sensitive Rival Penalized Competitive Learning* (DSRPCL), that is a generalization of the RPCL model discussed in Section 3.2.2.

The general idea of the analysis in (MW06) is to show that a DSRPCL process decreases the cost given by the error function to a local minimum, ending up with a number of weight vectors eventually falling into a hypersphere surrounding the sample data, while the other prototypes diverge to infinity. To the extent of our knowledge, there is no full mathematical proof that an algorithm converges by identifying the *correct* number of clusters: in a sense, the existence of such proof would imply the existence of a computable way of determining the true cluster number or, in other words, that the bias-variance dilemma can be solved by applying the same algorithmic process that is at the basis of the proof. Rather, the analysis in (MW06) should be considered as a formal study of the general asymptotic properties of the DSRPCL algorithm.

In this section, we present a convergence analysis for CoRe clustering that founds on the DSRPCL approach, describing how CoRe satisfies the three correctness properties defined in (MW06), that are

Separation nature Given $\mathcal{G}$, a bounded hypersphere containing all the sample data, then after sufficient iterations of the algorithm, each prototype c_i will finally either fall into $\mathcal{G}$ or stay outside $\mathcal{G}$ and never enter back into $\mathcal{G}$.

Correct division The number of the weight vectors inside the hypersphere $\mathcal{G}$ should be equal to the number n_c of actual clusters in the sample data.

Correct location The n_c weight vectors will finally converge to, or locate around, the centroids of the n_c actual clusters.

The intuitive study proposed in (MW06) for DSRPCL is enforced with theoretical considerations showing that CoRe pursues a global optimality criterion for the vector quantization problem. In order to do this, we exploit the kernel interpretation of the CoRe loss given in Section 3.4 to generalize the results presented in (MPGRLRGB02) for Euclidean vector quantization models to kernel-based algorithms.

Starting from the error formulation in (3.3), we can derive the CoRe learning equations for epoch e by applying gradient descent to minimize the cost $\mathcal{J}_e(\chi, U)$ with respect to the parameters $\{c_i, \sigma_i\}$. Notice that, in the remainder of the chapter, we will consider the loss in (3.3) as being decomposed into a winner and a loser dependent term, that is

$$\mathcal{J}_e(\chi, U) = \mathcal{J}_e^{win}(\chi, U) + \mathcal{J}_e^{lose}(\chi, U) \tag{3.36}$$

where

$$\mathcal{J}_e^{win}(\chi, U) = \sum_{i=1}^{N}\sum_{k=1}^{K} \delta_{ik}(1 - \varphi_i(x_k)) \tag{3.37}$$

and

$$\mathcal{J}_e^{lose}(\chi, U) = \sum_{i=1}^{N}\sum_{k=1}^{K} (1 - \delta_{ik})(\varphi_i(x_k) RS_k)^2. \tag{3.38}$$

The prototype increment for the e-th epoch can be calculated as follows

$$\begin{aligned} \triangle c_i^e = \alpha_c \sum_{k=1}^{K} \Bigg[& \delta_{ik}\varphi_i(x_k)\frac{(x_k - c_i^e)}{(\sigma_i^e)^2} + \\ & -(1 - \delta_{ik}) \left(\frac{\varphi_i(x_k) RS_k^{(e|\chi|+k)}}{\sigma_i^e} \right)^2 (x_k - c_i^e) \Bigg] \end{aligned} \tag{3.39}$$

where α_c is a suitable learning rate ensuring that $\mathcal{J}_e$ decreases with e and δ_{ik} is the indicator function for the set win_k. Similarly, the spread update

can be calculated as

$$\triangle\sigma_i^e = \alpha_\sigma \sum_{k=1}^{K} \left[\delta_{ik}\varphi_i(x_k) \frac{\|x_k - c_i^e\|^2}{(\sigma_i^e)^3} + \right. \\ \left. -(1-\delta_{ik})(\varphi_i(x_k) R S_k^{(e|\chi|+k)})^2 \frac{\|x_k - c_i^e\|^2}{(\sigma_i^e)^3} \right]. \tag{3.40}$$

As one would expect, unit prototypes are attracted by similar patterns (first term in (3.39)) and are repelled by the dissimilar inputs (second term in (3.39)). The essence of the convergence analysis is to show that the repetition suppression penalization is capable of deflecting a number of prototype vectors, possibly only the excessive ones, from the data and that, after sufficient epochs, such repelled prototypes freeze in a position where the units cannot be winners for any sample. Moreover, we will show that the repulsive term does not impair the convergence of the prototypes that remains inside the data hypersphere $\mathcal{G}$; rather, its effect would be that of speeding up convergence by avoiding local minima of the vector quantization term in the error function (3.3).

3.5.2 Separation Nature

To prove the separation nature of the CoRe process we need to demonstrate that, given a a bounded hypersphere $\mathcal{G}$ containing all the sample data, then after sufficient iterations of the algorithm the cluster prototypes will finally either fall into $\mathcal{G}$ or remain outside it and never get into $\mathcal{G}$. In particular, those prototypes remaining outside the hypersphere will be driven far away from the samples by the RS repulsion.

We consider a prototype c_i to be *far away* from the data if, for a given epoch e, it is in the loser pool for every $x_k \in \chi$. To prove CoRe separation nature we first demonstrate the following Lemma.

Lemma 1 *When a prototype c_i is far away from the data at a given epoch e, then it will always be a loser for every $x_k \in \chi$ and will be driven away from the data samples.*

Proof The definition of *far away* implies that, given c_i^e, $\forall x_k \in \chi.\ i \in lose_k^e$, where the e in the superscript refers to the learning epoch. Given the

prototype update in (3.39), we obtain the weight vector increment Δc_i^e at epoch e as follows

$$\triangle c_i^e = -\alpha_\sigma \sum_{k=1}^{K} \left(\frac{\varphi_i(x_k) RS_k^{(e|\chi|+k)}}{\sigma_i^e} \right)^2 (x_k - c_i^e). \tag{3.41}$$

As a result of (3.41), the prototype c_i^{e+1} is driven further from the data. On the other hand, by definition of the winner set in (2.20), for each of the data samples there exists at least one winner unit for every epoch e, such that its prototype is moved towards the samples for which it has been a winner. Moreover, not every prototype can be deflected from the data, since this would make the term $\mathcal{J}_e^{win}(\chi, U)$ in (3.36) grow and, consequently, the whole $\mathcal{J}_e(\chi, U)$ will diverge since the loser error term $\mathcal{J}_e^{lose}(\chi, U)$ in (3.3) is lower bounded. However, this would contradict the gradient descent assumption that $\mathcal{J}_e(\chi, U)$ is non-increasing with respect to the epochs e. Therefore, there must exist at least one winning prototype c_l^e that remains close to the samples at epoch e. On the other hand, c_i^e is already far away from the samples and, by (3.41), c_i^{e+1} will be further from the data and won't be a winner for any $x_k \in \chi$. To prove this, consider the definition of win_k in (2.20): for c_i^{e+1} to be a winner, it must hold either

1. $\varphi_i(x_k) \geq \theta_{win}$,
2. or $i = \arg\max_{j \in U} \varphi_j(x_k, \lambda_j)$.

The former does not hold because the receptive field area where the firing strength of the i-th unit is above the threshold θ_{win} does not contain any sample at epoch e. Consequently, it cannot contain any sample at epoch $e+1$ since its center c_i^{e+1} has been deflected further from the data. The latter does not hold since there exist at least one prototype, i.e. c_l, that remains close to the data, generating higher activations than unit u_i. As a consequence, a far away prototype c_i will be deflected from the data until it reaches a stable point where the corresponding firing strength φ_i is negligible. ∎

Now, we can proceed to demonstrate the following

Theorem 1 *For a CoRe process there exist an hypersphere $\mathcal{G}$ surrounding the sample data χ such that, after sufficient iterations, each prototype c_i will finally either (i) fall into $\mathcal{G}$ or (ii) keep outside $\mathcal{G}$ and reach a stable point.*

Proof The CoRe process is a gradient descent (GD) algorithm on $\mathcal{J}_e(\chi, U)$, hence, for a sufficiently small learning step, the loss decreases with the number of epochs. Therefore, being $\mathcal{J}_e(\chi, U)$ always positive the GD process will converge to a minimum $\mathcal{J}^*$.

The sequences of prototype vectors $\{c_i^e\}$ will converge either to a point close to the samples or to a point of negligible activation far away from the data. If a unit u_i has a sufficiently long subsequence of prototypes $\{c_i^e\}$ diverging from the dataset then, at a certain time, it will no longer be a winner for any sample and, by Lemma 1, will converge at a point far away from the data. The attractors for the sequence $\{c_i^e\}$ of the *diverging* units lie at a certain distance r from the samples' mean, that is determined by those points $x \in X$ such that a Gaussian centered in x produces a negligible activation in response to any pattern $x_k \in \chi$ (see Fig. 25), that is

$$\exp\left(-\frac{\|x_k - x\|^2}{\sigma^2}\right) \approx 0 \; \forall x_k \in \chi.$$

Hence, $\mathcal{G}$ can be chosen as any hypersphere surrounding the samples with radius smaller than r.

On the other hand, since $\mathcal{J}_e(\chi, U)$ decreases to $\mathcal{J}^*$, there must exist at least one prototype that is not far away from the data, otherwise the $\mathcal{J}_e^{win}(\chi, U)$ term in (3.3) will diverge. In this case, the sequences $\{c_i^e\}$ must have accumulation points close to the samples. Therefore any hypersphere $\mathcal{G}$ enclosing all the samples will also surround the accumulation points of $\{c_i^e\}$ and, after a certain epoch E, the sequence will be always within such hypersphere. ∎

In summary, Theorem 1 tells that the separation nature holds for a CoRe process: some prototypes are possibly pushed away from the data until their contribution to the error in (3.3) becomes negligible. Far away prototypes will always be losers and will never head back to the data. Conversely, some prototypes will converge to the samples, heading to a saddle point of the loss $\mathcal{J}_e(\chi, U)$ by means of a gradient descent process.

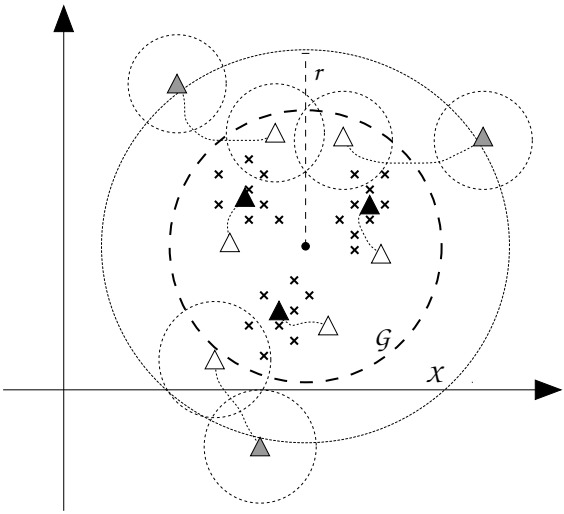


Figure 25: Separation property of the CoRe clustering process: a certain number of prototypes remains close to the data, while other weight vectors are driven far away from the samples. White triangles denote initial prototype positions, while black and grey triangles identify final positions of close and far away prototypes, respectively. Small dotted ellipses depict the edges of the neurons' receptive fields (for clarity, only the receptive fields of the far away units are shown). The far away prototypes have attractor points roughly at a distance r from the dataset mean, that is to the lieu of the space where the activation of CoRe units becomes negligible (i.e. ≈ 0), i.e. the boundary of the hypersphere X. Hence $\mathcal{G}$ can be chosen as any hypersphere surrounding the data and having radius smaller than r.

3.5.3 A Global Minimum Condition for Vector Quantization in Kernel Space

Before stepping over to study the correct division and location properties, we introduce a global optimality criterion for kernel vector quantization problems that will be used to prove the non-distortive behavior of the repetition suppression penalization.

The classical problem of *hard vector quantization* (VQ) in Euclidean space is to determine a codebook $V = v_1, \ldots, v_N$ minimizing the total distortion, calculated by Euclidean norms, resulting from the approxima-

tion of the inputs $x_k \in \chi$ by the code vectors v_i. Here, we focus on a more general problem that is vector quantization in kernel space. Given the nonlinear mapping Φ and the induced feature space norm $\|\cdot\|_{F_\kappa}$ introduced in the section 3.4.1, we aim at optimizing the distortion

$$\min D(\chi, \Phi_V) = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{N} \delta_{ik}^1 \|\Phi(x_k) - \Phi_{v_i}\|_{F_\kappa}^2 \tag{3.42}$$

where $\Phi_V = \{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$ represents the codebook in the kernel space and δ_{ik}^1 is equal to 1 if the i-th cluster is the closest to the k-th pattern in the feature space F_κ, and is 0 otherwise. It is widely known that VQ generates a Voronoi tessellation of the quantized space and that a *necessary condition* for the minimization of the distortion requires the codevectors to be selected as the centroids of the Voronoi regions (YZG92). In (MPGRLRGB02), it is given a *necessary and sufficient condition* for the global minimum of an Euclidean VQ distortion function. In the following, we generalize this result to vector quantization in feature space.

To prove the global minimum condition in kernel space we need to extend the results in (Spa80) (Proposition 3.1.7 and 3.2.4) to the general case of a kernel induced distance metric. Therefore we introduce the following lemma.

Lemma 2 *Let κ be a kernel and $\Phi : \chi \to F_\kappa$ a map into the corresponding feature space F_κ. Given a dataset $\chi = x_1, \ldots, x_K$ partitioned into N subsets C_i, define the feature space mean*

$$\Phi_\chi = \frac{1}{K} \sum_{k=1}^{K} \Phi(x_k) \tag{3.43}$$

and the i-th partition centroid

$$\Phi_{v_i} = \frac{1}{|C_i|} \sum_{k \in C_i} \Phi(x_k), \tag{3.44}$$

then we have

$$\sum_{k=1}^{K} \|\Phi(x_k) - \Phi_\chi\|_{F_\kappa}^2 = \sum_{i=1}^{N} \sum_{k \in C_i} \|\Phi(x_k) - \Phi_{v_i}\|_{F_\kappa}^2 + \sum_{i=1}^{N} |C_i| \|\Phi_{v_i} - \Phi_\chi\|_{F_\kappa}^2. \tag{3.45}$$

Proof Given a generic feature vector Φ_1, consider the identity $\Phi(x_k) - \Phi_1 = (\Phi(x_k) - \Phi_{v_i}) + (\Phi_{v_i} - \Phi_1)$: its squared norm in feature space is

$$\|\Phi(x_k)-\Phi_1\|^2_{F_\kappa} = \|\Phi(x_k)-\Phi_{v_i}\|^2_{F_\kappa}+\|\Phi_{v_i}-\Phi_1\|^2_{F_\kappa}+2(\Phi(x_k)-\Phi_{v_i})^T(\Phi_{v_i}-\Phi_1).$$

Summing over all the elements in the i-th partition we obtain

$$\begin{aligned}\sum_{k\in C_i}\|\Phi(x_k)-\Phi_1\|^2_{F_\kappa} &= \sum_{k\in C_i}\|\Phi(x_k)-\Phi_{v_i}\|^2_{F_\kappa} + \sum_{k\in C_i}\|\Phi_{v_i}-\Phi_1\|^2_{F_\kappa}+ \\ &\quad + 2\sum_{k\in C_i}(\Phi(x_k)-\Phi_{v_i})^T(\Phi_{v_i}-\Phi_1) \\ &= \sum_{k\in C_i}\|\Phi(x_k)-\Phi_{v_i}\|^2_{F_\kappa} + |C_i|\|\Phi_{v_i}-\Phi_1\|^2_{F_\kappa}.\end{aligned} \tag{3.46}$$

The last term in (3.46) vanishes since $\sum_{k\in C_i}(\Phi(x_k) - \Phi_{v_i}) = 0$ by definition of Φ_{v_i}. Now, applying the substitution $\Phi_1 = \Phi_\chi$ and summing up for all the N partitions yields

$$\begin{aligned}\sum_{k=1}^{K}\|\Phi(x_k)-\Phi_\chi\|^2_{F_\kappa} &= \sum_{i=1}^{N}\sum_{k\in C_i}\|\Phi(x_k)-\Phi_{v_i}\|^2_{F_\kappa}+ \\ &\quad + \sum_{i=1}^{N}|C_i|\|\Phi_{v_i}-\Phi_\chi\|^2_{F_\kappa}\end{aligned} \tag{3.47}$$

where the left side of equality holds since

$$\bigcup_{i=1}^{N} C_i = \chi \text{ and } \bigcap_{i=1}^{N} C_i = \emptyset. \quad \blacksquare$$

Using the results from Lemma 2 we can proceed with the formulation of the global minimum criterion by generalizing the results of Proposition 1 in (MPGRLRGB02) to vector quantization in feature space.

Proposition 1 *Let $\{\Phi_{v_1^g},\dots,\Phi_{v_N^g}\}$ be a global minimum solution to the problem in (4.18), then we have*

$$\sum_{i=1}^{N}|C_i^g|\|\Phi_{v_i^g}-\Phi_\chi\|^2_{F_\kappa} \geq \sum_{i=1}^{N}|C_i|\|\Phi_{v_i}-\Phi_\chi\|^2_{F_\kappa} \tag{3.48}$$

for any local optimal solution $\{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$ *to (4.18), where* $\{C_1^g, \ldots, C_N^g\}$ *and* $\{C_1, \ldots, C_N\}$ *are the* χ *partitions corresponding to the centroids*

$$\Phi_{v_i^g} = \frac{1}{|C_i^g|} \sum_{k \in C_i^g} \Phi(x_k)$$

and

$$\Phi_{v_i} = \frac{1}{|C_i|} \sum_{k \in C_i} \Phi(x_k)$$

respectively, and where Φ_χ *is the dataset mean in (3.43).*

Proof Since $\{\Phi_{v_1^g}, \ldots, \Phi_{v_N^g}\}$ is a global minimum for (4.18) we have

$$\sum_{i=1}^{N} \sum_{k \in C_i^g} \|\Phi(x_k) - \Phi_{v_i^g}\|_{F_\kappa}^2 \leq \sum_{i=1}^{N} \sum_{k \in C_i} \|\Phi_{v_i} - \Phi_\chi\|_{F_\kappa}^2 \tag{3.49}$$

for any local minimum $\{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$. From Lemma 2 we have that

$$\sum_{k=1}^{K} \|\Phi(x_k) - \Phi_\chi\|_{F_\kappa}^2 = \sum_{i=1}^{N} \sum_{k \in C_i^g} \|\Phi(x_k) - \Phi_{v_i^g}\|_{F_\kappa}^2 + \sum_{i=1}^{N} |C_i^g| \|\Phi_{v_i^g} - \Phi_\chi\|_{F_\kappa}^2 \tag{3.50}$$

and

$$\sum_{k=1}^{K} \|\Phi(x_k) - \Phi_\chi\|_{F_\kappa}^2 = \sum_{i=1}^{N} \sum_{k \in C_i} \|\Phi(x_k) - \Phi_{v_i}\|_{F_\kappa}^2 + \sum_{i=1}^{N} |C_i| \|\Phi_{v_i} - \Phi_\chi\|_{F_\kappa}^2. \tag{3.51}$$

Since (3.49) holds, we obtain

$$\sum_{i=1}^{N} |C_i^g| \|\Phi_{v_i^g} - \Phi_\chi\|_{F_\kappa}^2 \geq \sum_{i=1}^{N} |C_i| \|\Phi_{v_i} - \Phi_\chi\|_{F_\kappa}^2. \quad \blacksquare$$

3.5.4 Correct Division and Location

To evaluate the correct division and location properties we first analyze the case when the number of units N is equal to the true cluster number n_c.

Consider the CoRe loss decomposition in (3.36), where

$$\mathcal{J}_e^{win}(\chi, U) = \sum_{i=1}^{n_c} \sum_{k=1}^{K} \delta_{ik}(1 - \varphi_i(x_k))$$

must have, by definition, at least one minimum point. By applying the necessary condition $\partial \mathcal{J}_e^{win}(\chi, U)/\partial c_i = 0$, we can obtain a fixed point estimate of the prototypes as

$$c_i^e = \frac{\sum_{k=1}^{N} \delta_{ik} \varphi_i(x_k) x_k}{\sum_{k=1}^{N} \delta_{ik} \varphi_i(x_k)}. \tag{3.52}$$

When the number of prototypes equals the number of clusters, the fixed point iteration in (3.52) converges by positioning each unit weight vector close to the true cluster centroids. It can be shown that (3.52) approximates a local minimum solution of the kernel vector quantization problem in (4.18). To prove this, consider the CoRe loss formulation in kernel space (3.35): focusing only on the winner dependent term we have that

$$\mathcal{J}_e^{win}(\chi, U) = \frac{1}{2} \sum_{i=1}^{n_c} \sum_{k=1}^{K} \delta_{ik} \|\Phi(x_k) - \Phi(c_i)\|_{\mathcal{F}_\kappa}^2,$$

where c_i is estimated by (3.52).

Now, consider the kernel vector quantization problem in (4.18): a necessary condition for loss minimization requires the computation of the cluster centroids as

$$\Phi_{v_i} = \frac{1}{|C_i|} \sum_{k \in C_i} \Phi(x_k). \tag{3.53}$$

In general, computing analytically Φ_{v_i} requires to know the form and definition of the implicit nonlinear mapping Φ, so as to be able compute the sum in (3.53) and to solve the so-called *pre-image problem* (SMB$^+$99), that is determining the input space vector z such that

$$\Phi(z) = \Phi_{v_i}.$$

Unfortunately, such a problem is insolvable in the general case (SMB$^+$99). Rather then trying to compute the exact pre-image $\Phi(z)$ we can look for

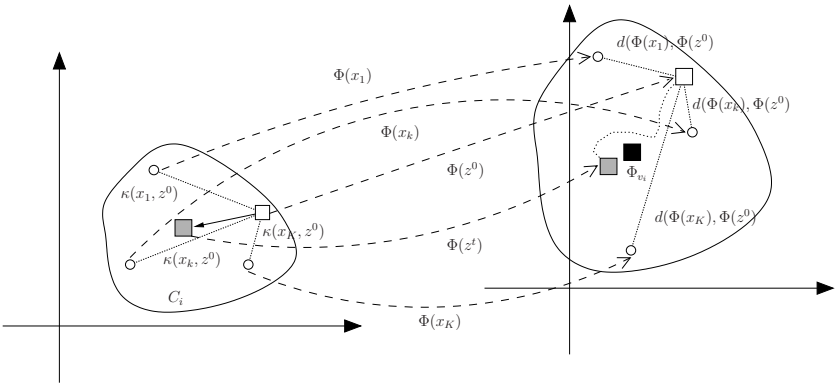


Figure 26: The actual centroid in feature space Φ_{v_i} is approximated by estimating the z^t vector in input space whose embedding $\Phi(z^t)$ minimizes the kernel induced distance from Φ_{v_i}. Again, feature space distances are computed by means of kernels without needing to explicitly compute the mapping Φ. The initial and final position of the prototype are depicted as white and gray squares, respectively. The black square identifies a plausible feature space centroid, while white circles stand for the data samples.

an approximation, by seeking the input space vector z that minimizes

$$\rho(z) = \|\Phi_{v_i} - \Phi(z)\|^2_{F_\kappa}, \tag{3.54}$$

that is the feature space distance between the centroid in kernel space Φ_{v_i} and the mapping $\Phi(z)$ of its approximated pre-image (see Fig. 26).

Instead of optimizing $\rho(z)$, it is convenient to minimize the distance between Φ_{v_i} and its orthogonal projection onto the span $\Phi(z)$ ((SMB+99)), that is

$$\left\| \frac{\langle \Phi_{v_i}, \Phi(z) \rangle}{\langle \Phi(z), \Phi(z) \rangle} \Phi(z) - \Phi_{v_i} \right\|^2_{F_\kappa} = \|\Phi_{v_i}\|^2_{F_\kappa} - \frac{\langle \Phi_{v_i}, \Phi(z) \rangle^2}{\langle \Phi(z), \Phi(z) \rangle}. \tag{3.55}$$

Since CoRe uses normalized kernels, it holds $\|\Phi_{v_i}\|^2_{F_\kappa} = 1$ and $\langle \Phi(z), \Phi(z) \rangle = 1$, hence the problem in (3.55) reduces to maximizing $\langle \Phi_{v_i}, \Phi(z) \rangle^2$. To determine the extremum we apply the gradient condition

$$\langle \Phi_{v_i}, \Phi(z) \rangle \frac{\partial \langle \Phi_{v_i}, \Phi(z) \rangle}{\partial z} = 0. \tag{3.56}$$

Substituting the centroid definition in (3.44) into the expression in (3.56) and applying the kernel trick, allows to write the gradient in terms of the kernel κ alone, obtaining the sufficient condition

$$0 = \frac{\partial \langle \Phi_{v_i}, \Phi(z) \rangle}{\partial z} = \frac{1}{|C_i|} \sum_{k \in C_i} \frac{\partial \kappa(x_k, z)}{\partial z} = \sum_{k \in C_i} \frac{\kappa'(x_k, z)}{|C_i|} (x_k - z) \quad (3.57)$$

where the last equality holds because κ is a Gaussian kernel. Solving (3.57) by fixed point iteration yields

$$z^e = \frac{\sum_{k \in C_i} \kappa(x_k, z^{e-1}) x_k}{\sum_{k \in C_i} \kappa(x_k, z^{e-1})} \quad (3.58)$$

that is the preimage estimate corresponding to the feature space centroid Φ_{v_i}. Clearly this is the same as the CoRe prototype estimate obtained in (3.52) for a Gaussian kernel centered in z^e.

Notice that the indicator function δ_{ik} in (3.19) is not null only for those points x_k for which unit u_i was in the winner set. This does not ensure the partition conditions over χ, since, by definition of win_k, some points can be associated with two or more winners. However, by (3.40) we know that the variance of the winners tends to reduce as learning proceeds. Therefore, using the same arguments exploited by Yair et al in their soft competition scheme (YZG92), it can be demonstrated that, after a certain epoch E, the CoRe competition will become a winner takes all process where δ_{ik} will be ensuring the partition conditions over χ.

Summarizing, the minimization of the CoRe winners error $\mathcal{J}_e^{win}(\chi, U)$ generates an approximated solution to the vector quantization problem in feature space in (4.18). As a consequence, the prototypes c_i can be interpreted as local solutions satisfying the conditions of the global minimum criterion in Proposition 1. Hence, by substituting the definition of Φ_χ in the results of Proposition 1 we obtain that $\{c_1, \ldots, c_{n_c}\}$ is an approximated global minimum for (4.18) if and only if

$$\sum_{i=1}^{n_c} \sum_{k=1}^{K} \frac{|C_i|}{K} \|\Phi(c_i) - \Phi(x_k)\|_{F_\kappa}^2 \geq \sum_{i=1}^{n_c} \sum_{k=1}^{K} \frac{|\tilde{C}_i|}{K} \|\Phi(\tilde{c}_i) - \Phi(x_k)\|_{F_\kappa}^2 \quad (3.59)$$

holds for every $\{\tilde{c}_1, \ldots, \tilde{c}_{n_c}\}$ that are approximated pre-images of a local minimum for (4.18).

In summary, a global optimum for (4.18) should minimize the feature space distance between the prototypes and the samples assigned to the respective clusters, while maximizing the weight vector distance from the sample mean, or, equivalently, the distance from all the data points in χ. The loser component $\mathcal{J}_e^{lose}(\chi, U)$ of the kernel CoRe loss in (3.35) depends on the term

$$\left(1 - \frac{1}{2}\|\Phi(c_i) - \Phi(x_k)\|^2_{F_\kappa}\right)$$

that maximizes the distance between the prototypes c_i and those x_k that do not fall in the respective Voronoi sets C_i. Hence, $\mathcal{J}_e^{lose}(\chi, U)$ produces a distortion in the estimate of c_i that pursues the global optimality criterion except for the fact that it discounts the repulsive effect of the $x_k \in C_i$. In fact, (3.59) suggests that c_i has to be repelled by all the $x_k \in \chi$. On the other hand, we know that c_i is a linear combination of the $x_k \in C_i$: applying the repulsive effect in (3.59) would subtract the contribution of such $x_k \in C_i$ resulting in either the cancelation of the attractive effect (which would be catastrophic) or in a simple scaling of the magnitude of the learning step, without affecting the orientation of the estimate. Hence, the CoRe loss makes a reasonable assumption discarding the repulsive effect of the $x_k \in C_i$ when calculating the estimate of c_i. Summarizing, CoRe locates the prototypes close to the centroids of the n_c clusters by means of (3.52), escaping from local minima of the loss function by approximating the global minimum condition of Proposition 1.

So far, we have proved that CoRe satisfies the correct location property whenever the number of units is equal to the actual cluster number. Now, we need to study the behavior of $\mathcal{J}_e(\chi, U)$ as the number of units N varies with respect to the actual cluster number n_c. This is the part of the analysis in (MW06) that relies less on formal arguments: rather, it tries to show that the loss formulation is well-founded and produces a rival penalization that counteracts the bias of the vector quantization part without introducing distortions in the prototype location. With respect to the CoRe loss, we have just given theoretically sound motivations that the repetition suppression penalization does not introduce such distortion whereas RS has a positive effect since it deflects prototypes from

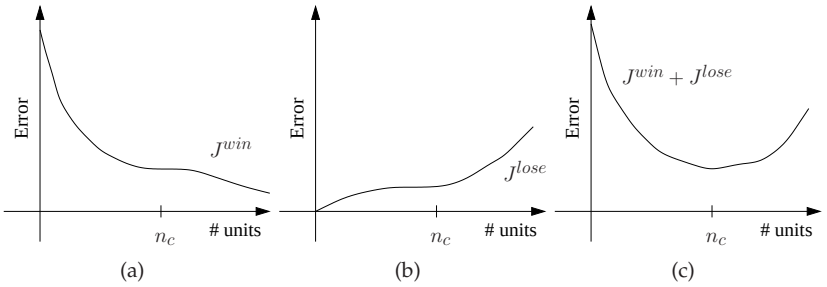


Figure 27: Sketches of the behavior for the CoRe loss: the winner dependent term (a) reduces as the size of the network becomes larger, hence producing overfitting. The loser dependent term (b) grows with the number of units falling into the data hypersphere $\mathcal{G}$. As a consequence, the total error (c) reaches a minimum when the number of identified clusters approximates the true cluster number n_c

local minima. Now, using the same motivations in (MW06) we show that the joint effect of winner and loser error terms produces a cluster number estimate that approximates the actual cluster number.

The winner dependent loss $\mathcal{J}_e^{win}$ essentially measures the quantization distortion when approximating the samples with the nearest prototype: clearly $\mathcal{J}_e^{win}$ tends to reduce as the the number of units increases (see the behavior in Fig. 27.a). At the same time, if the number of units falling into $\mathcal{G}$ increases, so will do the loser dependent loss $\mathcal{J}_e^{lose}$ (see Fig. 27.b). In particular, if the number of units falling into $\mathcal{G}$ is larger than n_c there will be a number of clusters that are erroneously split. Therefore, the samples from these clusters are likely to produce an increased level of error in $\mathcal{J}_e^{lose}$ contrasting the reduction of $\mathcal{J}_e^{win}$. On the other hand, $\mathcal{J}_e^{lose}$ tends to reduce when the number of units inside $\mathcal{G}$ is lower than n_c. This however produces increased levels of $\mathcal{J}_e^{win}$ since the prototype allocation won't match the underlying sample distribution. Hence, the full CoRe error $\mathcal{J}_e^{win} + \mathcal{J}_e^{lose}$ has its minimum when the number of units inside $\mathcal{G}$ approximates n_c (see Fig. 27.c), coherently with the correct division property.

3.6 Conclusion

We analyzed two important theoretical aspects of the CoRe clustering algorithm, that are robustness to noise and convergence. Founding on an alternative formulation for the CoRe clustering error, we pointed out its relationships with the theory of robust estimators, as well as the similarities with state-of-the art robust clustering algorithms such as SCM and RCA. The batch error formulation introduced in Section 3.2.1 emphasizes the role of CoRe as a robust version of RPCL, showing how CoRe strengthens the rival penalization mechanism by exploiting robust M-estimators to produce a learning process that is resistent to noise and outliers in the data.

Throughout the experiments, we showed the fundamental role played by robust estimation when addressing cluster number estimation from noisy data. This is a key issue when dealing, for instance, with image segmentation, where the number of homogenous areas in the image is not known a-priori and has to be estimated from the pictures' color-space distribution. The experimental results showed that CoRe can estimate the actual number of segmentation classes from MR images even if they are strongly corrupted by noise, while the non-robust learning process of RPCL can tolerate only low levels of data distortion: above such levels RPCL fails to identify the actual cluster number, resulting in a strong degradation of the task's performance. The robustness issues pointed out for RPCL and DSRPCL continue to hold also for the recent multi-agent divide-and-conquer framework (Xu07) based on the rival penalization mechanism: in particular, the loss function that has been defined for this model mixes robust Gaussian functions with non-robust Euclidean metrics, resulting in estimators that fail to meet the robustness conditions on the respective influence functions. For the sake of clarity, we point out that the experiments were not intended to address specific techniques for noise reduction in image segmentation. Rather, we pointed out the impact of robust estimation with respect to the broad cluster number identification problem; however, in general, nothing prevents to extend the CoRe loss formulation with application-dependent noise canceling terms,

e.g. penalty functions based on pixel continuity (ZC04a).

The simulation results in Section 3.3 suggest another peculiar property of the CoRe model, apart from robustness. The CoRe algorithm, in fact, tends to be extremely parsimonious when distributing prototypes in the denser areas of the input space. In particular, CoRe does not show the typical tendency of unsupervised vector quantization algorithms that allocate units to portions of the input space proportionally to the number of data points. As a consequence, CoRe can detect small-sized, low density clusters even in asymmetrical scenarios where there are clusters that contain a large portion of the samples in the data set. This information compression mechanism resembles much the behavior of repetition suppression, where more frequent patterns tend to be coded by fewer units sharply tuned to their preferred stimuli, and may serve as a correlate of novelty detection.

The second part of the chapter presented a sound analysis of the convergence behavior of CoRe clustering, showing how the minimization of the CoRe error satisfies the properties of separation nature, correct division and location (MW06). As the loss reduces to a minimum, the CoRe algorithm is shown to converge allocating a number of prototypes that approximates the actual cluster number. Moreover, it is given a sound optimality criterion that shows how the CoRe learning process pursues the global minimum of a vector quantization problem in kernel space. In particular, the global minimum theorem suggests that the repetition suppression penalization produces a local minima escape term that drives the prototypes to the global minimum of the kernel vector quantization loss. In other words, the repulsion produced by the repetition suppression mechanism does not introduce a distortion in the centroid identification process, whereas it offers a means for escaping the local optima of a vector quantization problem in the kernel space induced by the activation function. Such escape term allows CoRe to reach a clustering accuracy that is comparable to that of state-of-the-art kernel approaches and motivates why CoRe achieves, as detailed in Section 2.6.3, a faster convergence rate than RPCL although using a highly nonlinear loss formulation.

Besides the theoretical results regarding CoRe convergence, the global minimum condition described in Section 3.5.3 offers interesting practical implications with respect to the minimization of the vector quantization distortion. In the next chapter, we introduce a novel adaptive learning vector quantization algorithm that draws inspiration from Proposition 1 and we show how it addresses the sensitivity to initial condition and the local minima issues of traditional unsupervised vector quantization models.

Chapter 4

Expansive Competitive Learning for Kernel Vector Quantization

4.1 Introduction

The previous chapter has introduced a necessary and sufficient condition for a local minimum of a kernel-based distortion metric to be a globally optimal solution for a Vector Quantization (VQ) problem. In this chapter, we draw inspiration from the same criterion to describe a novel neural learning algorithm that addresses the local minima issue of unsupervised vector quantization models.

Vector quantization is a technique that is used to approximate a multi-dimensional input source with a finite number of representative vectors, i.e. the *codevectors*. An unsupervised learning vector quantizer constructs a finite collection of codevectors, i.e. the *codebook*, that minimizes the expected distortion introduced by approximating the inputs by means of the representative vectors. Learning a vector quantizer is a nonconvex optimization problem for which two necessary optimality conditions have been defined, that are the *centroid* and the *nearest neighbor* conditions. Based on these two conditions, (LBG80) proposed the Linde-Buzo-Gray

(LBG) algorithm that performs a local search to minimize the total vector quantization error. However, the two conditions alone ensure only convergence to a local minimum: this negatively affects the performance of the LBG algorithm, that has the tendency of getting stuck into local minima of the quantization error. Several models have been proposed to address this issue: for instance, competitive learning-based approaches such as Self Organizing Maps (SOM) (Koh82) and Neural Gas (MS91) (NG) tackle the problem by optimizing the codebook by means of a process of mutual competition between the codewords. Deterministic annealing (RGF92), on the other hand, eliminates the sensitivity to the initial configuration by smoothing the distortion measure to produce a convex estimate of the vector quantization error, controlled by a temperature parameter. The optimization process first identifies the global minimum at an high temperature, when the smoothed error is convex. Then, following the annealing scheme the temperature is lowered and, at each phase, the global minimum is tracked down until the temperatures reaches zero, corresponding to the original un-smoothed quantization error.

A large group of algorithms tackles the local minima problem by directly addressing the dependance of LBG from initial conditions. Enhanced LBG (PR01), for instance, defines a codeword utility measure that is used to move the least effective codevectors to positions of higher utility, while Adaptive Incremental LBG (SH06a) incrementally introduces or remove codevectors to minimize a local measure of the quantization error.

Munoz-Perez et al (MPGRLRGB02) presents an alternative approach to the local minima issue: instead of addressing the dependance on the starting conditions, they propose a sufficient condition for globally optimal solutions of vector quantization problems in Euclidean space (that is the basis of our kernel-based extension). Along with the theoretical result, (MPGRLRGB02) presents a competitive VQ algorithm, i.e. *Expansive Competitive Learning Vector Quantization* (ECLVQ), that exploits the minimum condition to define a *repulsive* codevector update term that favors the convergence to the global optimum of the loss function.

The idea of exploiting repulsive terms in stochastic optimization re-

sembles much RPCL and CoRe penalization schemes: however, while in the latter two models repulsion is used to estimate the size of the codebook, in ECLVQ it is used solely as a means for optimizing the choice of the codevectors. Other models based on this latter use of repulsion include, for instance, the *Electromagnetism-like Method* (cIBFS04), that uses an attraction-repulsion mechanism to seek the global optimum of a population-based optimization algorithm. The *lotto-type competitive learning* (LL00), on the other hand, splits neurons into two classes comprising multiple winner and loser units: unlike CoRe and RPCL, the repulsive effect exerted on the losers is not used to determine the codebook size, whereas it serves to favor the converge of the algorithm by avoiding dead neurons.

In this chapter, we generalize the ECLVQ algorithm to account for generic distance metrics in Hilbert spaces: based on this idea, we present an expansive competitive learning kernel vector quantization algorithm that pursues the global minimum condition in feature space introduced by Proposition 1 in Chapter 3. The remainder of the chapter is organized as follows: in Section 4.2 we briefly recall the formalization of the vector quantization problem in Euclidean space, discussing popular related work in the field, including the original ECLVQ algorithm. Section 4.3 formalizes the VQ problem in kernel space, introduced previously in Section 3.5.3, and presents the Expansive Competitive Kernel Learning Vector Quantization (ECKLVQ) algorithm. Finally, in Section 4.4, we test the performance of ECKLVQ on image quantization tasks.

4.2 Learning Vector Quantization

4.2.1 Definition

The classical problem of hard vector quantization in Euclidean space resides in determining a mapping $\mathcal{M} : \mathbb{R}^M \rightarrow V$ from the M-dimensional Euclidean space $\mathbb{R}^M$ to a set of N codevectors $V = v_1, \ldots, v_N$. The vector quantizer is chosen to minimize the average distortion measure

$$D(\mathbb{R}^M, V) = E[d(x, \mathcal{M}(x))]$$

resulting from the approximation of the input $x \in \mathbb{R}^M$ by the code vector $\mathcal{M}(x)$, where $d(\cdot)$ is usually chosen as the squared error distortion $d(x_1, x_2) = \|x_1 - x_2\|^2$. The mapping $\mathcal{M}$ determines a partition of the input space $\mathbb{R}^M$ into N disjoint *Voronoi regions* $C_1, \ldots, C_N$, such that

$$C_i = \{x \in \mathbb{R}^M : \mathcal{M}(x) = v_i\}. \tag{4.1}$$

For a finite input dataset $\chi = \{x_1, \ldots, x_k, \ldots, x_K\}$, the VQ problem reduces to the minimization of the sample distortion

$$\min D(\chi, V) = \frac{1}{K} \sum_{i=1}^{N} \sum_{k \in C_i} \|x_k - v_i\|^2 \tag{4.2}$$

where, for the sake of notational simplicity, we have used $k \in C_i$ as a short form for $x_k \in C_i$.

A vector quantizer in Euclidean space is optimal if no other quantizer has a smaller sample distortion (4.2): (YZG92) reports two necessary conditions for global minimization of the VQ error in (4.2), that are

Condition 1 Nearest Neighbor (or Voronoi) Condition: *For a given codebook $V = \{v_1, \ldots, v_N\}$, the optimal mapping $\mathcal{M}$ should be constructed in such a way that*

$$\mathcal{M}(x_k) = v_i \Leftrightarrow \|x_k - v_i\| \leq \|x_k - v_j\| \tag{4.3}$$

holds for every $v_j \in V$. Hence, the optimal partition $\{C_i\}$ results in the Voronoi condition

$$C_i = \{x_k \in \chi : \|x_k - v_i\| \leq \|x_k - v_j\| \forall j \neq i\}, \qquad i = 1, \ldots, N. \tag{4.4}$$

Condition 2 Generalized Centroid Condition: *For a given partition $\{C_i\}$, each codevector v_i is chosen as the centroid of the Voronoi region C_i. For a finite input set χ the centroid is calculated as the sample mean of the region, that is*

$$v_i = \frac{1}{|C_i|} \sum_{k \in C_i} x_k,$$

where $|C_i|$ is the cardinality of C_i.

Based on these two conditions, (LBG80) proposed a popular unsupervised vector quantization algorithm known as LBG. This algorithm starts

with randomly initialized centroids, iteratively computing the Voronoi regions in Condition 1 and updating the codevectors according to the formula in Condition 2: this process is repeated until the centroids' change becomes negligible.

The two conditions guarantee solely that the local minimum of (4.2) is pursued: consequently LBG leads to solutions that are locally optimal, but that might not be globally optimal. In (PR01), the authors study the performance of the LBG algorithm by analyzing how poorly chosen codeword positions leads to locally optimal solutions. They identify poorly positioned codevectors v_i by means of an utility function

$$U_i = \frac{D_i}{D_{av}}$$

that measures how much the distortion of the i-th Voronoi region C_i, that is

$$D_i = \sum_{k \in C_i} \|x_k - v_i\|^2,$$

departs from the average local distortion

$$D_{av} = \frac{1}{N} \sum_{i=1}^{N} D_i.$$

The ELBG algorithm (PR01) introduces a step in the LBG procedure that pursues an equalization of the codewords' utility, resulting in an equalization of the contribution D_i of the single Voronoi regions. In particular, ELBG identifies low utility codewords and moves them in the Voronoi region of those vectors that contribute more to the reduction of the quantization error, i.e. those with an high utility U_i. This optimization step has two effects on the Voronoi tessellation: on the one hand, it produces a join between a low utility cell and its adjacent regions; on the other hand, it splits an high utility Voronoi cell into two smaller partitions having, possibly, utilities that are closer to 1 than in the original cell.

Grounding on the idea of finding *good* initial codewords' positions for LBG, (LVV03) proposes the *global k-means* algorithm. This model incrementally adds codevectors, positioning them through a deterministic

global search procedure that requires to iteratively run the k-means algorithm, each time selecting a different data sample as the codevector initial position: the best solution out of those generated is retained and the process is iterated with an additional cluster. More in detail, to address vector quantization of K samples with codebook size N, global k-means proceeds as follows: it starts with one codevector and positions it at the optimal location, that is the sample mean. Then, it adds one more codebook and selects its optimal position by running K iteration of the k-means algorithm, each time initializing the new codevector as a different data point x_k. This process is iterated by keeping the solutions obtained for codebook size $n-1$ and positioning the n-th codevector at the best solution generated in the K k-means runs. Although this algorithm gives better results than LBG, it is very difficult to use in practice, even in its optimized versions, due to the high computational complexity.

Drawing on the ideas of both ELBG and global k-means, (SH06a) introduces the Adaptive Incremental LBG (AILBG). This algorithm incrementally inserts and deletes codewords in the codebook, pursuing direct equalization (i.e. without going through the definition of utility functions U_i) of the local distortion of the Voronoi regions. The insertion of new codevectors is done by means of a search procedure similar to that of global k-means, although AILBG restricts the search space to the Voronoi region of the current winner, thus reducing the computational effort with respect to the original model. Additionally, AILBG deletes codevectors with zero or minimal local distortion D_i, since they don't contribute significantly to the minimization of the quantization error in (4.2).

4.2.2 Expansive Competitive Learning Vector Quantization

The ELBG, global k-means and AILBG algorithms address the local optimality issue of LBG from the point of view of identifying *good* initial codevector positions. In order to do this, they resort to local search procedures that are often computational intensive. Recently, (MPGRLRGB02) have tackled the problem from a different perspective, proposing a vector quantization approach that founds on a global optimality term character-

ized by a low computational complexity. In particular, the two necessary conditions described in the previous section have been completed with a third necessary and sufficient condition for a local minimum codebook to be optimal, that is

Condition 3 Global Minimum Condition: *A local minimum codebook* $V^g = \{v_1^g, \ldots, v_N^g\}$ *for the problem in (4.2) is optimal if*

$$\sum_{i=1}^{N} |C_i^g| \|v_i^g - \overline{x}\|^2 \geq \sum_{i=1}^{N} |C_i| \|v_i - \overline{x}\|^2 \tag{4.5}$$

for any local minimum codebook $V = \{v_1, \ldots, v_N\}$, *where* $\overline{x} = \frac{\sum_{k=1}^{K} x_k}{|\chi|}$ *is the dataset mean.*

Basically, Condition 3 states that when the codevectors are a global minimum solution to the problem in (4.2) then they are the prototypes with larger variance. Based on this observation, (MPGRLRGB02) have developed a neural VQ algorithm, i.e. Expansive Competitive Learning Vector Quantization (ECLVQ), that tries to achieve global optimality by using a repulsive codevector update term that complies with the property in (4.5).

The ECLVQ network consist of a set of N units (neurons) u_i, each characterized by a prototype v_i and by a WTA activation function

$$o_i(x_k) = \begin{cases} 1 & \text{if} \quad i = \arg\max_j \{(x_k^T v_j) - \frac{1}{2}(v_j^T v_j)\} \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

The ECLVQ prototypes v_i are updated iteratively to generate a global minimum solution for the VQ problem in (4.2). The ECLVQ weight update rule is given by

$$\triangle v_i(t+1) = o_i(x_k) \left(\alpha(t) \cdot (x_k - v_i(t)) - (1 - \alpha(t)) \cdot \beta(t) \cdot \overline{x}\right) \tag{4.7}$$

where $\alpha(t)$, i.e. the *parameter of attraction*, is the learning rate controlling the effect of moving the prototype v_i of the winning unit u_i towards the current input pattern x_k, while $\beta(t)$, i.e. the *parameter of repulsion*, regulates the penalizing effect exerted by the dataset mean $\overline{x}$. Hence, the

weight update is the result of two forces: the first term in (4.7) attracts the prototypes towards the input patterns, while the second term deflects them from the center of gravity of the dataset χ, in accordance to the results of Condition 3. In (MPGRLRGB02) it is reported how the repulsive term enables the ECLVQ algorithm to avoid local minima, achieving consistently lower levels of VQ distortion than standard competitive learning vector quantization.

In the previous chapter (see Section 3.5.3), we have already shown how the results of Condition 3 can be extended to the general case of unsupervised vector quantization in a kernel-induced feature space. In the following, based on this generalized condition, we introduce the Expansive Competitive Kernel Learning Vector Quantization (ECKLVQ) algorithm, showing how it achieves superior vector quantization performance with respect to ECLVQ, LBG and its optimized version ELBG and AILBG.

4.3 Expansive Competitive Learning for Kernel Vector Quantization

4.3.1 Learning Vector Quantization in Kernel Space

The basics about kernel methods have already been discussed previously: in this chapter we will essentially make use of the same concepts introduced in Section 3.4.1. We recall that a kernel κ defines a feature space $\mathcal{F}_\kappa$, characterized by a metric $\|\cdot\|_{\mathcal{F}_\kappa}$ that can be calculated in terms of the inner product $\kappa(x_1, x_2) = \langle \Phi(x_1), \Phi(x_2) \rangle$, where the mapping between the input space and the feature space is realized by the implicit embedding function $\Phi(\cdot)$. For the sake of simplicity, as discussed in the previous chapter, we focus solely on normalized kernels.

By means of the the identity

$$d(\Phi(x_1), \Phi(x_2)) = \|\Phi(x_1) - \Phi(x_2)\|^2_{\mathcal{F}_\kappa} = 2 \cdot (1 - \kappa(x_1, x_2))$$

introduced in (3.34), we can straightforwardly adapt the Euclidean vector quantization formulation in Section 4.2.1 to general kernel induced distance metric. Given the non linear mapping Φ and the induced feature

space norm $\|\cdot\|_{\mathcal{F}_\kappa}$, the general kernel vector quantization (KVQ) problem aims at minimizing the distortion

$$\min D(\chi, \Phi_V) = \frac{1}{K}\sum_{i=1}^{N}\sum_{k\in C_i}\|\Phi(x_k) - \Phi_{v_i}\|^2_{\mathcal{F}_\kappa} \tag{4.8}$$

where $\Phi_V = \{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$ is the kernel space codebook and $\{C_i\}$ defines a Voronoi tessellation in the feature space. The necessary global minimum Conditions 1 and 2 can be adjusted to fit the kernel vector quantization (KVQ) problem as follows

Condition 4 Kernel Nearest Neighbor Condition: *For a given codebook $\Phi_V = \{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$, the optimal mapping $\mathcal{M}$ should be constructed in such a way that*

$$\mathcal{M}(\Phi(x_k)) = \Phi_{v_i} \Leftrightarrow \|\Phi(x_k) - \Phi_{v_i}\|_{\mathcal{F}_\kappa} \leq \|\Phi(x_k) - \Phi_{v_j}\|_{\mathcal{F}_\kappa} \tag{4.9}$$

holds for every $v_j \in \Phi_V$. Hence, the optimal partition $\{C_i\}$ results in the Voronoi condition

$$C_i = \{\Phi(x_k) \in \mathcal{F}_\kappa : \|\Phi(x_k) - \Phi_{v_i}\|_{\mathcal{F}_\kappa} \leq \|\Phi(x_k) - \Phi_{v_j}\|_{\mathcal{F}_\kappa} \forall j \neq i\} \tag{4.10}$$

for all $i = 1, \ldots, N$.

Condition 5 Generalized Kernel Centroid Condition: *For a given partition $\{C_i\}$, each codevector Φ_{v_i} is chosen as the centroid of the Voronoi region C_i. For a finite set $\mathcal{F}_\kappa$ the centroid is calculated as the sample mean of the region, that is*

$$\Phi_{v_i} = \frac{1}{|C_i|}\sum_{k\in C_i}\Phi(x_k). \tag{4.11}$$

As regards the Euclidean global minimum criterion in Condition 3, it has already been proved to hold for kernel induced distance metrics in Proposition 1 (see Section 3.5.3). Therefore, the sufficient global minimum condition in kernel space states

Condition 6 Kernel Global Minimum Condition: *A local minimum codebook $\{\Phi_{v_1^g}, \ldots, \Phi_{v_N^g}\}$ for the problem in (4.8) is optimal if*

$$\sum_{i=1}^{N}|C_i^g|\|\Phi_{v_i^g} - \Phi_\chi\|^2_{F_\kappa} \geq \sum_{i=1}^{N}|C_i|\|\Phi_{v_i} - \Phi_\chi\|^2_{F_\kappa} \tag{4.12}$$

for any local optimal solution $\{\Phi_{v_1}, \ldots, \Phi_{v_N}\}$, *where* $\Phi_\chi = \frac{1}{K}\sum_{k=1}^{K}\Phi(x_k)$ *is the data set mean.*

Several classical unsupervised learning vector quantization models have been adapted to kernels by exploiting different ways to solve the optimization of the dual, kernelized cost. In (SSM98), the authors proposed the first kernel extension for the k-means algorithm, expressing the codevectors as linear combinations of the samples projected in the embedding space. In other words, the approach in (SSM98) assumes that the codevectors lie in the span of the embedded dataset $\{\Phi_{x_1}, \ldots, \Phi_{x_K}\}$: thus, given the feature space centroid Φ_{v_i} it can be expanded as

$$\Phi_{v_i} = \sum_{k=1}^{K} \nu_{ik}\Phi(x_k) \tag{4.13}$$

where ν_{ik} is the coefficient determining the contribution of the embedded sample $\Phi(x_k)$ to the prototype Φ_{v_i}. The squared distance between the centroid Φ_{v_i} and a generic mapped sample $\Phi(x)$ can be written as

$$\begin{aligned}\|\Phi(x) - \Phi_{v_i}\|^2 = \|\Phi(x) - \sum_{k=1}^{K}\nu_{ik}\Phi(x_k)\|^2 = \kappa(x,x) + \\ - 2\sum_{k=1}^{K}\nu_{ik}\kappa(x,x_k) + \sum_{k,l=1}^{K}\nu_{il}\nu_{ik}\kappa(x_l,x_k)\end{aligned} \tag{4.14}$$

where we have used the definition in (4.13) and the kernel trick to express the scalar product in terms of the kernel κ. The identity in (4.14) can be used in the optimization of the kernel distortion (4.8) to obtain an incremental update rule for the coefficients ν_{ik}. In other words, instead of learning an explicit codebook (either in input or in feature space), the approach in (SSM98) learns a coefficient vector $\nu_{i\cdot}$ whose components determine the contribution of the single embedded patterns $\{\Phi_{x_1}, \ldots, \Phi_{x_K}\}$ to each *implicit* codevector Φ_{v_i}. The same representation has been used also in (HHB05) to introduce a kernel-based formulation for the fuzzy c-means algorithm, while kernel Self Organizing Maps have been proposed both as online (MF00) and batch (VR07) learning models.

In our model, we take an alternative approach to codebook representation: instead of generating implicit codevectors as combinations of the input samples projected in the embedding space, we explicitly estimate them by solving the approximated pre-image problem (see Section 4.3.2), hence obtaining *actual* prototypes in input space. Very similar in spirit to this approach is the supervised Neural Gas proposed by (HSV05): this algorithm extends standard NG to general distance metrics, i.e. including kernel-based measures, with application to supervised learning vector quantization.

4.3.2 The Expansive Competitive Learning for Kernel Vector Quantization Algorithm

Based on the kernel global minimum condition described in the previous section, we introduce the Expansive and Competitive Kernel Learning Vector Quantization (ECKLVQ) algorithm, that extends the formulation of ECLVQ (see Section 4.2.2) to feature space vector quantization. In particular, by exploiting the kernel trick and the identity in (3.34) we describe the ECKLVQ learning equations only in terms of the kernel κ, without the need of explicitly computing the nonlinear mapping Φ.

An ECKLVQ network consists of N competitive neurons u_i. At each time step, the units receive in input the current sample $x_k \in \chi$ and calculate their activation level by means of the following WTA rule

$$o_i(x_k) = \begin{cases} 1 & \text{if} \quad i = \arg\max_j \{\langle \Phi(x_k), \Phi_{v_j} \rangle - \frac{1}{2} \| \Phi_{v_j} \|^2_{\mathcal{F}_\kappa} \} \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

where Φ_{v_i} represents the feature-space prototype characterizing the i-th neuron. The condition in (4.15) can be simplified by considering that $\| \Phi_{v_j} \|^2_{\mathcal{F}_\kappa} = 1$ holds for every normalized kernel κ, thus obtaining

$$o_i(x_k) = \begin{cases} 1 & \text{if} \quad i = \arg\max_j \{\langle \Phi(x_k), \Phi_{v_j} \rangle\} \\ 0 & \text{otherwise} \end{cases} . \tag{4.16}$$

Equation (4.16) essentially states that the ECKLVQ neurons project the input sample x_k into the embedding space $\mathcal{F}_\kappa$ induced by the kernel κ.

The winning unit u_i is the neuron whose prototype is closest to the embedding of x_k in the feature space, since, from (3.34), we have that

$$\langle \Phi(x_k), \Phi_{v_j} \rangle = 1 - \frac{1}{2} \cdot \| \Phi(x_k) - \Phi_{v_j} \|^2_{\mathcal{F}_\kappa}.$$

In order to calculate $\langle \Phi(x_k), \Phi_{v_j} \rangle$ by means of the kernel κ we need to invert the mapping Φ so that we obtain the preimage v_j, such that $\Phi(v_j) = \Phi_{v_j}$. Finding an analytical expression for v_j requires to solve the *pre-image problem* (SMB$^+$99) which is, in practice, seldom possible. In fact, the explicit formulation of the mapping Φ is, normally, not known and the existence of the preimage v_j is not ensured in general. As shown in Section 3.5.4, rather than trying to find the exact preimage, we consider seeking an estimated solution $\tilde{v}_j$ such that $\Phi(\tilde{v}_j)$ closely approximates Φ_{v_j}. First, we derive the update rules for the estimated prototype $\tilde{v}_j \in \chi$, by applying gradient descent to minimize the distortion error in (4.8). Then, we will show that the solution to the problem in (4.8) is a suitable approximation for the preimage of Φ_{v_j}. For the sake of clarity, we point out that the discussion reported from now onwards, is restricted to differentiable, *isotropic* kernels, i.e. where $\kappa(x_1, x_2) = \kappa(\|x_1 - x_2\|^2)$.

The quality of the pre-image approximation strictly depends on the method used to estimate it: in this work we refer to the approach by (SMB$^+$99) that uses gradient descent and fixed point iteration. This nonlinear optimization method can, potentially, suffer from local minima and instability: for this reason, (KT04) proposed a non-iterative method involving only linear algebra that directly finds the location of the preimage based on distance constraints in the feature space. In (ARS07), on the other hand, the authors propose a pre-image estimation method based on Nyström extension, while (BWS03) suggest to use kernel principal component analysis and regression to learn pre-images. We employ the approach in (SMB$^+$99) since it allows a cleaner introduction of the ECKLVQ algorithm and, based on the simulation results, it has a competitive vector quantization performance: we plan, however, to study the effect of different pre-image estimation approaches in future works.

Consider the approximation $\Phi(\tilde{v}_j) \approx \Phi_{v_j}$: the ECKLVQ activation can

now be calculated as

$$o_i(x_k) = \begin{cases} 1 & \text{if} \quad i = \arg\max_j \{\kappa(x_k, \tilde{v}_j)\} \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

where it has been used $\kappa(x_k, \tilde{v}_j) = \langle \Phi(x_k), \Phi(\tilde{v}_j) \rangle$. Now, the winning unit u_i is the neuron whose prototype embedding $\Phi(\tilde{v}_j)$ is the closest to the mapping of x_k in the feature space. Similarly, applying the pre-image approximation to the distortion error in (4.8) yields to

$$\min D(\chi, \tilde{V}) = \frac{1}{K} \sum_{i=1}^{N} \sum_{k \in C_i} \|\Phi(x_k) - \Phi(\tilde{v}_i)\|^2_{\mathcal{F}_\kappa} \tag{4.18}$$

where $\tilde{V} = \{\tilde{v}_1, \ldots, \tilde{v}_N\}$ is the set of codevectors in input space. In order to solve the minimization problem in (4.18) we apply stochastic gradient descent to $D(\chi, \tilde{V})$; using the substitution in (3.34) we can replace the distance metric with the corresponding kernel, that is

$$\|\Phi(x_k) - \Phi(\tilde{v}_i)\|^2_{\mathcal{F}_\kappa} = 2 \cdot (1 - \kappa(x_k, \tilde{v}_i)). \tag{4.19}$$

The substitution in (4.19) yields to a reformulation of the problem in (4.18) that is now restated as

$$\max D(\chi, \tilde{V}) = \frac{1}{K} \sum_{i=1}^{N} \sum_{k \in C_i} \kappa(x_k, \tilde{v}_i) \tag{4.20}$$

The update rules for the codevectors $\tilde{v}_i$ are obtained by differentiating (4.20) with respect to the free parameters $\tilde{v}_i$. The exact form of the learning equations depends from the definition of the kernel κ: we recall that, in the following, we derive the ECKLVQ rules restricted to isotropic kernels.

Given the distortion function $D(\chi, \tilde{V})$ in (4.20) we obtain the codevector update for the i-th unit by applying gradient ascent in the $\triangle \tilde{v}_i$ direction, that is

$$\triangle \tilde{v}_i = \frac{\partial D(\chi, \tilde{V})}{\partial \tilde{v}_i} = \sum_{k \in C_i} \frac{\partial \kappa(x_k, \tilde{v}_i)}{\partial \tilde{v}_i} = \sum_{k \in C_i} \kappa'(x_k, \tilde{v}_i)(x_k - \tilde{v}_i) \tag{4.21}$$

where κ' is the first derivative of the kernel and $(x_k - \tilde{v}_i)$ is generated by the Euclidean metric dependence $\kappa(\|x_k - \tilde{v}_i\|^2)$. In order to obtain a competitive learning rule in the form of (4.7), we consider only the contribution in (4.21) that depends on the input x_k. Hence, the online (competitive) update becomes

$$\triangle \tilde{v}_i(t+1) = o_i(x_k)\alpha(t)\kappa'(x_k, \tilde{v}_i(t))(x_k - \tilde{v}_i(t)) \tag{4.22}$$

where $\alpha(t)$ is the time-dependent learning rate such that $\alpha(t) \to 0$ as $t \to \infty$, in accordance with the typical convergence conditions of WTA learning (YZG92).

The stochastic gradient ascent described so far seeks a codebook $V = \{\tilde{v}_1, \ldots, \tilde{v}_N\}$ of approximated preimages $\tilde{v}_i$, that minimizes the distortion in (4.18). In the following, we show that the preimages obtained by the process in (4.21) and (4.22) are close estimates of the actual preimages $v_i = \Phi^{-1}(\Phi_{v_i})$. Consider, for instance, the general problem of seeking an approximated preimage $z \in \chi$ such that z minimizes the feature space distance $\|\phi - \Phi(z)\|^2_{\mathcal{F}_\kappa}$ with a given embedding $\phi \in \mathcal{F}_\kappa$. As it has been shown in Section 3.5.4, rather than minimizing $\|\phi - \Phi(z)\|^2_{\mathcal{F}_\kappa}$, it is convenient to minimize the distance between ϕ and its orthogonal projection onto the span $\Phi(z)$ (SMB+99), that is

$$\left\| \frac{\langle \phi, \Phi(z) \rangle}{\langle \Phi(z), \Phi(z) \rangle} \Phi(z) - \phi \right\|^2_{\mathcal{F}_\kappa} = \|\phi\|^2_{\mathcal{F}_\kappa} - \frac{\langle \phi, \Phi(z) \rangle^2}{\langle \Phi(z), \Phi(z) \rangle}. \tag{4.23}$$

Given κ as a normalized kernel, it holds $\|\phi\|^2_{\mathcal{F}_\kappa} = 1$ and $\langle \Phi(z), \Phi(z) \rangle = 1$, hence the problem in (4.23) reduces to maximizing $\langle \phi, \Phi(z) \rangle^2$. Now, if we chose ϕ to be the feature space centroid Φ_{v_i} then the maximization of $\langle \Phi_{v_i}, \Phi(z) \rangle^2$ produces the preimage z that approximates most closely Φ_{v_i}. To determine the extremum we apply the usual gradient condition

$$\langle \Phi_{v_i}, \Phi(z) \rangle \frac{\partial \langle \Phi_{v_i}, \Phi(z) \rangle}{\partial z} = 0. \tag{4.24}$$

Using the centroid definition in (4.11) allows to evaluate the gradient in (4.24) in terms of the kernel, obtaining the sufficient condition

$$0 = \frac{\partial \langle \Phi_{v_i}, \Phi(z) \rangle}{\partial z} = \frac{1}{|C_i|} \sum_{k \in C_i} \frac{\partial \kappa(x_k, z)}{\partial z} = \sum_{k \in C_i} \frac{\kappa'(x_k, z)}{|C_i|}(x_k - z) \tag{4.25}$$

where the last equality holds since we are using Euclidean metric based kernels. Using fixed point iteration to solve (4.25) yields to

$$z = \frac{\sum_{k \in C_i} \kappa'(x_k, z) x_k}{\sum_{k \in C_i} \kappa'(x_k, z)} \tag{4.26}$$

that is the preimage estimate corresponding to the feature space centroid Φ_{v_i}. Clearly, this corresponds to the estimate obtained from the distortion minimization condition in (4.21). In addition, the online competitive rule in (4.22) is known to converge to the solution of (4.21) for an appropriate choice of the learning rate α (SMB$^+$99). Hence, the solution $\tilde{v}_i$ generated by the distortion minimization process is a suitable preimage approximate for the feature space centroid Φ_{v_i}.

The kernel-based competitive learning rule derived so far does not yet exploit the global minimum criterion in Condition 6. Comparing the expression in (4.22) with the Euclidean competitive rule in (4.7) it follows that we need to extend the formulation of the ECKLVQ rule by adding a repulsive term that deflects the input-space prototype $\tilde{v}_i$ from the preimage of the feature space mean Φ_χ, defined in (4.11). Again, this requires to solve an approximated preimage problem, where we seek for a $z_\chi \in \chi$ such that $\|\phi_\chi - \phi(z_\chi)\|^2_{\mathcal{F}_\kappa}$ is minimum. If we apply the optimization procedure described above, we can derive the approximated preimage of ϕ_χ by fixed point iteration as

$$z_\chi^{t+1} = \frac{\sum_{k=1}^{K} \kappa'(x_k, z_\chi^t) x_k}{\sum_{k=1}^{K} \kappa'(x_k, z_\chi^t)} \tag{4.27}$$

where the estimation process stops when $\|z_\chi^{t+1} - z_\chi^t\| \leq \epsilon$.

The ECKLVQ competitive learning rule can, finally, be written as

$$\begin{aligned} \triangle \tilde{v}_i(t+1) \quad = \quad & o_i(x_k) \cdot (\alpha(t) \cdot \kappa'(x_k, \tilde{v}_i(t)) \cdot (x_k - \tilde{v}_i(t)) + \\ & -(1 - \alpha(t)) \cdot \beta(t) \cdot \kappa(\tilde{v}_i(t), \overline{z}_\chi) \cdot \overline{z}_\chi) \end{aligned} \tag{4.28}$$

where $\overline{z}_\chi$ is the approximated preimage of Φ_χ resulting from (4.27). Again, $\alpha(t)$ is the parameter regulating the attraction towards the current input pattern x_k, while $\beta(t)$ is the parameter controlling the penalizing effect exerted by the kernel-weighted estimate of the dataset mean. The repulsive

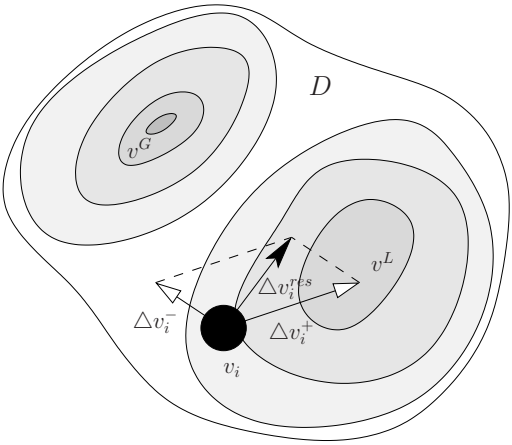


Figure 28: The codeword update $\triangle v_i$ is the result of a positive component v_i^+ along the direction of the minimization of the distortion D and a repulsive component v_i^- that has a distortive effect with respect to the original local minima direction.

effect is regulated by the kernel weighting term $\kappa(\tilde{v}_i, \overline{z}_\chi)$, that determines the strength of the repulsion based on the feature space distance between the mean estimate $\overline{z}_\chi$ and the prototype $\tilde{v}_i$. This choice is consistent with the results Condition 6, since the repulsive effect tends to vanish as a prototype maximizes its feature space distance from the estimated mean.

The attraction-repulsion dynamics is partly regulated by the choice of the learning rates α and β: we allow α to be larger than β and to gradually decrease to zero slower than the repulsion parameter. In particular, the attraction parameter follows a linear decay rule $\alpha(t) = \alpha_0(1 - t/T)$, where $\alpha_0 \in [0, 1]$ is the initial learning rate and T is the maximum number of allowed learning epochs. Similarly, the repulsion parameter decay follows

$$\beta(t) = \begin{cases} \beta_0(1 - t/T_0) & \text{if} \quad t \leq T_0 \\ 0 & \text{if} \quad t > T_0 \end{cases}$$

where β_0 is the initial repulsion rate and $T_0 < T$ is the repulsion cancelation epoch.

Figure 28 exemplifies the attractive $\triangle v_i^+$ and repulsive $\triangle v_i^-$ updates acting on a winner codevector v_i. In particular, Fig. 28 represent a scenario where the quantization error surface D comprises a local and a

global minimum (depicted as darker areas) denoted by v^L and v^G, respectively. The effect of the positive update $\triangle v_i^+$ is to move the prototype on the steepest descent direction with respect to D: as a result of this local minimization step, the prototype v_i might converge to the local minimum v^L. On the other hand, the negative reinforcement $\triangle v_i^-$ has a distortive effect on the codevector update, that prevents v_i from being trapped in the local minimum. In a sense, the effect of the repulsive term plays the role of classical neural network strategies for avoiding local minima in gradient descent such as *momentum* and *simulated annealing*. A remark that might rise consequently to this observation, is whether performance gain has to be ascribed fully to the effect of the global minimum condition or if it can be obtained as well with a generic repulsive term pointing towards a random direction (i.e. similarly to a simulated annealing strategy). In the experimental part, we will show that the quantization error reduction of ECKLVQ is fully due to the global minimum condition, illustrating how a random repulsive term does not produce significant performance improvements with respect to a purely attractive vector quantizer.

Algorithm 2 summarizes the steps of the ECKLVQ process used in the experimental evaluation in Section 4.4. In the following, we will focus on ECKLVQ networks characterized by Gaussian kernels $\kappa(x, \tilde{v}_i) = \exp\left(-0.5\|x - \tilde{v}_i\|^2/\sigma^2\right)$, where the codevector $\tilde{v}_i$ is the mean of the i-th Gaussian function having spread σ^2. This particular kernel induces an infinitely dimensional feature space and has nice properties of differentiability, normalization and robustness to noise (STC04). We remark that the general scheme described so far can be straightforwardly applied to any differentiable kernel function and that can be used as an heuristic for developing competitive learning algorithms in feature space even for non differentiable kernels.

4.4 Experimental Evaluation

The effectiveness of the ECKLVQ approach is evaluated in image vector quantization, analyzing the effect of the introduction of the global optimality term and showing how a kernel-based approach has signifi-

Algorithm 2 ECKLVQ

Given: the dataset $\chi = \{x_1, \ldots, x_K\}$*, the maximum number of learning epochs* T*, the repulsion cancelation epoch* T_0*, the initial learning rates* α_0 *and* β_0*, the minimum estimate change* ϵ *and the network size* N*;*

$t = 0$;
repeat
  $t = t + 1$;
  Compute the sample mean estimate $\overline{z}_\chi^t$ using (4.27);
until $\|z_\chi^{t+1} - z_\chi^t\| \leq \epsilon$
$t = 0$;
$K = |\chi|$;
Initialize $\alpha(0) = \alpha_0$ and $\beta(0) = \beta_0$;
Randomly initialize the N prototypes v_i^0;
repeat
  Generate a random presentation order for the patterns in χ;
  for all $x_k \in \chi$ **do**
    $t = t + 1$;
    for $i = 1$ to N **do**
      Calculate unit activation $o_i^t(x_k)$ using (4.17);
    end for
    Calculate the winner unit u_w s.t. $w = \arg\max_j \{\kappa(x_k, v_j^t)\}$;
    Update the winner codevector v_w^t using (4.28);
    $\alpha(t) = \alpha_0(1 - t/(T \cdot K))$;
    $\beta(t) = \begin{cases} \beta_0(1 - t/(T_0 * K)) & \text{if} \quad t \leq (T_0 * K) \\ 0 & \text{if} \quad t > (T_0 * K) \end{cases}$;
  end for
until $t > T * K$

(a) Goldhill (b) Lena (c) Lake

Figure 29: Test images for the vector quantization task.

cant advantages with respect to LVQ in Euclidean space. All the trials reported hereafter have been performed using ECKLVQ with Gaussian kernels having the same spread σ^2 (initialized as the data variance), initial attraction parameter $\alpha_0 = 0.5$ and initial repulsion parameter $\beta_0 = 0.2$. Moreover, the repulsion cancelation epoch T_0 is chosen to be slightly less than a half of the total epoch number T, i.e. $T_0 = 0.4 \cdot T$.

First, to test the efficacy of the global minimum term, we considered the quantization of three 513×513 graylevel images (see Fig.29), that are *Goldhill*, *Lena* (both available in the BragZone repository[1]) and *Lake* (downloaded from USC-SIPI database[2]). Three LVQ algorithms have been applied to the quantization of the 3×3 subimages extracted from the original pictures: the popular LBG algorithm has been used to determine the baseline performance on the task, and ECKLVQ is compared with a purely attractive Kernel LVQ (KLVQ) model obtained from (4.28) with $\beta = 0$, that is without the effect of the global optimality term. Table 6 reports the Mean Squared Error (MSE) for the experiments performed with 32 and 64 codevectors: each trial has been repeated 10 times and averaged (standard deviations are reported in brackets).

The results in Table 6 show that the addition of the global minimization term generates a significant advantage in terms of reduction of the quantization error. In particular, the standard kernel quantizer KLVQ

[1] `http://links.uwaterloo.ca/bragzone.base.html`
[2] `http://sipi.usc.edu/database/index.html`

Table 6: Mean Squared Error for LVQ on the Goldhill, Lena and Lake images.

	32 Codevectors			64 Codevectors		
	LBG	KLVQ	ECKLVQ	LBG	KLVQ	ECKLVQ
Goldhill	27.4 (.7)	34.8 (0)	22.4 (.2)	31.6 (0)	31.5 (0)	31.5 (0)
Lena	22.5 (.5)	22.2 (.5)	18.9 (.1)	20.1 (.4)	20.3 (.4)	16.5 (.1)
Lake	30.7 (.9)	29.7 (1)	26.4 (.2)	27.4 (.5)	27.4 (.6)	23.2 (.1)

achieves distortion errors that are close, or even higher than those obtained by LBG, while ECKLVQ reaches significantly lower levels of quantization error. Hence, the kernel itself (in KLVQ) does not produce a performance gain, that is due, instead, to the presence of the ECKLVQ repulsive term. It appears that both LBG and KLVQ solutions are trapped in local mininima of the respective objective functions. The values of the standard deviation (reported in brackets in Table 6) seems to confirm this behavior: ECKLVQ has a stabler learning dynamics with smaller oscillations around the minimum, while both LBG and KLVQ show consistent fluctuations. The Goldhill image (see Fig. 30.a) appears to be especially difficult to quantize due to the higher number of details in the scene that produce a more complex error quantization surface. The performance of the KLVQ algorithm seems to be particularly affected, resulting in a local minimum solution with a significantly higher MSE when 32 codebooks are used. On the other hand, the addition of the repulsive term in ECKLVQ allows to avoid such local minima, achieving an image reconstruction quality that is higher than that of the LBG algorithm (compare the two reconstructed images in Fig. 30.b and 30.c).

To quantitatively assess the quality of the reconstruction produced by ECKLVQ, we use the image quality index (QI) proposed by (ZB02). This index models the distortion between the original and the reconstructed image as a combination of three factors, that are loss of correlation, luminance distortion and contrast distortion. The final quality index is the result of the product of three components, each measuring one of the factors described previously. The three statistical features, i.e. linear correlation, mean luminance proximity and contrast similarity, are not computed globally; rather, they are measured locally using a sliding window

approach and then combined together. The superior reconstruction result of ECKVLQ on the *Goldhill* image is confirmed numerically by the QI measured using the code provided by the authors (ZB02)[3]: for instance, LBG achieves a $QI = 0.51$, while ECKLVQ obtains $QI = 0.66$, where higher QI values imply a better reconstruction quality.

As we have seen, the use of a kernel does not produce, per-se, a significant performance gain. However, a kernel quantization algorithm with a proper global minimization term can produce consistent advantages with respect to standard Euclidean VQ. In particular, we expect the ECKLVQ algorithm to be more effective than its Euclidean counterpart ECLVQ when dealing with data with noise and outliers. In the second set of simulations, we tested the effectiveness of the kernel generalization in ECKLVQ with respect to the original Euclidean ECLVQ algorithm on the 513×513 *Elaine* image[4] perturbed by 20% additive Gaussian noise and by 5% salt and pepper noise (see Fig. 32.a). Both ECLVQ and ECKLVQ have been trained on the noisy data for 30 learning epochs: Fig. 31 shows that ECKLVQ achieves a consistently lower mean squared error than ECLVQ already on the training data. As a second step, the learned codevectors have been used to reconstruct the original *Elaine* image: Fig. 32.c and 32.d shows the images produced by the two algorithms. Again, ECKLVQ achieved a lower quantization error ($MSE = 2.43$) then the Euclidean ECLVQ ($MSE = 5.35$). A fast visual inspection of the two images shows that the quality of the reconstructed ECKLVQ picture is higher, since the robust estimation of the Gaussian kernel produced a smother image with less aliasing on the skin and hat areas (compared with the ECLVQ image in Fig. 32.d).

The visual impression is confirmed by the QI measures: the ECLVQ algorithm achieves an image quality of 0.72, while ECKLVQ obtained a QI equal to 0.91. Figure 33 shows the quality map for the two reconstructed pictures: white pixels identify areas where the QI is higher, while the darker regions identify the corrupted portions of the images (i.e. with low QI). As expected, the kernel-based quantizer has a superior recon-

[3] `http://www.cns.nyu.edu/~zwang/`

[4] USC-SIPI: `http://sipi.usc.edu/database/index.html`

(a)

(b)

(c)

Figure 30: The 513×513 Goldhill image (a) reconstructed by LBG (b) and ECKLVQ (c).

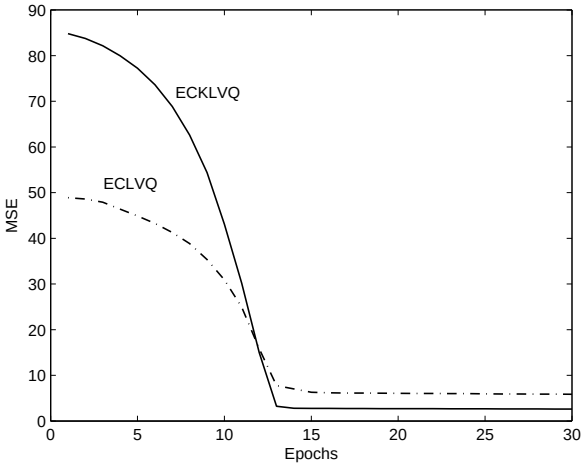


Figure 31: ECLVQ and Gaussian ECKLVQ mean squared error (MSE) during training on the noisy Elaine image.

struction performance when dealing with noisy images and outlier data. However, this is not the unique advantage of ECKLVQ with respect to the original ECLVQ: in order to validate the robustness of the two models, we run several simulations on the quantization of the 3×3 blocks of the original Elaine image, starting with different initial learning rate values. Figure 34 depicts a significant sample of the behavior of the two algorithms for different α_0 and β_0, i.e. the attraction and repulsion parameters, respectively. The results in Fig. 34 show that the choice of the initial learning rates does not influence substantially the convergence of the ECKLVQ algorithm, while the final quantization error of ECLVQ seems to be strongly influenced by the initialization of the meta-parameters. As a consequence, the ECLVQ learning rates have to be heuristically determined by trial and error[5], while ECKLVQ shows to be robust also with respect to the choice of the meta-parameters.

[5]Note that all the simulation results presented in this section have been obtained based on the most advantageous setup for ECLVQ, i.e. using the initial learning rates $\alpha_0 = 0.5$ and $\beta_0 = 0.2$.

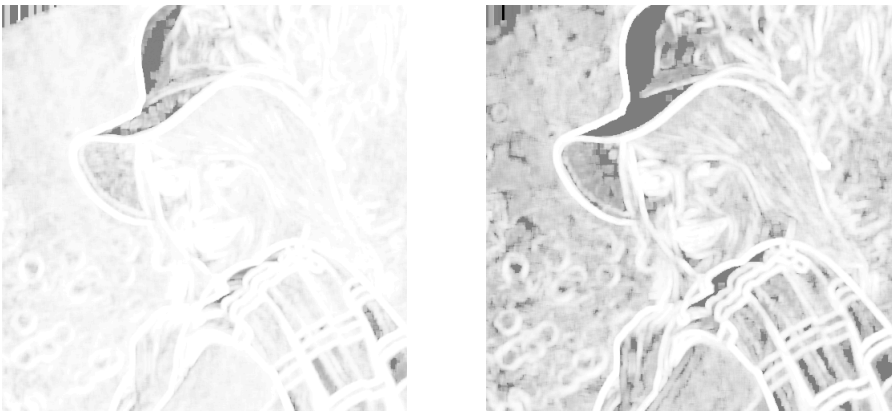
(a)

(b)

(c)

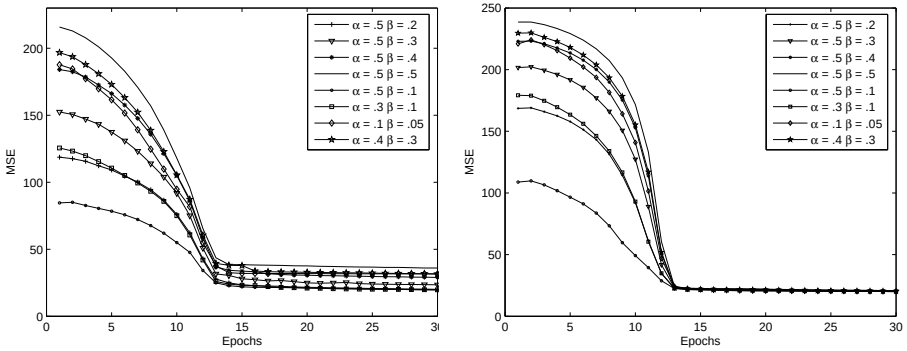
(d)

Figure 32: Application of ECLVQ and ECKLVQ to the quantization of the Elaine image corrupted by 20% additive Gaussian noise plus 5% salt and pepper noise: (a) the corrupted image used in the learning phase; (b) the original Elaine image; (c) the ECKLVQ reconstructed image and (d) the ECLVQ reconstructed image (best viewed in electronic format).



(a) (b)

Figure 33: Quality map for the Elaine image reconstructed by ECKLVQ (a) ($Qi = 0.91$) and ECLVQ (b) ($QI = 0.72$).



(a) (b)

Figure 34: Mean squared error on the original Elaine image quantization with varying learning rates initialization: (a) ECLVQ and (b) ECKLVQ MSE trajectories.

(a) (b)

Figure 35: (a) Original Lena image and (b) ECKLVQ reconstruction (256 codewords and $PSNR = 32.21$).

The final set of simulations compare the gain produced by the ECKLVQ global optimality term with respect to the enhanced versions of the LBG algorithm presented in Section 4.2. In particular, the quantization quality of the 512×512 *Lena* image (see Fig. 35.a) is evaluated in terms of the Peak Signal to Noise Ratio (PSNR), that is a measure used in image compression to evaluate the reconstructed images after the encoding-decoding cycle. The PSNR is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{N} \sum_{i=1}^{N} (o(i) - r(i))^2} \tag{4.29}$$

where $o(i)$ identifies the graylevel of the i-th pixel of the original image, while $r(i)$ refers to the reconstructed image. Following the testbed described in (PR01; SH06a), we extract 4×4 sub-images from the original Lena, and perform VQ with 256, 512 and 1024 codevectors. As a first step, we compare the performance of ECKLVQ with respect to the original ECLVQ, the purely attractive kernel-based KLVQ and the baseline LBG. Additionally, following the analysis in Section 4.3, we study the per-

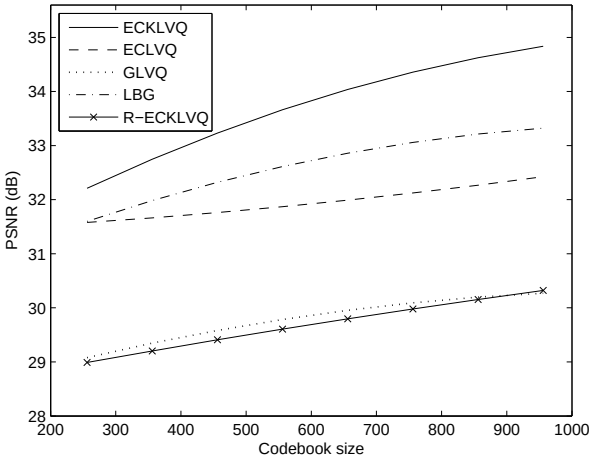


Figure 36: Comparison of PSNR for different vector quantizers on the 512×512 Lena image.

formance of a modified ECKLVQ algorithm, named R-ECKLVQ, that uses randomly generated repulsion directions in place of the dataset mean in the global optimality term. This is intended to show whether the performance increase is truly generated by the global minimum condition or if it stems from a sort of annealing process that deflects the codewords from local minima irrespectively of the chosen repulsion direction.

Figure 36 shows the comparison of the results obtained by the five algorithms: ECKLVQ obtains the best reconstruction quality (i.e. higher PSNR ratios). Again it is clear how such performance gain is not due to the kernel alone, but is produced by the global optimality term. The purely attractive KLVQ, in fact, obtains the worse result together with R-ECLVQ: in particular, it seems clear how a random repulsion direction does not produce any performance increase with respect to the results of the KLVQ, thus confirming the global optimality hypothesis. Figure 36 shows that ECKLVQ has a consistently superior reconstruction ability with respect to the original ECLVQ even when the data is not corrupted by noise and outliers (see Fig. 35.b for an example of Lena reconstructed

Table 7: Comparison of VQ performance on the 512×512 Lena image for different codeword sizes.

Codewords	PSNR (dB)			
	LBG	ELBG	AILBG	ECKLVQ
256	31.60	31.94	32.01	32.21
512	32.49	33.14	33.22	33.48
1024	33.37	34.42	34.59	34.95

by ECKLVQ). Notice that, as the number of codewords grows, ECLVQ obtains increasingly worse results than LBG. In our opinion, this is due to the fact that LBG is a batch algorithm, i.e. the codevector update is performed only at the end of the learning epoch, while ECLVQ is an online algorithm and is, thus, more exposed than LBG to the local minima problem. This characteristics produces disruptive learning interference as the codebook size increases and the quantization error surface becomes more complex.

Table 7 shows a comparison of ECKLVQ with the original LBG and two optimized algorithms, that are EILBG and AILBG. The proposed algorithm achieves again the best reconstruction quality in terms on PSNR on all the codebook sizes used in the test. In particular, ECKLVQ obtains a significative performance improvement with respect to AILBG and EILBG without resorting to time consuming local search operations (see Section 4.2.2), by using a simple constant-time optimality term.

4.5 Conclusion

To prove the correctness of the cluster identification process in the CoRe algorithm, we introduced, in Chapter 3, a global minimum criterion for learning vector quantization problems in general kernel-induced spaces. Kernels are known to produce implicit mappings of the data onto high dimensional spaces where linear operations result in nonlinear operations in the input space. Such an high dimensional mapping generates complex Voronoi surfaces that quantize the input space, allowing to discover nonlinear data structures that would not show in the Euclidean space.

Founding on these theoretical considerations, in this chapter we derived an online competitive learning algorithm for a vector quantizer based on Gaussian kernels. The simulation results proved the effectiveness of the global minimum term, that allows the codevectors to escape from the local minima of the quantization error and to achieve superior performance with respect to optimized LVQ algorithm such as EILBG and AILBG. Moreover, we have shown how the use of a distance measure induced by a Gaussian kernel allows to deal more effectively with noisy data, while enforcing the robustness of the algorithm with respect to the choice of the meta-parameters.

In general, we advise that the adoption of the global minimum heuristic might prove to be efficient in enhancing the performance of existing kernel based learning vector quantizers and clustering algorithms, at the cost of a small increase in the computational complexity of the algorithm. In particular, the addition of the repulsive term requires to compute, for each pattern presentation, an additional kernel distance between the winner prototype and the dataset mean. The latter term has to be estimated, e.g. by kernel-weighted fixed point iteration, only once at the beginning of learning.

In this chapter, we presented some encouraging preliminary results on small scale problems. However, we expect the performance increase to be higher in more complex tasks involving highly non-linear error surfaces. Some of these problems might not be solvable with the use of differentiable kernels: hence the formal derivation steps developed in this chapter might not be applicable. However, we think that the global minimum condition can serve as a guideline to devise heuristic repulsive terms that can enhance the performance on complex quantization and clustering tasks, such as, for instance, those involving kernels for sequences and structured data (e.g. strings, proteins, graphs, etc.) (XW05).

Chapter 5

Simultaneous Clustering and Feature Ranking by CoRe Clustering

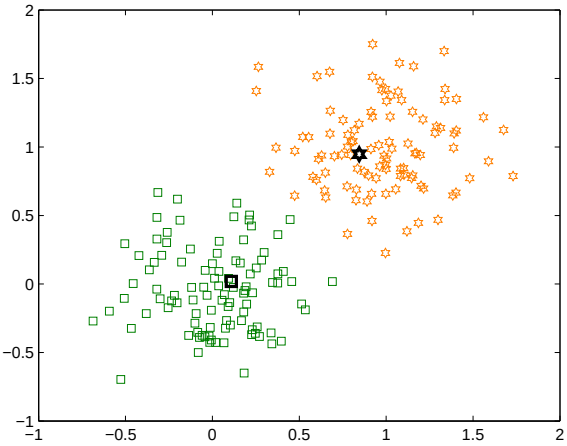
5.1 Introduction

So far, we have described how the repetition suppression mechanism can be used to deal with the cluster number identification problem by suppressing weakly selective units. In particular, we have focused our analysis, as well as the experimental evaluation, on problems characterized by small to medium dimensionality. In this chapter, we study the application of CoRe learning to high dimensional data, investigating how we can extend the repetition suppression mechanism to deal not only with cluster number identification, but also with feature selection.

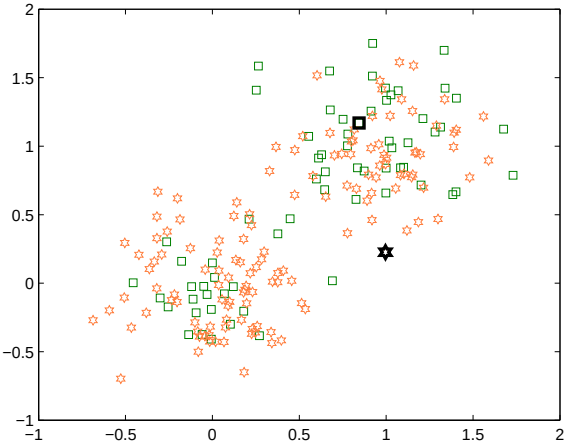
The starting point of our analysis comes from the observation of a simple example comprising the identification of two Gaussian clusters. Consider, for instance, the *toy* scenario depicted in Fig. 37.a: the plot shows 200 datapoints generated by two Gaussian sources in a bi-dimensional space as well as the prototypes identified by the CoRe procedure in Algorithm 1. As noted in Chapter 2, CoRe can estimate the actual cluster number positioning the prototypes close to the clusters' centroids. The orig-

inal two-Gaussian dataset has then been extended by adding 400 *noisy* components generated by random sampling from a uniform distribution in $[0, 1]$. Hence, the sample size has been increased from two to 402. Figure 37 shows the outcome of CoRe clustering on the extended dataset: for the sake of clarity only the two relevant features corresponding to the Gaussian covariates are shown. Obviously, CoRe is no more able to separate correctly the two clusters due to the interference produced by the 400 noisy components. However, it is clear that we can recover the original data partition by isolating the two Gaussian covariates from the 400 random noise features conveying no class information. In other words, the Gaussian covariates are *significative* features that enclose essential information concerning class separation, while the 400 noisy components are *irrelevant* features whose only effect is worsening the chances of identifying the actual data partitions. Therefore, it would be of great interest to extend CoRe's penalization scheme to tackle selective suppression of such irrelevant components, hence reducing their negative effect on cluster identification when dealing with high dimensional data.

Cluster discovery in high dimensional data has became a major research topic since the advent of high throughput data collection tools. Consider, for instance, how DNA microarray technology has made available extensive collections of data comprising information related to the expression levels of thousands of genes with respect to the most important biological processes. The discovery of the patterns hidden in gene expression data as well as the identification of the most informative portions of this consistent collections are two open challenges that require the development of innovative clustering models. The effectiveness of *classical* algorithms is, in fact, highly limited by the the nature of the genomic datasets, that are characterized by a small cardinality (i.e. the number of samples) in a high dimensional space (i.e. the number of genes). Typically, distance-based clustering algorithms such as k-means, fuzzy c-means and rival penalized competitive learning cannot effectively deal with high dimensional data, since they seek for areas where the samples are especially dense (FMR05). In addition, it is likely that the vast majority of the genes in the samples are irrelevant for the realization of the



(a)



(b)

Figure 37: Influence of redundant and noisy features on clustering performance: (a) CoRe correctly identifies the two Gaussian clusters on a simple 2-dimensional scenario; (b) the addition of 400 noisy components prevents CoRe from detecting the actual Gaussian clusters. The plots show only the two relevant features corresponding to the Gaussian covariates; identified centroids are marked in bold.

biological processes that we are willing to discover: hence, the presence of such noisy components has the sole effect of making cluster detection more complicated (Din02). Recently, several algorithms have been proposed to explicitly deal with the peculiarities of high dimensional data and, in particular, with feature selection. With respect to the latter issue, the approaches in the literature are usually differentiated into *filters*, that select variables by ranking them contemporarily to data clustering, and *wrappers* that exploit computationally intensive subset selection methods to generate a set of candidate clustering solutions. The CLIFF algorithm (XK01), for instance, tackles clustering of microarray data by exploiting feature filtering to reduce the effect of irrelevant and redundant features. Given a reference partition that approximates the correct clustering of the samples, CLIFF ranks the features according to their discriminative power, relevance to the reference partition, and non-redundancy with respect to other relevant features. This ranking is then used to select the features for the clustering phase, that is performed based on a normalized cut algorithm. The two modules (i.e. clustering and feature selection) are combined in such a way that a new reference partition is computed by the clustering module and used to rank the genes; the selected features are then used to generate a new data partition and the whole process is iterated until convergence. Consensus clustering (MTMG03) is another unsupervised learning algorithm that deals with high-dimensional data clustering using a resampling approach: in particular, this model constructs a consensus matrix that measures the agreement of sample partitioning between multiple runs of a traditional clustering algorithm, e.g. k-means, SOM, etc. Similarly, (TZZR01) addresses sample clustering by an interplay between sample partition detection and irrelevant gene pruning. First, samples and genes are grouped into several small partitions by conventional clustering methods. Then, the algorithm isolates *representative* partitions characterized by an high internal coherence and by a neat difference with respect to the other groups. Representative groups are selected as candidate dataset partitions and used to guide the elimination of irrelevant genes. The max-min cut hierarchical clustering algorithm (Din02) takes a rather different approach, isolating those genes

which show a large variance in the expression matrix and feeding them to a divisive clustering module that generates the data partitions. Notice that this approach relies on the assumption that informative genes exhibit larger variance than irrelevant features, which is not necessarily true, especially in DNA microarray data.

In general, the majority of models dealing with high dimensional data clustering and feature ranking are based on supervised learning, while the few unsupervised models often exploit computationally intensive approaches such as resampling, ensemble methods or combinatorial searches (FMR05) to obtain good clustering results (see (GE03) and (JTZ04) for extensive reviews on feature selection and high dimensional data clustering). Notice that, in the literature, there exist several model-based approaches to unsupervised cluster number estimation and feature selection (see the brief review in Section 2.3.2). However, model-based approaches are rarely applied to high dimensional data, due to their intrinsic fragility when dealing with insufficient data support to estimate the model parameters. In this chapter, we extend the CoRe clustering algorithm introduced in Chapter 2 to deal with the cluster number identification problem with high dimensional data: as a by-product of its execution, the extended CoRe algorithm produces a relevance ranking that can be used to select the most informative features of each cluster. In particular, we show how repetition suppression can be used to reduce the effect of irrelevant and redundant features and to increase the clustering performance when dealing with high dimensional feature spaces. In our intention, this model should serve as a computationally simple tool (i.e. in the order of magnitude of k-means or FCM) for the unsupervised exploration of high dimensional data, producing incremental evaluations of the data partitioning and features relevance as learning proceeds. Experimental evaluation of the feature-wise CoRe performance is given in Section 5.3.1, with respect to clustering and feature ranking with high dimensional DNA microarray data, while in Section 5.3.2, CoRe is compared with Bayesian model-based algorithms on a small dimensional dataset.

5.2 Learning to Suppress Irrelevant Features with CoRe Clustering

This section describes the feature-wise extension of the CoRe learning model described in Chapter 2: Section 5.2.1 summarizes the ideas underlying its development and introduces an alternative CoRe formulation that highlights its interpretation as a competitive neural network model; Section 5.2.2 focuses on the procedural details of the feature-wise CoRe clustering algorithm and describes the learning equations obtained for Gaussian Core units.

5.2.1 Feature-Wise Repetition Suppression

The CoRe clustering process described in Chapter 2 performs cluster identification without considering the relevance of the single features composing the input vectors neither in its decision procedure nor at the learning stage. In particular, the repetition suppression mechanism, as well as the stimulus predominance and the relevance factor parameters are defined and used only on a per-unit basis. Figure 38 visually describes the computation of the relevant variables in the original CoRe model: at each time instant, a unit u_i processes a vector x_k computing a *scalar response* $\varphi_i(x_k)$ that represents the unit activation. Similarly the units, or a subset of them corresponding to the winner neurons, update the scalar stimulus predominance ν_i^t and relevance factor $\hat{\nu}_i^t$. We recall that the former parameter represents an estimate of the frequency of patterns similar to the vector x_k, while the latter factor measures the relevance of the i-th unit in the neural coding of the data set. Finally, the stimulus predominance ν_i^t, along with the unit response $\varphi_i(x_k)$ of the winner neurons are used to compute the scalar RS_k^t term that is used to suppress the activation of the loser units. Obviously none of the mentioned terms differentiates between the single components of the input vector; rather, they encode information on a per-unit basis, i.e. related to the patterns x_k on the whole.

It is clear from the example in Fig. 37 that such an approach, which

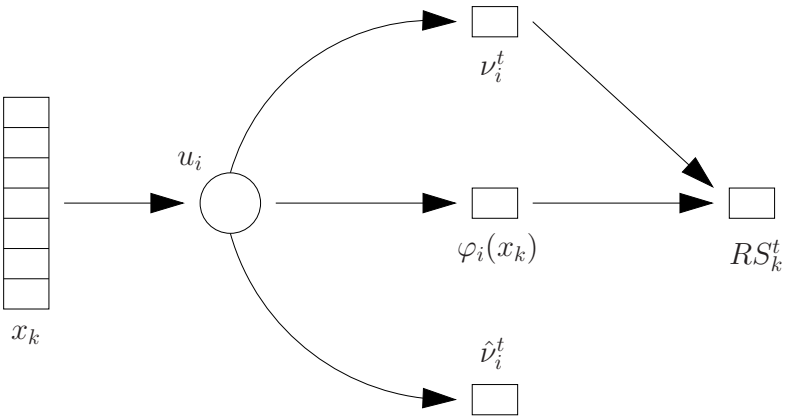


Figure 38: Scheme of a scalar CoRe unit.

flattens the information content of the single features, impairs CoRe performance when dealing with high dimensional input spaces. However, it is also clear that the CoRe model can be naturally extended by introducing the repetition suppression competition on a per-feature basis. In our intention this should lead to an algorithm that, by means of the RS mechanism, is capable of automatically establishing the unknown cluster number, while ranking the input features to determine those that are the most relevant for cluster membership, whilst suppressing the influence of those components that are not significant.

The key intuition underlying the feature-wise CoRe extension is to turn the scalar parameters of the model in Fig. 38 into vectorial terms: Fig. 39 sketches this idea by means of the same graphical notation used for the original algorithm. First of all, we differentiate between the feature activation vector $\varphi_i^t = [\varphi_{i1}, \ldots, \varphi_{il}, \ldots, \varphi_{id}]^T$ and the scalar unit activation $\overline{\varphi}_i$. The former term identifies the activation of the i-th unit in response to the single components x_{kl} of the input vector, while the latter determines the overall response of the unit to the pattern x_k. The stimulus predominance ν_i^t and the relevance factor $\hat{\nu}_i^t$ are both d-dimensional vectors with components ν_{il}^t and $\hat{\nu}_{il}^t$, respectively. The parameter ν_{il}^t mea-

sures the mean firing frequency of the l-th component of the i-th prototype vector and is used, along with the feature activation φ_{il}, to compute the repetition suppression vector $RS_k^t = [RS_{k1}^t, \ldots, RS_{kl}^t, \ldots, RS_{kd}^t]^T$.

The suppression component RS_{kl}^t can be used to selectively penalize the single features of the activation vector φ_i of loser neurons, resulting in a fine-grained update policy for the prototypes components c_{il} (as well as for the rest of the learned CoRe parameters, e.g. the variance σ_i in Gaussian units). Essentially, the features that frequently generate stronger responses to patterns ascribed to the prototype c_i, i.e. those with high ν_{il}^t, produce higher amounts of repetition suppression RS_{kl}^t. In other words, the components that characterize more a given pattern repel the prototypes of the other clusters more strongly than non selective features, resulting in a better separation between the classes. By means of this process, we obtain a neural coding where non-selective features have smaller influence on the pattern classification than components that are *specific* for a given cluster. The relevance vector $\hat{\nu}_i^t$, on the other hand, provides a measure of the significance of the single input features in determining the patterns' membership to the i-th cluster: the components $\hat{\nu}_{il}^t$ supply a feature relevance ranking that can be used, for instance, to prune non-significative input dimensions. Moreover, the vector $\hat{\nu}_i^t$ still provides information concerning the overall relevance of the i-th unit: this knowledge can be used to prune excessive neurons from the network, likewise the original CoRe algorithm described in Chapter 2.

The details of the feature suppression process described above can be well understood by resorting to an alternative formulation that introduces the CoRe learning model as a competitive neural network with lateral inhibition. Within this context, a CoRe model is a two-layer neural network (see Fig. 40) where input nodes are fully connected to a layer of output units $U = \{u_1, \ldots, u_N\}$ that compete with each other through lateral connections. Each output unit u_i is associated to a prototype vector $c_i \in \mathbb{R}^d$, determining its preferred stimulus, and to an activation $\varphi_i(x_k) \in \mathbb{R}^d$ that determines the feature-wise response to the input pattern $x_k \in \mathbb{R}^d$. The activation $\varphi_{il}(x_k)$ is determined for each of the l components of the d dimensional input vector x_k (for the sake of sim-

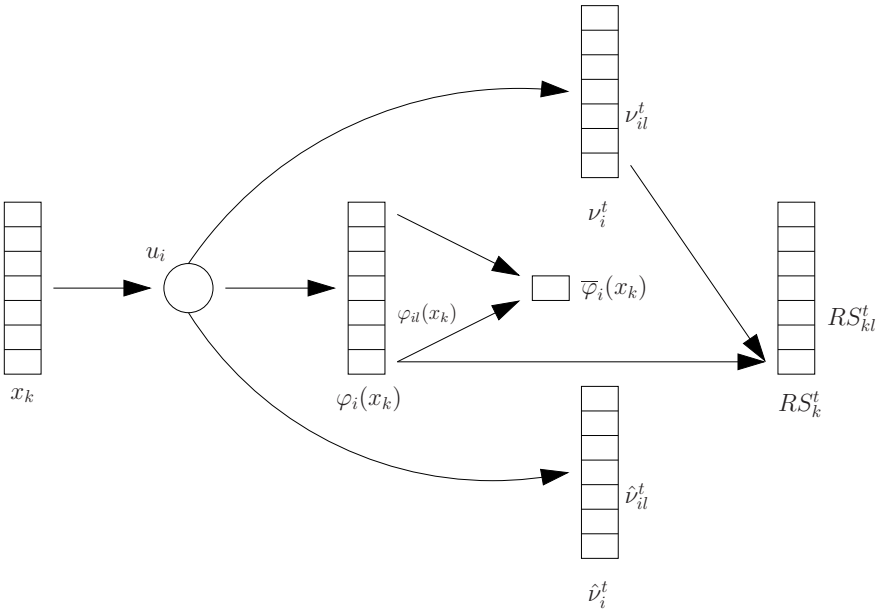


Figure 39: Scheme of a vectorial CoRe unit.

plicity we take the usual assumption $\varphi_{il} \in [0, 1]$); moreover, the overall (scalar) unit output $\overline{\varphi}_i^t$ is calculated as a function f of the feature activation, i.e. $\overline{\varphi}_i^t = f(\varphi_{i1}^t(x_k), \ldots, \varphi_{id}^t(x_k)) = f(\varphi_i(x_k))$. The definition of the activation function as well as the form of the pooling function f determines the amount of interaction between the features. In the remainder of the chapter we assume that the activation of a component l depends solely on the l-th input feature, i.e. $\varphi_{il}^t(x_k) = \varphi_{il}^t(x_{kl})$: this choice corresponds to the common assumption of feature independence. Typically, the function f is chosen to return the sum or the average of the features' activations.

The units in the outer layer are fully connected through a set a lateral inhibitory connections that serve to convey the suppressive potential to the loser neurons. More in detail, given an input pattern x_k we first cal-

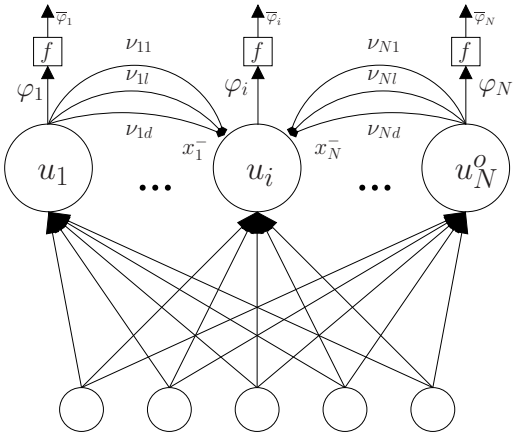


Figure 40: CoRe neural network with lateral inhibition: intra-layer connections link the competitive neurons u_i and propagate the Repetition Suppression. Lateral inhibition is applied selectively to the single input components l, i.e. there are d lateral connections from neuron u_j to neuron u_i, each weighted with a (possibly) different ν_{il}^t.

culate the feature-wise activation $\varphi_i(x_k)$ of each output neuron u_i. Then, following the CoRe competitive scheme in Chapter 2, we select the units with the highest overall activation $\overline{\varphi}_i(x_k)$ to form the *winners pool*, i.e.

$$win_k = \{i \mid \overline{\varphi}_i(x_k) \geq \theta_{win},\ u_i \in U\} \cup \{i \mid i = \arg\max_{u_j \in U} \overline{\varphi}_j(x_k)\}, \qquad (5.1)$$

while the remainder of the neurons is inserted into the *losers pool*, that is

$$lose_k = \{i \mid \overline{\varphi}_i(x_k) < \theta_{win},\ u_i \in U\} \setminus \{i \mid i = \arg\max_{u_j \in U} \overline{\varphi}_j(x_k)\}. \qquad (5.2)$$

Notice that, while we calculate separated activations for each of the input components, we determine winners and losers of the competition in accordance with the overall unit activation. This aspect is fully coherent with the repetition suppression behavior, where the winner neurons are those that are more selective for the input pattern considered as a whole, i.e. in all its features. The key difference concerns the update process that now allows to treat each feature selectively, assigning different rewards and penalizations depending on the components significance.

Besides the scalar activation $\overline{\varphi}_i$, each unit u_i generates an inhibitory output x_{il}^- for each of the l components of the d dimensional input, that is

$$x_{il}^- = \begin{cases} \varphi_{il}(x_{kl})\nu_{il}^t & \text{if} \quad i \in win_k \\ 0 & \text{if} \quad i \in lose_k \end{cases} \tag{5.3}$$

where $0 \leq \nu_{il}^t \leq 1$ (i.e. the stimulus predominance in Chapter 2) represents the weight of the inhibitory connection for the l-th component (see lateral links in Fig. 40). As discussed previously, the inhibitory weight vector $\nu_i^t \in \mathbb{R}^d$ measures the frequency of the patterns that are preferred by the i-th unit. The value of its l-th component (at time t) is computed as

$$\nu_{il}^t = \frac{1}{|\chi_t|} \sum_{x_k \in \chi_t} \min\left(\frac{\varphi_{il}(x_{kl})}{\varphi_{z(U,x_k)l}(x_{kl})}, 1\right) \tag{5.4}$$

where χ_t is the set of patterns x_k presented to the network up to time t and $z(U, x_k)$ returns the index of most active unit for the pattern x_k, that is

$$z(U, x_k) = \arg\max_{u_j \in U} \{\overline{\varphi}_j(x_k)\}. \tag{5.5}$$

The expression in (5.4) recalls the original definition of the stimulus predominance in (2.22): notice that the normalization factor in (5.4) does not ensure the condition $\nu_{il}^t \leq 1$ since the most active unit $u_{z(U,x_k)}$ might be characterized by feature responses that are weaker than in the other units. As a consequence, there might exist some components l' such that $\varphi_{z(U,x_k)l'} \leq \varphi_{il'}$. Hence, we use the thresholding function $\min(\cdot, 1)$ to enforce the condition $\nu_{il}^t \leq 1$. In general, the denominator $\varphi_{z(L^0,x_k)l}(x_{kl})$ can be replaced with other suitable terms such as, for instance, the total feature activation $\sum_j \varphi_{jl}(x_{kl})$. Notice that, by using this latter normalization, there is no more need of thresholding ν_{il}^t by means of the $\min(\cdot, 1)$ function, whose argument becomes the well known soft-max function.

The inhibitory output generated by the network is propagated to all the competing neurons through the lateral connections and is accumulated at each unit i, yielding

$$RS_{il}^t(x_k) = \frac{1}{|win_k|} \sum_{j \in I_i^-} x_{jl}^- \tag{5.6}$$

that is the l-th component of the *repetition suppression* generated at time t for the pattern x_k. The term $|win_k|$ represents the cardinality of the winners pool for x_k and is used to ensure $0 \leq RS_{il}^t \leq 1$, while I_i^- is the set of the inhibiting connections for the i-th neuron. In a sense, the RS_{il} term can be considered as the net value of the inhibiting inputs in the typical notation of the artificial neural network literature.

Following the CoRe model described in Section 2.4, we can exploit the generated RS_{il} inhibition to selectively suppress the prototype features. For instance, by generalizing the results in (2.32) and (2.34) we can update the l-th component of the prototype vector c_i proportionally to

$$\triangle c_{il}^t \approx \alpha_c \left[\delta_{ik} - (1 - \delta_{ik})\varphi_{il}(x_{kl})(RS_{il}^t(x_k))^2\right] \varphi_{il}(x_{kl})(x_{kl} - c_{il}^{t-1}) \quad (5.7)$$

where α_c is the learning rate, while δ_{ik} is the usual indicator function for the winners pool, that is $\delta_{ik} = 1$ if $i \in win_k$ and $\delta_{ik} = 0$ otherwise. We remark that equation (5.7) describes only an exemplar update rule: in Section 5.2.2 we discuss in detail the parameter learning rules for Gaussian CoRe units. Equation (5.7) suggests that CoRe's lateral inhibition plays a substantially different role with respect to the typical lateral connections in competitive learning. The latter ones are often used as part of the competitive mechanism to determine the winner units for a given input pattern, but they do not convey any learning signal to the neurons. ART (CG88), for instance, uses the lateral inhibition to selectively shut-off committed neurons, allowing other nodes in the network to win the competition. CoRe, on the other hand, exploits the intra-layer links to propagate a teaching signal that produces a *long-term silencing* of the neurons and a suppression of the irrelevant input components. CoRe's lateral connections have somewhat much more in common with SOM lateral connections, that are indeed used to convey positive learning to other nodes of the network, although the two models differ significantly in the way they determine the neurons that receive teaching signals through the lateral connections (e.g. topographic proximity in SOM and activation sharpness in CoRe).

5.2.2 The Algorithm

To introduce the feature-wise CoRe clustering algorithm we consider the same network setup used in Chapter 2, comprising Gaussian activation functions with independent features. Hence, the l-th feature activation for the i-th unit can be written as

$$\varphi_{il}(x_{kl}|\{c_{il},\sigma_{il}\}) = \exp\left(-\frac{(x_{kl}-c_{il})^2}{2\sigma_{il}^2}\right), \tag{5.8}$$

while the unit activation corresponds to the multivariate Gaussian

$$\overline{\varphi}_i(x_k|\{c_i,\sigma_i\}) = \exp\left(-\sum_{l=1}^{d}\frac{(x_{kl}-c_{il})^2}{2\sigma_{il}^2}\right). \tag{5.9}$$

The terms c_i and σ_i on the left-hand side of the identities in (5.8) and (5.9) point out which parameters are learned by CoRe. Notice that in (5.9), the function f (see Fig. 40), which pools the feature activations to generate the scalar unit response $\overline{\varphi}_i$, is implicitly chosen as the product of the components φ_{il}.

The feature-wise stimulus predominance of unit u_i at time t is the d-dimensional vector $\nu_i^t = [\nu_{i1}^t,\ldots,\nu_{il}^t,\ldots,\nu_{id}^t]^T$, where d is the cardinality of the input space χ and ν_{il}^t is the stimulus predominance restricted to the l-th feature defined previously in (5.4). Similarly, the feature-wise repetition suppression is a vector $RS_k^t = [RS_{k1}^t,\ldots,RS_{kl}^t,\ldots,RS_{kd}^t]^T$ that applies different levels of penalization RS_{kl}^t to each component l of the losers activation function φ_i. Hence, by adapting (2.24) to comply with the new formulation, we obtain

$$RS_{kl}^t = \frac{1}{M|win_k|}\sum_{i\in win_k}\nu_{il}^t\varphi_i(x_{kl}|\{c_{il},\sigma_{il}\}) \tag{5.10}$$

where the normalizing factor M can be omitted since we are dealing with Gaussian activation functions (i.e. $M=1$).

To specify the algorithm for feature-wise CoRe clustering we need to derive the update rules for the unit parameters , i.e. $\lambda_i = \{c_i,\sigma_i\}$ for Gaussian activation functions, such that c_i is the d-dimensional prototype and σ_i is a d dimensional vector containing the diagonal elements of the

variance matrix of unit u_i. Following the general CoRe framework in Chapter 2, we can derive the update rules for the Gaussian parameters by differentiating CoRe errors with respect to each parameter component $l = 1, \ldots, d$, that is

$$\triangle \underline{\lambda}_{il} = \frac{\partial \underline{E}^t_{il,k}}{\partial \lambda_{il}}$$

for the loser units and

$$\triangle \overline{\lambda}_{il} = \frac{\partial \overline{E}^t_{il,k}}{\partial \lambda_{il}}$$

for the winners, where

$$\underline{E}^t_{il,k} = \frac{1}{2}(\varphi_{il}(x_{kl}|\lambda_{il})(1 - RS^t_{kl}) - \varphi_{il}(x_{kl}|\lambda_{il}))^2 \tag{5.11}$$

and

$$\overline{E}^t_{il,k} = (1 - \varphi_{il}(x_{kl}|\lambda_{il})). \tag{5.12}$$

are the feature-wise loser and winner errors, respectively. In the remainder of the chapter, we will omit the explicit mention to λ_{il} in the arguments of φ_{il} to ease the notation. The parameter increments for the units $u_i \in lose_k$ are obtained by differentiating (5.11) with respect to c_{il} and σ_{il}. Hence, the prototype update at time t can be calculated as

$$\triangle \underline{c}^t_{il,k} = \frac{\partial \underline{E}^t_{il,k}}{\partial c_{il}} = \varphi_{il}(x_{kl}) RS^t_{kl} \frac{\partial(\varphi_{il}(x_{kl}) RS^t_{kl})}{\partial c_{il}} \tag{5.13}$$

where the differentiation on the right side can be expanded by chain rule as

$$\begin{aligned} \frac{\partial(\varphi_{il}(x_{kl}) RS^t_{kl})}{\partial c_{il}} &= \frac{\partial(\varphi_{il}(x_{kl}) RS^t_{kl})}{\partial \varphi_{il}(x_{kl})} \frac{\partial(\varphi_{il}(x_{kl}))}{\partial c_{il}} \\ &= \left(RS^t_{kl} + \varphi_{il}(x_{kl}) \frac{\partial(RS^t_{kl})}{\partial \varphi_{il}(x_{kl})} \right) \cdot \frac{\partial(\varphi_{il}(x_{kl}))}{\partial c_{il}} \\ &= RS^t_{kl} \varphi_{il}(x_{kl}) \frac{(x_{kl} - c_{il})}{\sigma^2_{il}} \end{aligned} \tag{5.14}$$

in which we have used $\frac{\partial(RS^t_{kl})}{\partial \varphi_{il}(x_{kl})} = 0$ if $u_i \in lose_k$ at time t (follows from the definition of RS in (5.10)). Substituting the results of (5.14) in (5.13)

we obtain

$$\triangle \underline{c}_{il,k}^{t} = \left(\frac{\varphi_{il}(x_{kl}) RS_{kl}^{t}}{\sigma_{il}} \right)^{2} (x_{kl} - c_{il}). \tag{5.15}$$

Similarly, the spread update at time t can be calculated as

$$\begin{aligned} \triangle \underline{\sigma}_{il,k}^{t} &= \frac{\partial \underline{E}_{il,k}^{t}}{\partial \sigma_{il}} = \varphi_{il}(x_{kl}) RS_{kl}^{t} \frac{\partial (\varphi_{il}(x_{kl}) RS_{kl}^{t})}{\partial \sigma_{il}} \\ &= \left(\varphi_{il}(x_{kl}) RS_{kl}^{t} \right)^{2} \frac{(x_{kl} - c_{il})^{2}}{\sigma_{il}^{3}} \end{aligned} \tag{5.16}$$

while the parameter increments for the units $u_i \in win_k$ can be written as follows

$$\triangle \overline{c}_{il,k}^{t} = \frac{\partial \overline{E}_{il,k}^{t}}{\partial c_{il}} = -\varphi_{il}(x_{kl}) \frac{(x_{kl} - c_{il})}{\sigma_{il}^{2}} \tag{5.17}$$

$$\triangle \overline{\sigma}_{il,k}^{t} = \frac{\partial \overline{E}_{il,k}^{t}}{\partial \sigma_{il}} = -\varphi_{il}(x_{kl}) \frac{(x_{kl} - c_{il})^{2}}{\sigma_{il}^{3}}. \tag{5.18}$$

The update rule for the unit parameters $\lambda_i^t = (c_i^t, \sigma_i^t)$ is

$$\lambda_i^t = \begin{cases} \lambda_i^{t-1} - \alpha_\lambda^{lose} \triangle \underline{\lambda}_{i,k}^{t} & \text{for} \quad i \in lose_k \\ \lambda_i^{t-1} - \alpha_\lambda^{win} \triangle \overline{\lambda}_{i,k}^{t} & \text{for} \quad i \in win_k \end{cases} \tag{5.19}$$

where α_λ^{win} and α_λ^{lose} are the learning and de-learning rate, respectively, with $\alpha_\lambda^{win} \gg \alpha_\lambda^{lose}$.

Algorithm 3 describes feature-wise CoRe clustering using pseudo-code: the cluster number identification process is, essentially, the same described for the original model (see Algorithm 1). Initially, a number units is generated and, at each pattern presentation, we compute the unit activations and update the parameters on a per-feature basis; if the relevance factor of a unit falls below a predefined threshold θ_{pr-u} then the unit is pruned from the network. As described in Chapter 2, this process is iterated until there is a residual RS competition between the units, measured by the Mean Applied Repetition Suppression (MARS) score. The original MARS definition in (2.39) is updated to comply with the current feature-based scenario, yielding

$$MARS^{t} = \frac{1}{|\chi^{t}|} \sum_{k=1}^{|\chi^{t}|} \sum_{i \in lose_k} \sum_{l=0}^{d} \frac{RS_{kl}^{t} \cdot \varphi_{il}(x_{kl})^{2}}{d}. \tag{5.20}$$

Algorithm 3 Feature-wise CoRe Clustering

Given: the dataset χ, the winners threshold θ_{win}, the unit pruning threshold θ_{pr-u}, the dimension pruning threshold θ_{pr-d} and the network size N

Set $t = 0$, $\theta_{decay} = 1 - \frac{1}{K}$, $\nu_i^t = 1$ and $\hat{\nu}_i^t = 1$;
Randomly initialize the N prototypes c_i^t;
Initialize $(\sigma_i^0)^2$ using (2.38);
repeat
  Generate a random presentation order for the patterns in χ;
  for all $x_k \in \chi$ **do**
    $t = t + 1$
    for $i = 1$ to N **do**
      Calculate feature activation $\varphi_{il}^t(x_{kl}) = \exp\left(-\frac{(x_{kl}-c_{il})^2}{2\sigma_{il}^2}\right)$;
      Calculate unit activation $\overline{\varphi}_i^t(x_k) = \exp\left(-\sum_{l=1}^{d} \frac{(x_{kl}-c_{il})^2}{2\sigma_{il}^2}\right)$;
      Update win_k, $lose_k$ and $win_{u_i}^t$;
    end for
    for $i = 1$ to N **do**
      Update ν_i^t using (5.4);
      if $i \in win_k$ **then** Update $\hat{\nu}_i^t$ using (5.21);
      else Apply the relevance factor decay $\hat{\nu}_i^t = \theta_{decay} \cdot \hat{\nu}_i^{t-1}$;
      end if
      Compute the repetition suppression RS_k^t using (5.10);
    end for
    for $i = 1$ to N **do**
      if $i \in win_k$ **then** Update c_i^t and σ_i^t using the increments in (5.17) and (5.18);
      else Update c_i^t and σ_i^t using the increments in (5.15) and (5.16);
      end if
      if $\hat{\nu}_i^t \leq \theta_{pr-u}$ for all $l = 1, \ldots, d^t$ **then**
        Prune the unit u_i;
        $N = N - 1$;
      end if
      if $\hat{\nu}_i^t \leq \theta_{pr-d}$ for all $i = 1, \ldots, n^t$ **then**
        Remove the l-th feature from all the units;
        $d = d - 1$;
      end if
    end for
  end for
until convergence

So far, we have discussed mainly the process for identifying the clusters number as well as to selectively update the components of CoRe parameters. However, our extension wouldn't be complete without re-defining the relevance factor in (2.27). The feature significance for the unit u_i is measured by the vector $\hat{\nu}_i^t = [\hat{\nu}_{i1}^t, \ldots, \hat{\nu}_{il}^t, \ldots, \hat{\nu}_{id}^t]^T$ where $\hat{\nu}_{il}^t$ is the relevance of the l-th feature for the i-th prototype and is defined as follows

$$\hat{\nu}_{il}^t = \frac{1}{\nu_{il}^t |\chi_t|} \sum_{x_k \in win_{u_i}^t} \left\{ \frac{\varphi_{il}(x_{kl})}{\varphi_{z(win_k, x_k)l}(x_{kl})} \right\}_{\mathbb{0}} \tag{5.21}$$

where

$$\{v\}_{\mathbb{0}} = \begin{cases} 0 & v > 1 \\ v & v \leq 1 \end{cases} . \tag{5.22}$$

The term $z(win_k, \mathbf{x}_k)$ in (5.21) follows the definition in (5.5), while $win_{u_i}^t$ refers to the set of patterns $x_k \in \chi_t$ for which unit u_i was in the winners pool, i.e. $win_{u_i}^t = \{x_k \mid \overline{\varphi}_i(x_k) \geq \theta_{win},\ x_k \in \chi_t\}$. The function $\{\cdot\}_{\mathbb{0}}$ in (5.22), flattens its argument to zero if it exceeds 1. The rationale behind this choice is to penalize the relevance of the l-th component of a prototype c_i whenever it produces an high feature activation $\varphi_{il}(x_{kl})$ in correspondence to a low feature activation $\varphi_{jl}(x_{kl})$ in the unit u_j that is the maximally active neuron for the pattern x_k, i.e. $j = z(win_k, x_k)$ (see Fig. 41 for a graphical interpretation). This process implicitly considers improper responses all those feature activations that do not correspond to highly active features in the maximally responsive unit. In this sense, the output of the best matching neuron becomes, itself, an endogenous training signal.

The vector $\hat{\nu}_i^t \in \mathbb{R}^d$ defines a measure of relevance that can be used to determine which prototype components characterize best the input patterns assigned to the i-th neuron. The feature relevance can be used, on the one hand, to help profiling the data clusters discovered by the CoRe algorithm and, on the other hand, to prune irrelevant or noisy data components. For instance, the same relevance-based strategy used for unit pruning can be applied to select a set of significant components. In Algorithm 3, the features whose relevance factor falls below the threshold θ_{pr-d} for all the units are considered irrelevant and are discarded.

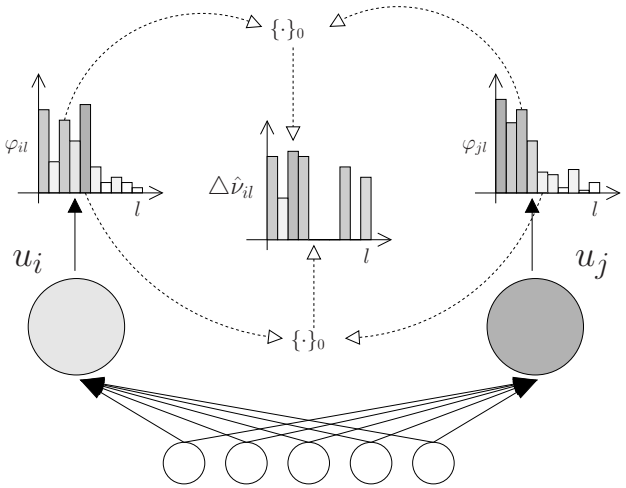
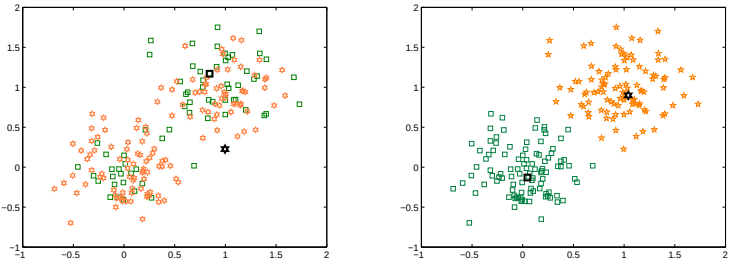


Figure 41: Graphical interpretation of relevance factor calculation in feature-wise CoRe. Darker gray-levels identify increasing levels of feature and unit activation: the most responsive unit u_j is identified by the darkest color. The term $\triangle\hat{\nu}_{il}$ represents the relevance factor increment produced in unit u_i by the current input x_k. Notice how the increment of the 5-th feature is flattened to 0 by the function $\{\cdot\}_0$ since the unit u_j, which is the most selectively tuned to x_k, has a low 5-th feature activation φ_{j5} (bottommost dashed lines). Conversely, the relevance of the 3-rd feature is reinforced proportionally to φ_{i3} because the 3-rd component of the activation function in u_j follows a similar activation pattern (see the topmost dashed lines).

Alternatively, we can opt for an *ex-post* evaluation of the feature relevance, where component pruning is not performed incrementally. In other words, the relevance factor is updated throughout training, but the feature selection step is performed only at the end of the training phase, e.g. by applying a top-k selection with respect to the relevance rank. We would like to remark that the process of feature and unit relevance ranking is performed incrementally during learning, irrespectively of which feature selection mechanism is implemented. Hence, CoRe can be a suitable tool for performing human supervised visual explorations of high dimensional data, where the estimate of cluster and feature relevance is incrementally refined during learning and is continuously made available



(a) (b)

Figure 42: Two Gaussian with 400 noisy components example: (a) original CoRe clustering; (b) feature-wise CoRe. The extended algorithm is able to identify the relevant input components, pruning the 400 noisy features and recovering the original clustering shown in Fig. 37.a.

to the user throughout the whole data analysis process.

Figure 42 compares the solutions generated by the original and the feature-wise CoRe algorithm on the toy example introduced in Section 5.1: clearly, the latter algorithm is able to recover the two original Gaussian clusters by isolating the 400 noisy components. In particular, Figure 42.b has bee obtained with the incremental feature pruning process described in Algorithm 3. However, it is interesting to note that the same results can be obtained with incremental pruning disabled: Figure 43 shows the feature ranking obtained in this scenario. CoRe is able to isolate the first two components, i.e. those corresponding to the Gaussian covariates, as being the only highly relevant for cluster membership (see the bottommost graphs in Fig. 43 for a zoomed plot of the first 50 features).

5.3 Experimental Evaluation

In this section we evaluate the effectiveness of the extended CoRe algorithm. In particular, Section 5.3.1 tests the performance of the model when dealing with sparse data in high dimensional space, while Section 5.3.2 studies how CoRe can be used to analyze a recently published breast can-

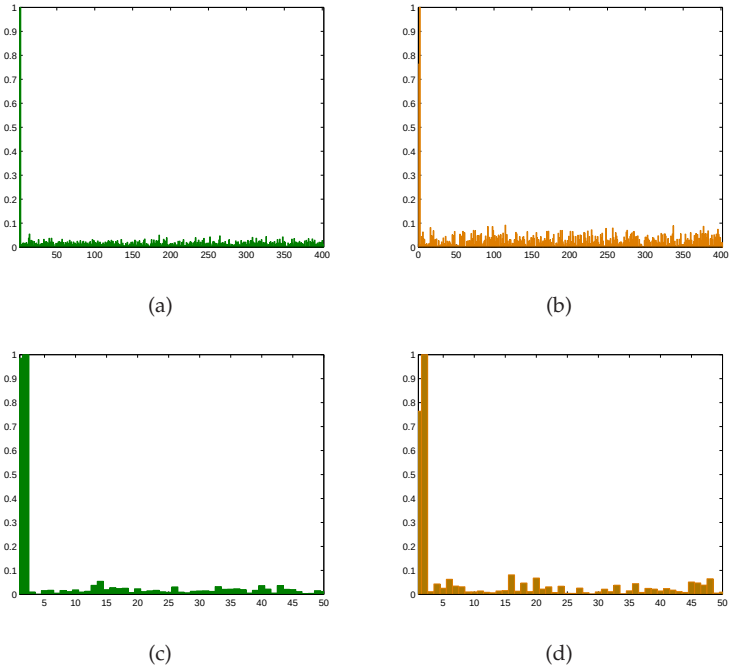


Figure 43: Feature relevance for the two gaussian example: (a) and (b) plot the relevance of the 402 components of cluster 1 and 2, respectively; (c) and (d) show a detail of the first 50 components.

Table 8: CoRe meta-parameters

Description	Name	Value
Winners threshold	θ_{win}	0.9
Unit pruning threshold	θ_{pr-u}	0.005
Dimension pruning threshold	θ_{pr-d}	0.01
Winners' prototype learning rate	α_c^{win}	0.05
Winners' variance learning rate	α_σ^{win}	0.0005
Losers' prototype learning rate	α_c^{lose}	0.005
Losers' variance learning rate	α_σ^{lose}	0.0001

cer case-study (ABQ$^+$06) with the intent of discovering tumoral profiles characterized by different disease dynamics, exploiting CoRe's feature ranking to gather insight into the markers that best describe the discovered bio-profiles.

5.3.1 High-dimensional Data Clustering

DNA microarray analysis is one of the principal sources of high throughput data in high dimensional space. It is therefore natural the we present a first assessment of the feature-wise CoRe clustering performance on consolidated synthetic and real gene expression data collected with DNA microarrays. All the results described in this section have been obtained with Gaussian CoRe units using the procedure described in Algorithm 3. The same meta-parameters and convergence condition described in Chapter 2 are employed: only the definition of the MARS score has been slightly adjusted in (5.20) to comply with the feature wise scenario. The meta-parameters choices are summarized in Table 8; the initial network size has been set to $K = 30$, which results from a trade-off between choosing an overestimation of the actual class number and keeping the execution time limited. All trials have been repeated 10 times and the results averaged.

First, to test the effectiveness of the feature relevance ranking, we considered the Simulated6 (MTMG03) dataset, comprising 60 artificial samples consisting of 600 genes and that is known to be partitioned in 6 clus-

ters of various sizes. This dataset has been specifically designed to have clusters that are characterized by known subsets of significant components, i.e. the gene markers, with varying degree of differential expression between the classes. In particular, the dataset can be partitioned into 6 classes consisting of 8, 12, 10, 15, 5 and 10 samples, respectively, each marked by 50 distinct genes uniquely up-regulated for that class (MTMG03). In addition, it contains 300 noisy genes not "coding" any particular class.

Table 9 describes the clustering performance in terms of the *Adjusted Rand Index* (HA85), that is

$$r = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{K}{2}}{\frac{1}{2}\left[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}\right] - \left[\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}\right] / \binom{K}{2}}$$

where n_{ij} is the number of samples from the i-th class that have been assigned to cluster j, n_i denotes the number of items members of class i and n_j denotes those that are members of cluster j, while K is the dataset size. The Rand index ranges between 0 and 1, where 1 corresponds to the perfect agreement between the two partitions, while 0 matches the expected value for two random partitions. The use of the Rand index is motivated by the fact that it allows to confront the performance of clustering algorithms even when they estimate a different cluster number for the same dataset (see the results in Table 9).

Feature-wise CoRe clustering is compared with agglomerative Hierarchical Clustering (HC) with single linkage, that is a popular tool for the unsupervised analysis of high dimensional data, mainly because of its visualization properties and its ability to provide multi-resolution views of the dataset. The HC model is, originally, not committed to identifying a specific number of clusters, whereas it is often used together with validity indices to obtain a cluster number estimate: the results in Table 9 have been obtained by hierarchical clustering in conjunction with the GAP statistics. The k-means algorithm is used to provide a baseline performance for iterative algorithms: in this case the estimated cluster number was chosen equal to the actual number of classes in the data. Moreover, we compare CoRe's performance with *Consensus Clustering* (MTMG03):

Table 9: Estimated number of clusters N and classification accuracy r (Rand index) obtained by feature-wise CoRe, k-means, hierarchical clustering with GAP statistics (HC), consensus clustering with hierarchical clustering (CC_{HC}) and self organizing maps (CC_{SOM}) (C identifies the actual cluster number).

	Simulated6 $C=6$		Leukemia $C=3$		St. Jude $C=6$		Lung $C=4+$	
Algorithm	N	r	N	r	N	r	N	r
k-means	6	.962	3	.419	6	.834	4	.339
HC	7	.986	5	.648 (.46)	5	.853 (.90)	5	.307 (.28)
CC_{HC}	7	.986	5	.648 (1.0)	5	.899 (.948)	5	.31 (.28)
CC_{SOM}	6	.986	4	.721 (.6)	5/7	.726 (.825)	5/7	.233 (.22)
CoRe	6	1.0	3	.845	5/6	.854 (.936)	4/5	.519

this model constructs a consensus matrix that measures the agreement of sample partitioning between multiple runs of a given clustering algorithm, e.g. k-means. The general idea of the model is to construct a series of perturbed datasets by means of resampling techniques and to run the clustering algorithm on the resampled data with varying cluster numbers N, generating a series of consensus matrices $\mathcal{M}(N)$. The estimated cluster number is that corresponding to the most *regular* consensus matrix, i.e. that containing mainly 0's or 1's, that are the values indicating full consensus on the samples' membership. The results shown in Table 9 have been obtained by consensus clustering based on hierarchical clustering (CC_{HC}) and self organizing maps (CC_{SOM}). There exist several other methods for automatically selecting the number of cluster: among these, model-based Bayesian clustering algorithms allow for the seamless combination of prior knowledge and observational data. However, most of these methods are based on asymptotic approximations of the marginal likelihood, whose accuracy tends to decrease as the sample number decreases with respect to the feature space size, which is typically the case when clustering DNA microarray data. Therefore, we prefer to postpone the comparison between model-based algorithms and feature-wise CoRe clustering to the Section 5.3.2, where we focus on the analysis of a small dimensional data set from breast cancer research.

The results in Table 9 show that CoRe clustering correctly identified the six clusters of the Simulated6 dataset, pruning the irrelevant units and partitioning the data with full agreement with the actual class distribution. Compared with the results obtained by k-means, consensus and hierarchical clustering, CoRe achieves better results both in terms of cluster number identification and classification accuracy. In particular, hierarchical clustering, both when used with the GAP statistics and with consensus clustering, has the tendency to allocate an excessive cluster in correspondence to sample x_8, that is characterized by up-regulated genes both in class 1 and class 2 (MTMG03). On the other hand, the parsimonious cluster allocation policy of CoRe learning does not allow the creation of this singleton cluster, thus obtaining a correct cluster number estimate, likewise the CC_{SOM} algorithm which, however, does not achieve full clustering accuracy. The perfect classification score of CoRe learning can be explained by the fact that it has correctly identified the significant features of each class, suppressing the influence on noisy and non relevant components. Figure 44 shows the bar graphs of the feature relevance corresponding to the 6 clusters identified by CoRe, where the topmost plot refers to the first partition C_1 and the bottommost to cluster C_6. In general, each cluster seems to be characterized by a set of 30 to 50 relevant genes, which is coherent with the a-priori information concerning the distribution of the up-regulated genes in the classes (MTMG03). In particular, in the first two clusters, that are those characterized by the largest gene differential expression, CoRe isolates the significant components with a sharp distinction between relevant and irrelevant genes. Conversely, clusters C_5 and C_6 are characterized by the weakest markers, i.e. small differential expression, which results in a reduced relevance factor variance between significant and non-significant features. However, CoRe is still able to clearly isolate most of the relevant genes characterizing these two clusters. Overall, the relevance factor seems capable of providing an interesting insight into the feature significance of the identified clusters: this ability can be exploited not only to visually explore the data, but also to enhance the performance of other clustering algorithms. Consider, for instance, to perform k-means clustering on a fil-

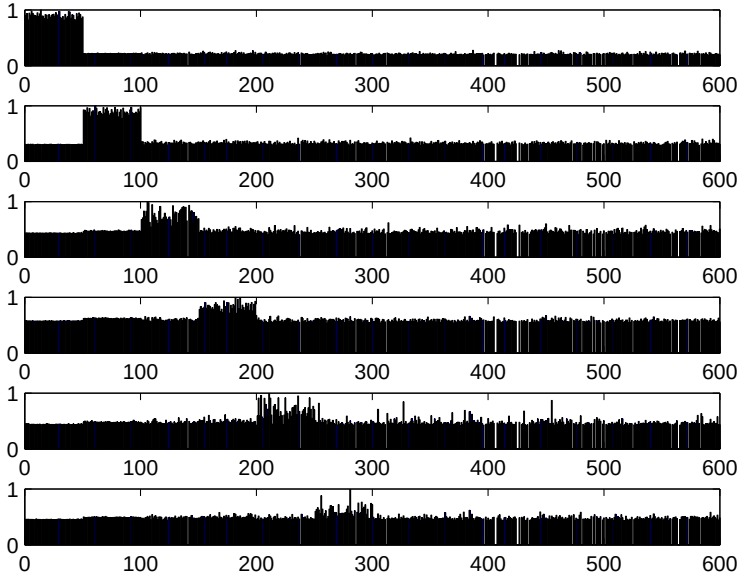


Figure 44: Expression patterns for the 6-class simulated dataset. The bar plots represent the relevance factor $\hat{\nu}_i^t$ (scaled to $[0, 1]$) of the single genes for each of the identified clusters: the topmost cluster refers to class C_1 while the bottommost pertains to C_6.

tered dataset, where samples comprise only the top-50 most informative genes identified by CoRe for each cluster: such a feature selection produces a performance improvement that allows k-means to reach perfect partitioning of the dataset, i.e. with Rand index $r = 1$.

The second set of simulations was run on real gene expression samples from a widely adopted benchmark dataset, that is the Leukemia data by Golub (GST$^+$99). The dataset consists of 38 bone marrow samples comprising the expression levels of 7129 genes obtained from acute leukemia patients at the time of the diagnosis. The data can be partitioned into 3 relevant classes: 8 T-Lineage Acute Lymphoblastic Leukemia (ALL) samples, 19 B-lineage ALL samples and 11 Acute Myeloid Leukemia (AML)

samples. However, there exist a large number of supervised learning algorithms tested on the Leukemia data focused on a reduced problem, that is partitioning the dataset into 11 AML and 27 ALL samples. In general, there is no unambiguous and clear solution when trying to determine the ground truth sample classification in real high dimensional dataset: in this analysis we consider the most common sample partitioning used in the literature for the Leukemia data, that corresponds to the three classes hypothesis. The results in Table 9 show that CoRe identifies three clusters corresponding to the original T-ALL, B-ALL and AML classes, with a clustering performance of 0.845. On the other hand, both the hierarchical clustering with GAP statistics and the consensus-based HC estimate five partitions, with two clusters isolating the AML and T-ALL classes, while the remaining three clusters partition further the B-ALL class. The sub-division of this latter class might, in principle, have an interesting clinical interpretation suggesting a further differentiation of the B-lineage type which can find confirmation in additional clinical investigations (MTMG03). Notice that when the HC model with GAP statistics is forced to partition the data into three classes, it obtains a worse clustering performance with respect to the five cluster solution (see the Rand index in brackets in Table 9). Conversely, the consensus-based hierarchical clustering model obtains full classification accuracy when forced to partition the dataset into 3 cluster. The SOM-based algorithm (i.e. CC_{SOM}), on the other hand, originally estimated four clusters with two clusters splitting the B-lineage type; if forced to partition the data into 3 classes, CC_{SOM} obtains a strong reduction of its clustering performance. Again, the k-means algorithm obtained the lowest Rand scores within the methods.

Overall, CoRe's analysis confirms the three class hypothesis although the best performance on the dataset is achieved by CC_{HC} when it is forced to partition the data into three clusters. It it interesting to note how hierarchical clustering alone (i.e. HC in Table 9) obtains an extremely poor performance, close to that obtained by k-means, suggesting that the resampling strategy embedded in consensus clustering is providing the performance speed-up to the HC algorithm. However, such a performance increase is obtained at the cost of a consistent load raise due to

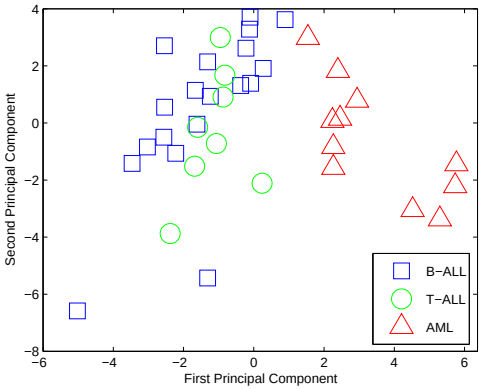


Figure 45: Principal component analysis for the Leukemia dataset on the 7129 genes.

the dataset resampling. The drastic reduction in the clustering performance of k-means and HC can be explained by looking more in depth at the nature of the Leukemia data. In (Din02), it is shown that the leukemia classes overlaps significantly in the 7129-dimensional space, while they are neatly separable when a subset of informative features are used. The CoRe algorithm exploits the feature-wise competition mechanism to suppress the contribution of the irrelevant genes, achieving competitive results even when the full set of features is used and without resorting to time consuming resampling strategies. Figure 45 and 46 give a visual interpretation of the problem, by projecting the Leukemia data along its first two principal components. Figure 45 shows a component plot when all the 7129 genes are used: the two classes in the ALL samples (B-linkage and T-linkage) are completely mixed, while the AML class is not clearly identifiable as a single cluster. Figure 46 shows the PCA result when only the 30 most relevant genes identified by CoRe are used: the AML data is tightly clustered, while the T-ALL and B-ALL classes are almost completely separated already in a bi-dimensional space.

The feature ranking produced by CoRe clustering can be confronted with the list of the 50 most relevant genes for the Leukemia data identified

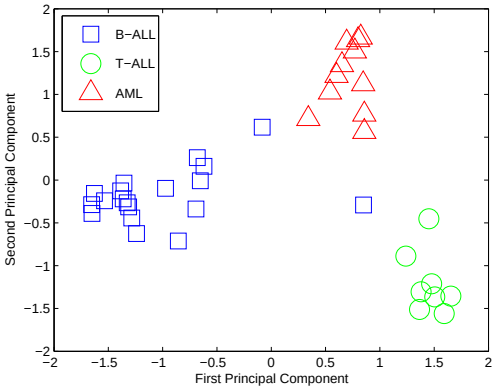


Figure 46: Principal component analysis for the Leukemia dataset on CoRe's top-30 features.

by Golub (GST$^+$99). Table 10 shows an example of 8 genes from the top-20 list generated by CoRe clustering, that are present also in Golub's list: this result confirms that CoRe relevance ranking produces results that are consistent with the state-of-art knowledge concerning informative genes in the leukemia disease. Moreover, if we perform again k-means clustering on the ALL/AML task restricted to the top-20 genes identified by CoRe ranking we obtain a neat improvement in clustering performance, i.e. a Rand index $r = 0.7901$.

To compare the quality of CoRe's feature ranking with other popular feature selection algorithms from microarray data analysis we force CoRe to identify the relevant genes in the 2-class task on the Leukemia dataset, that requires the discovery of the two clusters corresponding to the ALL and AML samples. To do so, we relax the CoRe stopping condition and we force the algorithm to converge again after having pruned one of the three clusters. To compare CoRe's performance with the results in the literature, in this test we express clustering performance in terms of the *Q-accuracy* score, that is the number of correctly classified samples with respect to the dataset size. The Q-accuracy obtained by CoRe on the 7129-dimensional ALL/AML task is 98.7%, reducing to 94.7% when

Table 10: Relevant features identified by CoRe clustering and listed in Golub's top-50 (GST$^+$99).

Code	Description
M27891_at	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)
M28130_rna1_s_at	Interleukin 8 (IL8) gene
M57710_at	LGALS3 Lectin, galactoside-binding, soluble
Y00787_s_at	INTERLEUKIN-8 PRECURSOR
U05259_rna1_at	MB-1 gene
M96326_rna1_at	Azurocidin gene
U22376_cds2_s_at	C-myb gene extracted from Human (c-myb) gene, complete primary cds, and five complete alternatively spliced cds
X1704_at2	PRG1 Proteoglycan 1, secretory granule

only the top-20 genes are used. Running the k-means algorithm on the same ALL/AML task restricted to the top-20 most relevant genes generated by CoRe results in a Q-accuracy of 94.7%, that is the same value obtained by *max-min cut hierarchical clustering* in (Din02) using the top-50 genes. The *two-way unsupervised feature selection* algorithm (Din03), on the other hand, achieves a Q-accuracy score of 76.3% and 79.0% for the full-feature and the top-50 case, respectively. These results show that, on the one hand, CoRe achieves an higher clustering performance with respect to popular filter models in the literature. On the other hand, the fact that also k-means can obtain an high Q-accuracy when applied to CoRe's top-20 selected genes, suggests that the feature ranking produced by CoRe can efficiently identify the most informative, and discriminative, components of the input data. Recently, (FMR05) presented a wrapper approach based on simulated annealing, that achieves 100% Q-accuracy on the ALL/AML task with less than 20 genes. However, this approach requires a considerable computational effort, since it does not seek incremental estimates of the feature relevance, whereas it make an explicit combinatorial search for subsets of 20 genes minimizing the representation error.

The third experiment was performed on the St. Jude's dataset (YRS$^+$02)

that includes 248 diagnostic bone marrow samples from pediatric acute leukemia patients corresponding to six prognostically relevant leukemia subtypes. In particular, the 985-dimensional samples are partitioned as follows: 43 T-lineage ALL, 27 E2A-PBX1, 15 BCR-ABL, 79 TEL-AML1, 20 MLL rearrangements and 64 *hyperdiploid>50* individuals. The results in Table 9 show that none of the algorithms is capable of clearly identifying the actual structure of the dataset. With respect to hierarchical clustering, both the GAP statistics and the distribution of the consensus function criterion suggest that the most likely number of cluster is five. On the other hand, a visual inspection of the consensus matrices $\mathcal{M}$ supports a different hypothesis comprising a six-cluster structure of the data (MTMG03). More specifically, the best separation is obtained for a seven cluster solution, where six clusters correspond almost perfectly with the six leukemia subtypes and one cluster identifies a singleton sample from the *hyperdiploid>50* group. A similar result is obtained by CC_{SOM}, although the resulting sample classification does not match the actual data partition as well as hierarchical clustering does. CoRe learning, on the other hand, strongly supports the five clusters hypothesis. Notice that in a single experiment repetition CoRe learning converged by identifying six clusters instead of five, where the *excessive* partition identified a singleton cluster: as a consequence, the resulting data partition did not differentiate significantly from the five clusters hypothesis, i.e. with a rand Index within 2% deviation from the average CoRe result. Overall CoRe learning obtained the best results together with hierarchical clustering both on the $N = 5$ solution as well as when it was forced to partition the data in six clusters (see the results in parenthesis in Table 9). In the latter case, CoRe achieves an higher classification accuracy than standard HC, reaching a performance that is close to that obtained by the resampling-based HC algorithm CC_{HC} at a reduced computational complexity.

The last set of simulations has been performed on the Lung cancer tissue data (BRS$^+$01). This dataset is known to contain at least 4 classes corresponding to 139 adenocarcinomas (AD), 21 squamous cell carcinomas (SQ), 20 carcinoids (COID), and 17 normal lung (NL) samples. However, there is some discussion concerning the opportunity of partitioning fur-

ther the highly heterogenous AD class, although there is no established hypothesis concerning its actual substructure. The heterogeneity of the AD class makes the identification of the four partitions rather complicated: the results in Table 9 show that hierarchical clustering suggests the existence of five clusters, where three clusters identify quite closely the SQ, COID and NL classes while the remaining two split the AD class into two subgroups. Notice that when HC and CC_{HC} are forced to generate a four cluster partition, they obtain worse results with respect to the original five cluster configuration. The CC_{SOM} algorithm obtains a poorer performance and cannot generate a stable estimate of the cluster number which oscillates between five and seven. The complexity of the Lung dataset affects also CoRe's performance, although it achieves the highest rand index among the five algorithms. CoRe typically identifies four clusters, although two repetitions converged with an additional fifth cluster that splits the AD class into two subgroups. It has to be noted that CoRe obtains an higher rand score with respect to the other models even when it identifies five clusters, e.g. $r_{CoRe-5} = 0.471$ with respect to $r_{CC_{hc}} = 0.31$.

Overall, the Lung cancer tissue data is a challenging dataset that has not yet revealed its full structure. To gather a clearer insight into this data we analyzed the differential expression of the four classes in comparison with the feature relevance generated by the CoRe analysis. Figure 47 shows an histogram of the up-regulated genes in each of the four tumor subtypes, measured by computing a one-versus-all *t-test* using the *Gene Pattern* (RLL+06) online tool[1]. Positive bins in Fig.47 denote characteristic genes for the particular class at hand, while negative values identify differentially expressed features in the other classes. Figure 48 depicts the relevance of the clusters corresponding to the actual AD, COID, SQ and NL classes. Some degree of overlap exists between relevant features as identified by the t-test and by CoRe: for instance, by analyzing the top-50 features with respect to the two relevance rankings we notice that the `31888_s_at` and `40237_at` genes, known for controlling programmed cell death (LF98) and implicated in breast cancer and leukemia (TNY+06),

[1] http://www.broad.mit.edu/cancer/software/genepattern/

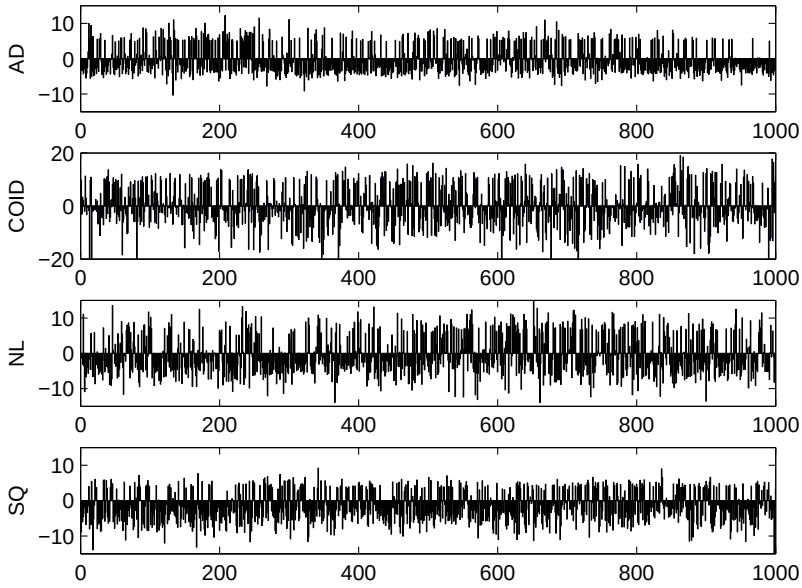


Figure 47: Histogram of the the differentially expressed genes in the four classes of the Lung dataset: differential expression is measured by one-versus-all t-test.

are strongly characterizing the AD samples in both rankings. In general, the COID class is that characterized by the largest overlap between the two relevance measures. However, rather than focusing on the similarities between the two approaches, we are more interested in investigating those prognostic hypothesis that may rise from considering features that are highly relevant in CoRe's analysis but that are not supported by the t-test. For instance, the *dynamin 1* gene (i.e. code `32138_at`) has high CoRe relevance (ranking 10) for COID samples and a rather weak significance in t-test (ranking 231): however, (MBCCGF05) has shown that *dynamin 1* is a strong bio-marker for discriminating pulmonary carcinoids from other tumoral histotypes, thus confirming CoRe's hypothesis.

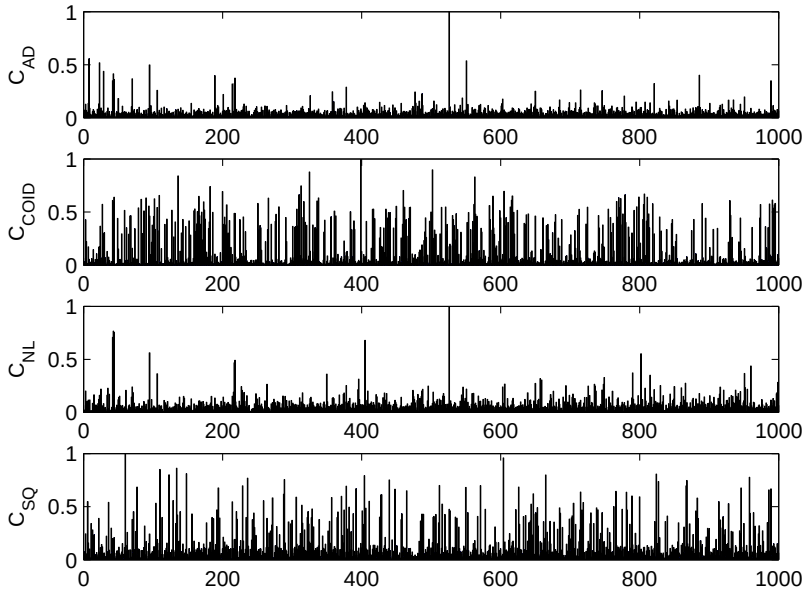


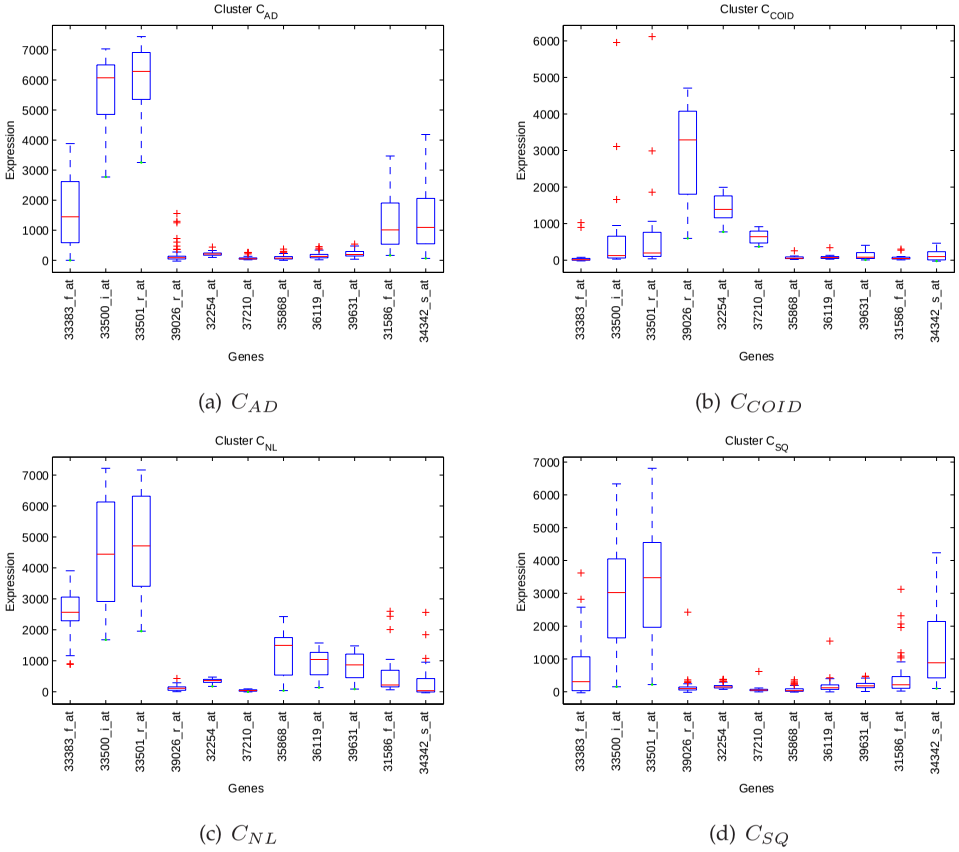
Figure 48: Histograms of the feature relevance obtained for the four CoRe clusters on the Lung dataset.

Interestingly, the same study suggests that a specific neuronal protein, that is the *internexin neuronal intermediate filament protein alpha* (gene code `37210_at`), is important in discriminating carcinoids from other lung tumors: the same protein is ranked by CoRe as the third most relevant in COID samples. On the other hand, the SQ samples seems to be strongly characterized by the `36119_at` gene, that is ranked second by CoRe and 434^{th} (which means irrelevant) by the t-test. This gene, known as *caveolin 1*, is thought to be a major indicator for predicting prognosis in patients with pulmonary squamous cell carcinoma (YPK$^+$03).

Analyzing the feature relevance in the control individuals (NL class) points out the strong dichotomy related to the *fascin* protein (gene code

39070_at) that is ranked as the fifth most significant gene by CoRe while is ranked 804^{th} by the t-test. Such a strong difference highlights the difference between CoRe's feature ranking and one-versus-all t-test: while the latter seeks overexpressed genes in a particular class, the former looks for features that play a key role in differentiating one class from the rest, i.e. genes that influence the separation between the classes. A closer look to the bio-medical interpretation of the fascin protein confirms this interpretation. In fact, fascin is known to be an independent prognostic predictor of various tumoral forms, including lung cancer and, more in detail, adenocarcinomas and squamous cell carcinomas (PPF$^+$03). Consequently, fascin is itself strongly relevant for the characterization of the NL class, since control samples are characterized by average fascin expression, while AD and SQ individuals show fascin overexpression.

By looking at the relevance histograms in Fig. 48, it can be noticed that both the AD and the NL classes are characterized by the same top-rank gene 33383_f_at, expressing the *N-acylsphingosine amidohydrolase-like* (ASAH-like) protein. This protein has recently been ranked fifth in the list of the top markers for lung adenocarcinoma, with 93% of prediction accuracy (SH06b), hence it is fairly straightforward to understand its significance for the AD class. On the other hand, its relevance for the NL class can be understood by looking the distribution of the top-3 genes identified by CoRe for each of the four clusters in the dataset (see Fig. 49). The expression levels of 33383_f_at strongly differentiates NL samples (see plot 49.c) from COID and SQ individuals (see Fig. 49.b and 49.d, respectively). This fact, alone, should produce an high relevance in the 33383_f_at gene of the NL cluster. Moreover, by comparing the NL profile with the AD class we notice that there is a certain amount of overlap in the distribution of 33383_f_at: this superposition produces a strong competition between the two units on gene 33383_f_at. As a result, both clusters experience an additional relevance increase that is due to the dynamics of the repetition suppression competition. In particular, since the AD samples constitute the 70% of the dataset, the expression levels of the ASAH-like protein become significant for separating NL samples from the rest of the dataset.



(a) C_{AD} (b) C_{COID}

(c) C_{NL} (d) C_{SQ}

Figure 49: Profiles of the CoRe top-3 relevant genes for each Lung tissue cluster.

5.3.2 Tumor Classes and Bio-markers Discovery

In the previous section we have analyzed the performance of the feature-wise CoRe clustering algorithm when dealing with the cluster number identification problem in high dimensional space. In this section, we shift our focus on an application example showing how CoRe can be used for the exploratory analysis of biomedical data, providing a means for the unsupervised discovery of clinical profiles and related bio-markers. In particular, we study the performance of the proposed neural approach on a recently published dataset from breast cancer research (ABQ$^+$06), that lacks a consolidated characterization of the actual class distribution. Compared with DNA microarray data, this dataset contains small dimensional samples, allowing to compare CoRe performance with that of Bayesian model-based algorithms that have been excluded from the analysis presented in the previous section for high-dimensional data. In particular, CoRe results are compared with those obtained by three model-based algorithms described in Section 2.3.2, that are

- Gaussian Mixture Model (GMM) (FJ02): this algorithm fits the input samples to a mixture of Gaussians using EM, while the number of components is estimated from the data by pruning those mixtures with zero mixing weights.

- Variational Bayesian Gaussian Mixture (VBG) (CB01): this algorithm uses a variational approximation to fit a fully Bayesian model to the samples; as with GMM it determines the number of mixtures by component pruning.

- Variational Bayesian Mixtures with Splitting (VBS) (CL07): similarly to VBM uses variational methods to fit the data to the model; the number of mixtures is determined by recursively splitting and pruning the existing components.

Additionally, the results obtained by each algorithm are compared with the sample labeling discovered in previous work (ABQ$^+$06) by Partitioning Around Medoids (PAM) (KR90) and k-means (KM).

The dataset comprises information from biomarker proteins that was collected from a consecutive series of 922 patients who underwent surgery for primary infiltrating breast cancer between 1983 and 1992 at the University of Ferrara (ABQ$^+$06). This study analyzed proteins measured on preserved tissue samples, with exploratory aims and without diagnostic nor risk communication to the patient. Data on patient age, pathologic tumor size, histologic type, pathologic stage, and number of metastatic axillary lymph nodes were collected, as well as immunohistologic determinations of estrogen receptor (ER) status, progesterone receptors (PR) status, Ki-67/MIB-1 proliferation index (Pro), HER2/NEU (NEU), and P53 levels. Only the latter five biomarkers have been analyzed as part of this study: the 633 samples used in (ABQ$^+$06), for which is present complete information on all patho-biological measurements, are included in this work. Coherently with (ABQ$^+$06), the expression values of ER, PR, and NEU have been discretized to the following percentages: 0%, 10%, 25%, 50%, 75%, and 100%. Percentages of Ki-67- and p53-expressing cells were recorded without discretization.

The results presented in the following are based on 50 runs of each algorithm, with randomly initialized prototype positions. Both GMM and VBM are initialized with 20 mixtures, VBS is initialized with one component while CoRe starts with 30 units (for the rest of CoRe meta-parameter settings refer to Table 8).

As a first step of the analysis, we focus on determining the most likely number N of sample groups in the data. Figure 50 shows the histogram of the cluster numbers estimated by the four algorithms during the 50 runs. The distribution of N for GMM and VBG closely resembles a Gaussian centered on 7 and 5, respectively. CoRe and VBS produce sharper hypotheses: the former, in particular, suggests that the data can be grouped in 4 or 5 clusters, with few runs terminating with 6 clusters. The behavior of VBS is peculiar: this algorithm behaves similarly to a wrapper approach that tentatively splits the existing components and locally tests the existence of additional clusters, eventually pruning the newly generated mixture; this process is iterated until the splitting test fails for all the components. Within this stopping criterion, VBS always converged to a

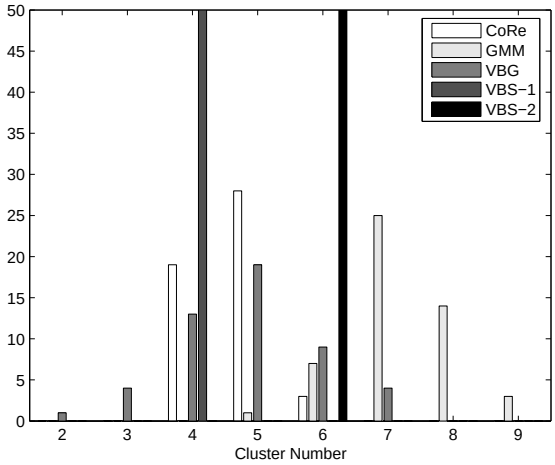


Figure 50: Distribution of the cluster number estimate for the 50 independent runs of the algorithms

solution where $N = 6$ (see the VBS-2 bar in Fig. 50). However, by taking a closer look at the learning dynamics of VBS one discovers that the maximum likelihood solution is obtained always for $N = 4$ (that is the VBS-1 solution in Fig. 50), while $N = 5$ is the second best-scoring solution with respect to the likelihood.

These results suggest that the most likely cluster number estimates are $N = 4$, $N = 5$ and $N = 6$. To evaluate the agreement between the data partitions produced by the different algorithms, we compared them with a baseline k-means clustering and with the sample labels discovered by PAM in (ABQ$^+$06). Table 11 summarizes the pairwise concordance of the generated solutions in terms of the κ statistics (Coh60), that is

$$\kappa = \frac{\Pr(a) - \Pr(r)}{1 - \Pr(r)},$$

where $Pr(a)$ is the relative observed agreement among two solutions while and $Pr(r)$ is the probability that agreement is due to chance. Complete agreement between the solutions is denoted by $\kappa = 1$, whereas $\kappa = 0$

Table 11: Clustering concordance evaluated by κ statistics: X's indicate when model comparison was not possible.

	CoRe				VBG			VBS		PAM
N	KM	PAM	VBS	VBG	KM	PAM	VBS	KM	PAM	KM
4	.83	.83	.65	.46	.54	.47	.36	.76	.74	.96
5	.83	X	.54	.38	.36	X	.53	.65	X	X
6	.68	X	.61	X	.23	X	.52	.49	X	X

denotes a concordance which does not exceed an agrement generated by chance.

The GMM model is not shown in Table 11 since it produced results only for $N = 6$, with a minimal overlap between its solutions and those produced by the other algorithms. Besides GMM, the model based algorithms seem to produce partitions that are quite uncorrelated with respect to those generated by the other algorithms. In particular, VBG produces groupings that are only marginally supported both by CoRe, PAM and KM, as well as by VBS. This latter model shows a fair agreement with CoRe on the hypotheses $N = 4$ and $N = 6$ and a rather weaker concordance on $N = 5$. The κ values in Table 11 show a substantial agreement between CoRe and k-means, especially on the hypotheses $N = 4$ and $N = 5$, that are those most strongly advised by CoRe. The concordance with the PAM labels is analyzed only on the $N = 4$ hypothesis since this is the cluster number estimated by (ABQ$^+$06) using PAM with model selection indices (e.g. Gap statistics, Kullback-Leibler, etc.). The KM solution shows the highest agreement with PAM labels, confirming the results in (ABQ$^+$06), while Gaussian-based algorithms seem to find different solutions with respect to both KM and PAM. As a general comment, the model-based algorithms seem unable to converge to a shared sample classification, producing quite discording dataset partitions. CoRe, on the other hand, produces solutions that are a trade-off between the concordant sample classification produced by PAM and KM, and the alternative partition discovered by VBS.

To gather a better insight into the CoRe's cluster profiles we looked at the relevance $\hat{\nu}_{il}$ of the biomarkers characterizing CoRe's sample groups.

Figure 51.a and 51.b show CoRe's feature relevance for the $N = 4$ and $N = 5$ models, respectively. The relevance plot suggests that the proliferation marker PRO does not play a key role in determining the group membership of the samples. In both hypotheses, there appears to be a single cluster that is strongly characterized by the NEU biomarker and that corresponds to the most aggressive tumor profile (ABQ$^+$06). The P53 biomarker appears to be strongly discriminant for the single C_1^4 cluster (in the $N = 4$ scenario). On the other hand, in the $N = 5$ hypothesis, the P53 covariate becomes relevant for two clusters, that are C_4^5 (corresponding to C_1^4) and C_3^5. Finally, the ER and PR markers seem to follow a correlated relevance pattern: in other words, for a given cluster, they are either both relevant or both irrelevant in determining the cluster membership.

Table 12 shows a detailed comparison between PAM and CoRe results in terms of their confusion matrices: when $N = 4$ both algorithms seem to agree on the existence of a large C_2^4-P_1 group, that CoRe characterizes with an high PR-relevance and a medium relevance of the ER marker (see Fig.51.a). This hypothesis seems to be consistent with the distribution of the PR and ER values in the Core and PAM clusters (see Fig.52 and Fig.53, respectively). In particular, the boxplots in Fig.52 and Fig.53 show that the C_2^4-P_1 group is the sole characterized by high PR values: therefore CoRe has correctly identified PR as the most discriminative marker for the cluster. On the other hand, the ER marker shows high levels both for C_2^4-P_1 and C_3^4-P_2 obtaining a lower relevance due to its smaller discriminative effect.

The results in Table 12 show that part of PAM's P_1 cluster is split by CoRe and assigned to C_3^4, that roughly corresponds to PAM's P_2: Fig.52.c and Fig.53.b confirm that the two clusters have a similar markers' distribution. Notice that C_3^4 has a very similar pattern of feature relevance with respect to C_2^4, except for a slightly higher significance of the ER covariate (see Fig.51.a). In particular, the high PR relevance is due to the fact that these two clusters have a very similar markers' distribution except for the PR covariate, that characterizes C_2^4 and C_3^4 respectively by a high and low marker expression (see the boxplots in Fig.52.b and Fig.52.c).

Table 12: Samples cross-distribution between PAM and CoRe ($N = 4$ and $N = 5$)

	CoRe ($N = 4$)				CoRe ($N = 5$)				
PAM	C_1^4	C_2^4	C_3^4	C_4^4	C_1^5	C_2^5	C_3^5	C_4^5	C_5^5
P_1	9	210	32	5	193	28	32	3	0
P_2	24	0	159	24	0	76	131	0	0
P_3	76	0	0	15	0	0	1	79	11
P_4	0	0	3	76	0	4	5	0	70

The two smaller groups identified by PAM, i.e. P_3 and P_4, roughly correspond to CoRe's C_1^4 and C_4^4 groups, respectively. CoRe clusters, however, are larger than PAM's P_3 and P_4, since they gather samples from the highly populated P_2 group (see the confusion matrix in Table 12). Interestingly, CoRe isolates sharply the small P_3 and P_4 groups in the $K = 5$ hypothesis: the results in Table 12 show a strong P_3-C_4^5 and P_4-C_5^5 overlap. Such an overlap is confirmed by the similarity of the corresponding marker distributions in Fig.53 and Fig.54. Seemingly, the addition of the cluster C_2^5 attracted those *spurious* P_2 samples that, in the $N = 4$ hypothesis, were assigned by CoRe to partition C_1^4 and C_4^4. In our opinion, this suggests the existence of a fifth cluster that is situated *in between* the P_2, P_3 and P_4 groups from PAM's solution. Already in (ABQ$^+$06), the authors pointed out the heterogeneity of the samples in P_2 with respect to the other three clusters, hypothesizing the existence of a further subdivision of P_2 in smaller subgroups. The outcome of our analysis seems to confirm this initial hypothesis. In particular, by analyzing the biomedical features associated with the clusters we can derive interesting prognostic consequences underlying the existence of a fifth tumor subgroup.

The results in (ABQ$^+$06) shows that, in general, the P_1 and P_2 samples are associated with the least aggressive tumor subgroups and are characterized by minimal-change lesions as well as hormone sensitivity. On the other hand, individuals from the P_3 and P_4 groups tend to develop an increased number of metastatic lymph nodes and the corresponding tumors are characterized by an higher proliferative rate and by more frequent oncogene suppressor alterations. More in detail, the analysis in (ABQ$^+$06)

shows that the P_4 group, characterized by high levels of HER2/NEU (see Fig.53.d), has an intermediate prognosis amongst patients non treated with hormone therapy but has the poorest response among the treated patients. Conversely, the P_3 group shows the worse event free survival (EFS) rate among the non-treated patients. The Kaplan-Meier curves (KM58) of cluster P_1 and P_2 for non-treated cases essentially overlaps (see Figure 5 in (ABQ$^+$06)), showing an event free survival rate above 80% for a five-years period, thus confirming the low aggressiveness of these two tumor subgroups compared to P_3 and P_4. However, it is worth noticing a peculiar behavior in the Kaplan-Meier curve of the P_2 cluster for patients treated with hormone therapy: while the P_1 group shows an EFS pattern that is close to the non-treated cases, P_2 shows a notably different behavior after the first 30 months. In fact, the event free survival of P_2 patients drops to values close to those obtained by individuals in the P_3 group. This behavior might confirm the existence of two tumor subtypes inside the original P_2 group: our hypothesis is that the CoRe clusters C_2^5 and C_3^5 correspond to these two tumor subtypes. In particular, the former group might be related to individuals that are responsive to hormone therapy, while the latter can possibly describe a tumor bioprofile characterized by a stronger resistance to hormone treatment.

Our conjecture is supported by the analysis of the marker profiles of the C_2^5 and C_3^5 groups (see Fig.54.b and Fig.54.c,respectively). The C_2^5 cluster is characterized by an high expression of the estrogen receptor (ER), which is considered a major predictor of response to hormone therapy (ABQ$^+$06). In addition, standing to CoRe's relevance measure in Fig.51.b, the ER covariate is the most significant feature in determining the membership of samples to cluster C_2^5. The C_3^5 group, on the other hand, is characterized by a low activity of both the ER and PR hormone receptors (see Fig.54.c): in our opinion, this aspect might have triggered the reduced response to treatment of some P_2 individuals. From the boxplots in Fig.54.c and Fig.54.d it is clear that the key difference between the profiles of cluster C_3^5 and C_4^5-P_3 lies in the expression of the P53 marker (which is confirmed by its high CoRe relevance in both clusters). In a sense, the identification of the C_3^5 cluster suggests the existence of an in-

termediate tumor subgroup that situates between the aggressive cancer form of the C_4^5-P_3 individuals and the (possibly) treatment-responsive cases in C_2^5.

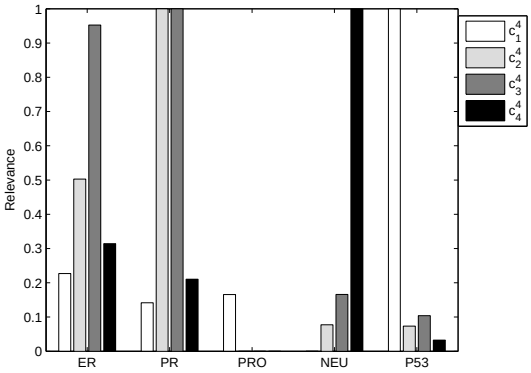
5.4 Conclusion

We presented an extension of the CoRe learning algorithm that enables repetition suppression competition on a feature-wise basis. The model described in this chapter overcomes previous limitations of the CoRe algorithm when dealing with high-dimensional data and redundant or noisy input components. In particular, we have shown how the extended CoRe model achieves automatic detection of the unknown cluster number while simultaneously ranking the sample features with respect to a given relevance measure.

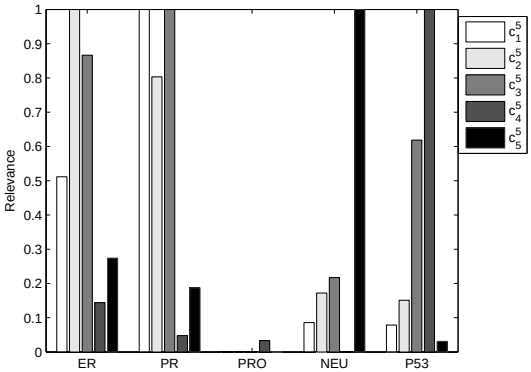
The feature-wise CoRe model has been applied to the analysis of several freely available datasets from gene expression data: the results show that CoRe clustering is able to estimate the latent structure of an high dimensional dataset in a completely unsupervised way, while developing a feature ranking that is consistent with the state-of-the-art knowledge regarding the functional role of gene expression in disease realization. In particular, the experimental results suggest that feature-wise CoRe has performances that are comparable with that of computationally intensive wrapper approaches, e.g. based on ensemble learning and resampling (MTMG03; FMR05; DMBD04), while retaining the low computational complexity of an incremental filter model. We advise that this CoRe property can be exploited for the development of a tool addressing interactive gene data exploration, that can provide the user with incremental estimates of features' relevance as learning proceeds. Within this context, it would be interesting to study further the adaptivity of our model by analyzing the effect of the introduction of fresh data on a consolidated training base.

In addition to DNA microarray data, we have studied unsupervised cluster number estimation on small dimensional breast cancer dataset, comparing the performance of feature-wise CoRe with Bayesian model-

based clustering. The experimental results showed how the CoRe algorithm can be used to unsupervisedly explore biomedical data, estimating the number of the clusters in the dataset and providing a relevance for the samples' covariates that can be used to identify significant markers of the discovered bio-profiles. The comparison with the Bayesian models has highlighted how their performance can be seriously affected by the nature of the dataset, preventing them to reliably estimate the data clusters. With respect to the breast cancer case study, CoRe's analysis suggests the existence of a fifth tumor subgroup. Our hypothesis, that shall by validated by clinical studies, is that such sub-group might differentiate two breast cancer subtypes: the former is characterized by a high expression of the estrogen receptors which should determine a low aggressiveness and a good response to hormonal therapy; the latter should differentiate a more aggressive tumoral type that is less responsive to treatment and is characterized by an event free survival profile close to that of individuals with high P53 expression.

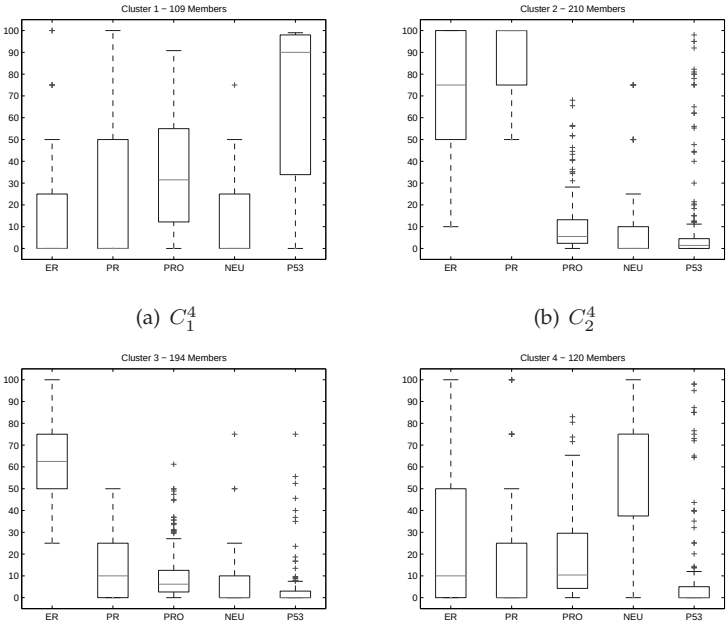


(a)



(b)

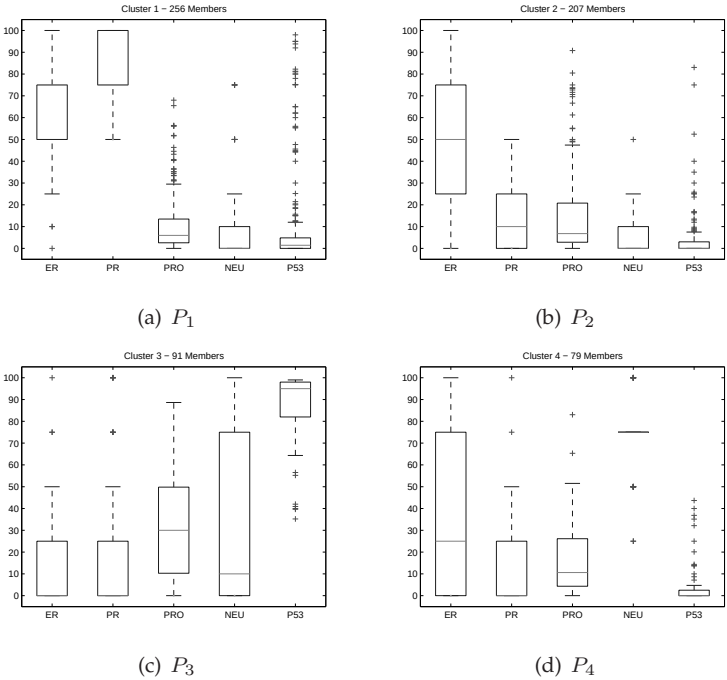
Figure 51: CoRe relevance factors of the 5 features for the (a) 4 clusters and (b) 5 clusters scenario



(a) C_1^4 (b) C_2^4

(c) C_3^4 (d) C_4^4

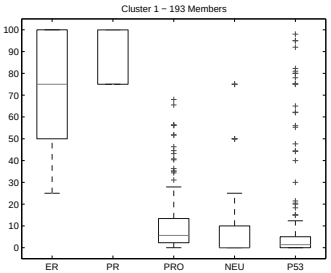
Figure 52: Boxplots of CoRe cluster profiles for the $N = 4$ solution



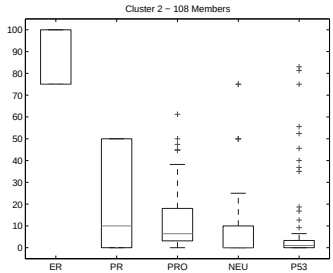
(a) P_1 (b) P_2

(c) P_3 (d) P_4

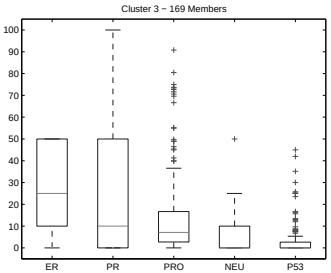
Figure 53: Boxplots of PAM cluster profiles



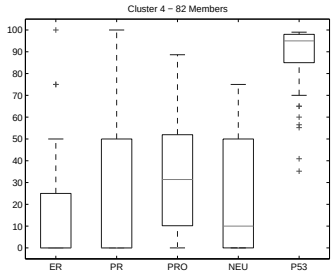
(a) C_1^5



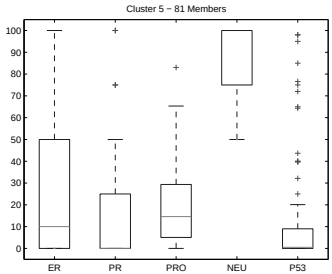
(b) C_2^5



(c) C_3^5



(d) C_4^5



(e) C_5^5

Figure 54: Boxplots of CoRe cluster profiles for the $N = 5$ solution

Part II

Semantic Annotation by Hierarchical Latent Aspect Discovery

Chapter 6

Image Processing Background

6.1 Introduction

So far, the dissertation focused on the analysis of the theoretical properties and performance of the CoRe learning algorithm. In this part, we move forward by applying CoRe learning within a machine vision model that addresses the issue of attaching semantic labels to meaningful image portions, in the attempt to reduce the semantic gap between the low level pixel representation and the high level concepts represented in the scene. However, before proceeding with the formalization of the machine vision model (in the next chapter), we need to lay down the foundations of image representation and to review the recent relevant works in image content understanding.

The first part of this chapter (Section 6.2) discusses the most relevant approaches to low level visual information representation. First, in Section 6.2.1, we describe how the single relevant bits of the perceptual information can be represented by means of global and local descriptors: the former features mostly rely on color, edge and texture statistics computed on the whole image, while the latter describe the information content of a portion of the whole scene. The use of localized representation

approaches rises an additional challenge, that is how to identify significant and informative portions of the images where local descriptors are computed: in Section 6.2.2, we review the most relevant contributions with this respect, focusing in particular on the popular *affine region detectors* (MTS$^+$05). Finally, in Section 6.2.3 we discuss how the single bits of visual information extracted from a picture, and represented either by global or local descriptors, can be aggregated to describe the whole image.

Once that a suitable description for the low level visual information has been obtained, it can be processed to discover higher level semantic concepts describing the image content. The second part of the chapter, Section 6.3, reviews approaches dealing with the identification of semantically meaningful entities in cluttered scenes, i.e. *object recognition*, as well as with the inference of category labels summarizing the scene information, i.e. *scene recognition*. Particular attention is paid in Section 6.3.1 to *latent aspect* models: these approaches allows to discover an intermediate representation (known as *topic*, *aspect* or *theme*) of the visual information that has proved to be extremely helpful in bridging the gap between the low level image representation and the high level semantic concepts. Section 6.3.2, on the other hand, describes structured part-based and hierarchical approaches: instead of searching a latent aspect summarization of the visual content, they fit a, more or less, flexible model of the object or scene to be identified, that is typically based on a generative probabilistic formulation of the sought items.

The third part of the chapter, i.e. Section 6.4, concludes the background analysis by reviewing *automatic image annotation* models. These systems, typically process collections of images and related textual information in order to learn the association between visual content and significant caption words. In a sense, they can be considered as a weak generalization of the object and scene recognition models, since the textual annotations that they learn to correlate to images can be considered as category labels in a object or scene recognition scenario. Notice, however, that annotations are not constrained to refer strictly to objects categories or scene descriptions, but can contain any information related to

the pictures in the collection (i.e. localization information, patient name in medical images, etc.).

6.2 Visual Content Representation

The description of the the visual content of an image is fundamental prerequisite for performing any image analysis task. In general, the choice of the visual descriptors is deeply influenced by the final objective of the image processing. Much of the early approaches to image content description rely on a *global* characterization of the visual features, that is to say that the signature of a picture is computed based on information regarding the distribution of given features in the whole scene depicted in the image. This representation has the undoubtable advantage of conciseness, since the whole image content can be summarized in a relatively low dimensional pattern which, in general, does not necessitates consistent computational efforts in order to be calculated. However, such an approach inherently discards much information regarding, for instance, the single objects depicted in the image and has a weak tolerance to variation in the image content, that is to say that slight alterations (e.g. luminance, occlusion, etc.) to the depicted scene might result in strong variations in image representation. As a result, global approaches are more effective for high-level scene categorization tasks with few classes, but they are rather ineffective for object recognition. Lately, *local* feature descriptors has started gathering increasing attention in the machine vision community due to their superior ability in representing fine-grained visual knowledge. These approaches describe the information content restricted to a localized portion of the image. This aspect immediately rises a fundamental questions, that is how to determine the portions of the image as to obtain the most informative local features. This latter issue has been addressed by several visual feature detectors, performing region of interest identification at various image scales. Section 6.2.2 reviews the main feature detectors in the literature, while 6.2.1 overviews global and local image descriptors. Finally, Section 6.2.3 shows how such feature detectors and descriptors can be used and combined to generate effective

representations of the visual content.

6.2.1 Feature Descriptors

Global Descriptors The first attempts to capture the visual content exploited the global distribution of pixel intensities and color information in the image. Within this context, images are typically represented by means of histograms describing color distribution in one the several existing color spaces such as RGB, HSV, YUV, CIE-LUV and CIE-LAB (Hun98). Each of these color spaces has different advantages and drawbacks: RGB and YUV are common color encodings in image and video hardware but are strongly influenced by intensity variations. CIE-LAB and CIE-LUV, on the other hand, are good choices for measuring the distance between two colors because of their perceptual uniformity, although the transformation from the RGB space is computationally intensive.

Swain and Ballard (SB91) have probably been the first to formalize the use of color histograms to index images. Later, color moments have started to be used to obtain compact representations of the color distribution of an image. Color coherence and color autocorrelogram (HKM$^+$97; HKM$^+$99) have been developed to take into account also the spatial distribution of colors in an image, although experimental results (MZ99) suggest that these representations do not perform substantially better than basic color histograms, which require lower computational efforts.

Shape features such as texture orientations have been used in the classification of natural versus man-made scenes: in (GP94), for instance, a set of directional filters is convolved with the images over multiple scales in order to extract histograms of perceptually relevant local orientations. Gabor filters have been widely used to characterize edge orientations in textures as well as in natural/man-made scenes; typically a multiresolution dictionary is built out of a set of Gabor wavelets that are convolved with the image and the feature descriptor is constructed based on the mean and standard deviations of the magnitude of the wavelet transform (MM96). The PicSom model (BLO02), on the other hand, describes images by means of an eight-bins histogram of edge orientations

computed by Sobel operator as well as by 128-dimensional magnitude vectors computed from a Fast Fourier Transform (FFT) of the Sobel filtered images, converted in log-polar coordinates to enforce translation and scale invariance. Different types of features are often combined to enhance the discriminative power of the visual content description: the work in (VFJHJ01) is a relevant example of such hybrid approaches where diverse features are used to separate image classes at different levels of a taxonomy. For instance, color histograms are used at the top of the hierarchy to tell outdoor images from indoor pictures. The former class is then partitioned into a city and a landscape class based on global shape features derived from local edge orientations. Finally the landscape class is further sub-divided into sun, forest and mountain scenes based on color histograms, coherence vectors and spatial moments. Such an approach has the advantage of breaking an articulated multi-group classification task into several two-way classification problems, which are best solved by a classifier that relies only on global features.

Texture is another fundamental aspect characterizing visual content that has been exploited to build global feature descriptors. Texture characterizes regions instead of single pixels and has been widely exploited by the image retrieval community. One of the first attempts to characterize texture is due to Tamura (TMY78), that provided a description based on six features derived from human visual perception, i.e. coarseness, contrast, directionality, line-likeness, regularity, and roughness. The first three features are those that encountered more success and have been widely used to represent image content. In (Har79), on the other hand, it has been proposed to describe textures using moments computed from grey-level co-occurrence matrices (GLCM) derived from the image at given pixels. As discussed previously, Gabor filters have been used also to characterize textures in terms of mean and variance of their outputs in feature space (Tur86). See (HR04) for an experimental comparison of several texture descriptors in an image retrieval scenario.

Recently, Oliva and Torralba (OT01) proposed a low dimensional global characterization of images for scene classification tasks that founds on the concept of *Spatial Envelope*. In this model, the dominant spatial structure

of a scene is described by five perceptual dimensions, namely naturalness, openness, roughness, expansion and ruggedness, that are estimated using spectral and coarsely localized information.

So far, we have described global approaches that generate a visual content descriptor based on the information contained in all the pixels in the image. Local representation approaches, on the other hand, model the pictorial content by dividing the image into regions on which individual features are computed, building the final image descriptor by aggregating the computed local features. These approaches have started gathering increasing attention in the latter years since they provide robustness to scene or object occlusion as well as a richer description of the visual content. Indeed, the simplest form of local descriptor is a vector of pixel intensities in the region of interest: this representation allows to compare image patches using cross-correlation but has an high computational complexity given by the high dimensionality of the representation. Moreover, these descriptors lack of invariance with respect to affine transformations which is a key property when targeting at generalizable and flexible representations of the visual content. Therefore the use of pixel vectors is mostly restricted to finding correspondences between images. In general, the sought properties for local descriptors are robustness to photometric and geometrical variations, in order to obtain representations that are discriminative while being as much invariant as possible to scene illumination, rotation as well as to object occlusion.

Local Descriptors The majority of *local descriptors* can be interpreted as multi-scale filters applied to specific regions of an image: the response of such filters represents the information content of the region of interest. Clearly, the *Gabor filters* introduced previously in the context of global image representation are natural candidates to describe parts of an image. The Gabor transform is often used to represent textures, although a large number of Gabor filters is required to capture small changes in frequency and orientation of the image patch, resulting in a consistent computational load and a larger dimensionality of the descriptor. The intensity domain *spin image* (LSP03) is an histogram-based local descriptor pro-

posed in the context of texture classification. Spin images represent visual patches by means of two dimensional histograms of space and intensity values and have been shown to outperform Gabor filters in texture classification tasks (LSP03). However, texture descriptors do not transpose their performance when applied to generic images, due to the lack of repeated patterns and statistical dependencies that characterize textures but do not apply to more generic visual contexts (MS05).

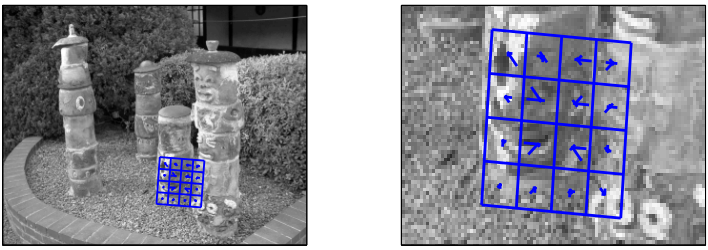
The class of *differential descriptors* relates closely to Gabor filters. These descriptors represent geometric features of an image patch by means of a set of partial derivatives (up to a given order K) with respect to the space coordinates, that is called a *local jet* (KvD87). Essentially the partial derivatives are obtained by convolution with a receptive field profile, typically given by Gaussian derivatives (Bau00). In (SZ02) it is proposes to induce a coordinate change in the Gaussian derivative responses by means of a linear filter bank; invariance to intensity and rotation is achieved by combining the single components produced by the local jet (FtHRKV94). Notable examples of differential descriptors are the popular *steerable filters* (FA91), that consist of linear combinations of basis filters that can be easily steered to any orientation. This model is shown to achieve rotation invariance when Gaussian derivatives are steered in the direction of the gradient.

Distribution-based descriptors represent local patches by means of histograms describing different properties of the local neighborhood. The simplest distribution-based representation is the classical histogram of pixel intensities restricted to the given local patch. The spin image descriptors introduced above, provide a richer information content by accompanying intensity values with spatial information in a two-dimensional histogram. An attempt to devise distribution-based descriptors that are robust with respect to intensity variations is the *non-parametric local transform* by (ZW94). Rather than representing the intensity values, this approach codes the relative ordering of local pixel values by means of a *rank transform*; then, a second non-parametric transform, called *census transform*, is used to encode the neighborhood of the pixels in the patch into a bit string. This approach has originally been developed to address

the correspondence problem and requires high-dimensional descriptors in order to reliably represent generic image collections: however, it can be effectively applied to texture representation (MS05). On the other hand, *geometric histogram* (ATRB95) and *shape context* (BMP02) descriptors have a similar approach to visual content representation that relies on the computation of histograms describing the edge distribution within local patches. Therefore, these descriptors can be successfully exploited to represent pictures where edges are reliable and discriminative features, e.g. handwritten data and drawings.

The most popular histogram-based descriptor is undoubtedly the *Scale Invariant Feature Transform* (SIFT) by Lowe (Low99). To compute a SIFT descriptor at a given scale σ, it is first needed to smooth the image using a Gaussian with spread σ. Then, for each image keypoint (identified previously by a suitable interest point extraction algorithm), the algorithm computes the gradient magnitude and orientation of the smoothed image. Finally, a three dimensional histogram of magnitude orientations is computed in a 4×4 neighborhood of the keypoint (see Fig. 55). Gradient angle is computed along eight directions, resulting in a $4 \times 4 \times 8 = 128$ dimensional descriptor. Notice that the quantization of both gradient locations and orientations produces a representation that is robust with respect to small geometric distortions. Illumination invariance is obtained by L_2 normalization of the SIFT vector. The SIFT algorithm provides also a scale invariant region detector that first blurs the image by convolving it with Gaussian filters at different scales σ_i and, then, extracts interest points at the peaks of the differences between successive Gaussian-smoothed images.

Several variants to the original SIFT descriptor have been proposed recently: (KS04), for instance, applies Principal Components Analysis (PCA) to the SIFT representation, obtaining more distinctive PCA-SIFT descriptors that are robust with respect to image deformations and offer the compactness of a principal component representation. The *Gradient Location and Orientation Histogram* (GLOH) (MS05), on the other hand, considers more spatial regions than the standard SIFT descriptor, obtaining a superior distinctiveness at the price of a higher computational load



(a) (b)

Figure 55: SIFT descriptor computed by VLFeat toolbox (Ved07) on an exemplar image (a); a close up (b) on the descriptor reveals the 4×4 neighborhood and the magnitude orientations.

and histogram dimensionality which, however, is reduced (typically to 64) by means of PCA. The *Speeded Up Robust Features* (SURF) (BETG08) descriptor uses a different representation of the information content based on the distribution of Haar-wavelet responses within the interest point neighborhood. SURF computes a 4-dimensional vector for each of the 4×4 neighboring subregions of the interest point, resulting in a 64-dimensional descriptor. Each subregion vector contains the sum of the responses of the Haar-wavelets in the horizontal and vertical direction, as well as the magnitude along the two directions. The wavelet responses are, themselves, invariant to illumination offsets, while invariance to contrast is achieved by descriptor normalization. The SURF computation and matching process uses only integral images, resulting in computation time that is about an halve of that required for SIFT calculation (BETG08).

There exist other distribution-based approaches that do not rely on an histogram characterization of the local patch: (GMU96), for instance, uses invariant moments to characterize shape and intensity distribution within the region of interest. Moments of higher degree are in general sensitive to geometric and photometric distortions, hence (GMU96) computes a mix of different type of moments to keep their order (and sensitivity) low. In general, invariant moments require a consistent amount

of information to generate discriminative descriptors; therefore they are more suitable for application to color images representation (MS05).

The local descriptors introduced so far represent a choice of the most prominent contributions in a lively scientific field; for an in depth analysis of the state of the art in this area refer to (MS05), which presents a notable experimental comparison of several local descriptors. Additional experimentations with the latest local descriptors can be found in (BETG08). Overall, the experimental analysis show that there is no unique descriptor which performs better than its competitors on generic image collections: the family of SIFT-like descriptors has in general the best performance and, in between the family components, the SURF descriptor seems to be the best performer. However the SIFT descriptor still has an implicit role of de-facto standard for benchmarking new local approaches to image understanding and retrieval.

A localized representation of the visual content requires tools for determining the regions or points of interest within an image. For instance, previously we have mentioned that the SIFT method provides a means for determining the picture keypoints (and similarly do GLOH and SURF): in the next section, we review the principal contributions to interest regions and interest points identification.

6.2.2 Feature Detectors

So far, we have focused our overview on how to represents the relevant visual features enclosed in an image, however little has been said concerning how to localize such features. In order to identify them, we need first to define what is an interesting and informative feature. In its wider meaning a visual feature is a characterizing portion of an image: within this definition, we might enclose several visual structures such as segments, blobs, edges and corners. In the following, we will review the main feature detectors in the literature, grouping them based on the type of image feature they seek to identify.

Before delving into the discussion, we need to identify what are the desirable properties of an efficient detector. The first and most funda-

mental property of a feature detector is indeed *repeatability*, that is the ability to detect the same feature in different images as well as in different portions of the same scene. In order to enforce repeatability, a feature identification algorithm must be able to cope with photometric and geometrical transformations of the image since, by this means, it ensures that given an image and a slightly altered version of the same scene, the algorithm detects the same interest points in both pictures. The typical transformations that are encountered when dealing with visual content are

Photometric variations Changes in brightness and luminance should not impair the detection of a visual features.

Translation Interest regions tend to change location in different scenes; feature detector must be able to locate them irrespectively of their position in the image.

Rotation The rotation of a single object or of the whole scene influences the angle at which the interest points are processed; detectors should be able to identify a keypoint irrespectively of their absolute and relative orientation.

Scaling Image resizing or a change in camera zoom produce scaled version of the same scene and a change in the absolute size of the interest region; detectors must be able to identify the correct scale where the feature has to be detected.

Affine Transformations These changes are the most difficult to address and are typically generated by a change in the viewpoint angle that results in a contemporaneous transformation in position, angle, scale and shape of the interest region. Affine transformations are seen as generalization of the scale change to non-isotropic operations: usually feature detectors cope with it by using the adaptive shape matching procedure (LG97).

Finally, since feature detection is a basic low-level step of the image processing chain, it needs to be an efficient process with a limited computational complexity.

Segment Detectors In a sense, segmentation algorithms have been the predecessors of the modern local feature detectors. These algorithms seeks spatially contiguous pixel groups, i.e. the *segments*, characterized by color homogeneity: typically, detected image segments are represented by means of color, texture and shape histograms. For instance, second generation image understanding and retrieval models, such as (CBGM02; GGR01), as well as image auto-annotation system (Bar01; FML04) mostly relied on an image segments' representation. Blobword (CBGM02), for instance, represents each image pixel as a six dimensional vector comprising both color and texture information: pixel data is then used to fit a Gaussian Mixture Model using Expectation Maximization. At the end of the learning phase, the mixture components identify the image segments (blobs) that are used to represent the scene. Meanshift (CM99), on the other hand, is a non-parametric density estimator that has found wide application in machine vision due to its ability in reliably, and repeatedly, estimating image segments out of pure color intensity information. In (FOPA04), for instance, is presented an object recognition model that exploits Meanshift to detect interest regions, represented by vectors of color and texture moments, that are later used to train an object classification module. Barnard et al (BDG$^+$03) exploit Meanshift segmentation together with a machine translation model to attach text to relevant image segments. The same paper compares Meanshift's performance with that of another popular segmentation algorithm, that is the Normalized Cuts (NCuts)(SM00). NCuts has been particularly successful in machine vision community since it allows to smoothly control the degree of image oversegmentation, i.e. the tendency to generate more segments than the actual number, and since it produces global solutions where segments have a large chance to be capturing whole objects. Carbonetto et al (CdFB04), for instance, used NCuts within an image annotation framework, while (RFE$^+$06) exploited it to generate multiple image segmentations within a latent topic discovery model.

Blob Detectors Overall, segmentation algorithms do not show a clear attitude towards repeatability, since the extent of the image segments of-

ten varies not only with photometric changes, but also as the result of different initializations of the algorithm. Moreover, pure segmentation approaches are scarcely robust with respect to occlusion since the detected surfaces usually occupy consistent and contiguous portions of the image, hence the same segment has an high chance of being, at least partially, occluded in diverse scene arrangements. *Blob detection* (known also as *interest region detection*) can be seen as a segmentation approach taken on a smaller scale, where the regions of interest are image patches that are either brighter or darker than the surrounding pixels and that suggest the presence of object parts. Two approaches are commonly used to detect blobs: the former is based on the difference of image derivatives of different order, while the latter exploits the identification of local intensity extrema.

A popular algorithm relating to the former approach, i.e. the *Difference of Gaussians* (DoG) (Low04), has already been discussed when introducing the SIFT descriptor. DoG generates a sequence of blurred images by convolving the original gray-scale picture with Gaussian functions at different scales, obtaining a pyramidal representation where increasingly larger scales correspond to stronger suppression of high frequency spatial information. By subtracting blurred images at different scales, DoG obtains two results: (i) it implements a band-pass filter that enhances high frequency details of the original image, e.g. allowing to detect edges; (ii) it identifies areas of uniform intensity, i.e. blobs, since the convolution of a difference of two Gaussian with a uniform signal generates a null response. Besides the multiscale representation, DoG builds also a multi space pyramid by sub-sampling the image at difference octaves: this process allows to detect interest regions with high variation of scales and positions, obtaining an extremely stable characterization of the image. The resulting detector is scale, illumination, and orientation invariant.

The DoG operator can be seen as an approximation of the second derivative of a Gaussian smoothed image, known as the *Laplacian of Gaussian* (LoG) (Lin98). This operator produces strong responses (strong negative responses) for dark blobs (bright blobs, respectively) matching the squared root of the scale. The LoG operator has been the first to adopt

a scale-space representation in order to bypass the dependency between region size and scale: in particular, blobs are identified by means of the scale-normalized Laplacian (Lin98), that allows to detect those points that are simultaneous extrema both in scale and space. This Laplacian scale selection mechanism has been a fundamental building block for constructing advanced feature detectors, such as the Harris-affine approach.

The *Determinant of the Hessian* (DoH) operator is a blob detector with automatic scale selection which also responds to ridges (Lin98). This detector is based on the Hessian matrix, that describes the neighborhood of an image point in terms of the second order derivatives computed by convolution with Gaussian kernels: the resulting derivatives are integrated in the neighborhood by Gaussian smoothing. Local maxima of the determinant of the Hessian identify image blobs. In order to make the detector scale invariant, (Lin98) proposes to select the scale that maximizes the Hessian match with the structure of the keypoint neighborhood. Alternatively, (MS01) exploits the determinant of the Hessian to obtain spatial localization while the Laplacian operator is used for scale selection, yielding the so-called *Hessian-Laplace* detector.

The *maximally stable extremum regions* (MSER) (MCUP02) is a popular blob detector seeking image regions that maintain a stable shape over a range of thresholds of the image intensities. In practice, MSER considers a sequence of possible intensity thresholds for a given gray-scale image: for each threshold, MSER builds a binary mask of the pixels that exceed the given level. The selected blobs correspond to a subset of the regions detected at all thresholds whose shape remains stable across a range of thresholds. The resulting regions are invariant to transformations of the image intensity and produce a multi-scale image representation without resorting to Gaussian smoothing.

Corner/Interest Point Detectors Detectors formerly addressing corner detection have recently evolved to embrace the identification of a larger class of characteristic image points, known has *interest points*. Typically, interest points identify image positions characterized by local intensity extrema, such as line endings or points where line curvature is locally

maximal.

The *Harris detector* identifies interest points by a measure of saliency, named *cornerness*, defined in terms of the determinant and trace of the second order matrix of local image gradients, known as the *scatter matrix*. Local image derivatives are obtained by convolution with Gaussian kernels and successive integration by weighted average using Gaussian smoothing. The second moment matrix characterizes the signal autocorrelation on the neighborhood of a point. The eigenvalues of the scatter describe the two principal directions of curvature in the neighborhood of a point, hence the maxima of the cornerness identifies those keypoints where the curvature is significative in both orientations (i.e. a corner). The original Harris detector determines the region of interest around the keypoint with a scale defined by the Gaussian used in the convolution, hence it is not scale invariant. In (DSH00) it has been extended to multi-scale detection, where keypoints are identified by local maxima of the cornerness over multiple scales; *Harris-Laplace* (MS01), on the other hand, applies the LoG operator to the Harris interest points detected at multiple scales in order to locate the right resolution for the interest region.

Smallest Univalue Segment Assimilating Nucleus (SUSAN) (SB97) is a corner detection algorithm that finds circular masks such that the similarity between the intensity of the mask center and its local area is maximized. In presence of edges and, in particular, of corners the size of SUSAN notably reduces. Hence, by seeking the smallest circular masks this detector is able to identify image corners without resorting to local gradient computations. The *Features from Accelerated Segment Test* (FAST) (RD05) operator takes a similar approach, defining a circular region around the candidate keypoint and testing whether there exist at least a number n of contiguous neighboring points that have a substantially higher (respectively lower) pixel intensity with respect to the mask center. This algorithm is reported to produce stable features at a small computational complexity.

Affine Invariant Point Detectors So far, we have reviewed local feature detectors that can deal with photometric changes and that are invariant

to translations, rotations and uniform scale changes. However, images are often subject to more articulated geometric distortions: in particular, a robust visual content detector should be able to address perspective modifications that result in affine transformations of the local patch. Analytically, affine invariance can be obtained as a generalization of scale invariance, that is to say that full scale matrices Σ^{-1} are used in place of uniform scale parameters σ. Hence, affine invariant interest points are obtained by applying affine shape adaptation (LG97) either to the smoothing kernel or the the image patch.

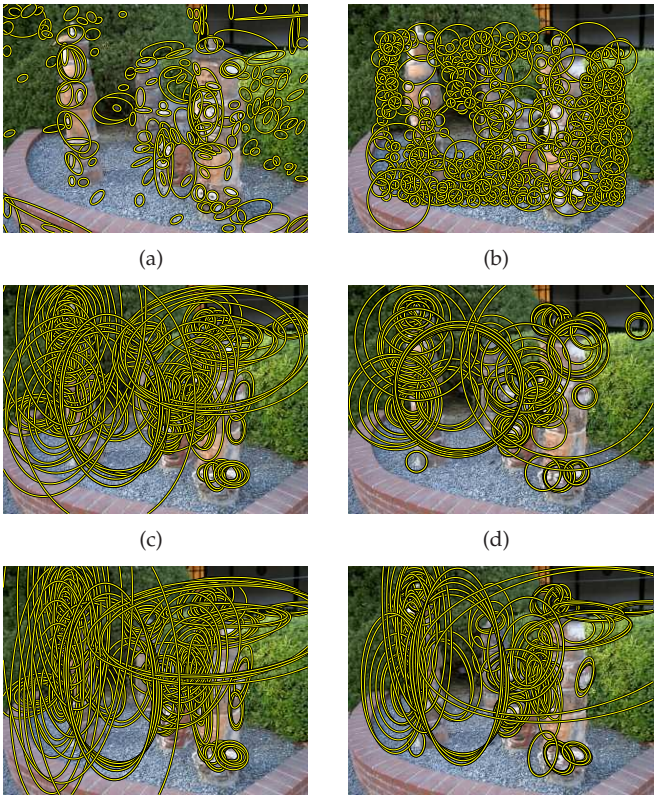
Most of the interest point and blob detectors described above can be adapted to achieve affine invariance. Mikolajczyk and Schmid (MS04), for instance, extended both the Hessian-Laplace and the Harris-Laplace detectors to address non isotropic transformations, proposing the *Hessian-Affine* and *Harris-Affine* operators. Initially, spatial interest points' localization and scale estimates are obtained using the original algorithms; then, affine shape adaptation (LG97) is applied to normalize the keypoints' neighborhood: as a result, a new integration and differentiation scale, as well as an updated interest point localization are generated. The affine shape adaptation process is thus iterated until a stopping criterion is met.

Tuytelaars and Van Gool (TG04b) propose two alternative methods for obtaining affine invariant local regions. The former, known as *Edge-Based Region Detector* (EBR), exploits corners and edges information (obtained with Harris and Sobel operators, respectively) to identify interest regions. Starting from a corner and a set of recurrent neighboring edges, EBR constructs a parallelogram family bounded by the Harris corner and the points in two neighboring edges: the final interest region is determined by selecting the parallelogram associated to the extremum of an intensity function defined over the candidate regions. The EBR approach relies strongly on the quality of the edge information: hence, the authors proposed an alternative model, called *Intensity Extrema-Based Region Detector* (IBR) (TG04b), that relies purely on intensity information. The IBR detector starts by identifying local intensity extrema: given such points, it analyzes the intensity profiles along radial lines departing from the local

extrema. Intensity profiles are measured by a function describing the photometric transformation with respect to the keypoints: the maxima of this function identify positions of sudden change of intensity and are used to delineate regions of arbitrary shape enclosing the local maxima. These regions are then replaced by ellipses having the same shape moments up to the second order, resulting in an affine covariant construction.

The *Salient Region* detector has been first introduced as an isotropic feature detector but has later been extended to affine invariance (KZB04). The idea underlying this work is to evaluate the entropy of local image attributes (e.g. intensity or color), computed within an elliptical region of each image pixel. First the multi-scale extrema over the whole image are selected as candidate salient regions. Then the scales producing a peak in the entropy are retained as salient scales. Region saliency is thus calculated as the product of the entropy with the derivative of the attribute's probability density function with respect to scale. Finally, the P top-ranking regions with respect to the saliency measure are selected as interest patches. By seeking high entropy regions, this algorithm implicitly considers as salient those regions that exhibit unpredictability with respect to the expected behavior of given image attributes.

An experimental comparison of several affine invariant and covariant feature detectors is presented in (MTS$^+$05) and in Dorko's thesis (Dor06). The analysis suggests that it does not exist one detector that outperforms the competitors for all tasks and under all types of visual transformation. In particular, MSER obtains the best results in average, together with the Hessian-Affine, performing particularly well on images containing homogenous regions with sharp boundaries. Salient regions obtained low repeatability scores but they perform best in the context of object recognition (KZB04). Overall Hessian and Harris-based detectors produced the largest number of interest points ($\backsim$ 1500 for 800×640 images), while MSER and Salient Regions produced about one third the number of keypoints with respect to Hessian and Harris operators (see Fig. 56 for an example of interest point detection): in particular, the ability to detect a larger number of interest points might enforce the robustness of object recognition since classification is less affected by occlusion.



(a) (b) (c) (d) (e) (f)

Figure 56: Interest point detectors computed using the binaries provided by the Visual Geometry Group, Oxford University[1]: (a) MSER (b) salient regions, (c) Harris Laplace, (d) Hessian Laplace, (e) Harris affine and (f) Hessian Affine detectors. Notice how Harris and Hessian detectors return a larger number of interest points with respect to blob detectors, i.e. MSER and salient regions. Also, thresholds of Harris and Hessian detectors have been adjusted to return a smaller number of interest points to favor visualization.

In general, it is hard to determine which feature detector is best suited for a given task. Often different types of detectors are mixed to obtain more information concerning the visual content of an image: in (SRE+05), for instance, MSER is used together with Harris-Affine detectors, since the former can identify blob-like regions while the latter detects corner regions. In (NJT06), on the other hand, the authors suggest that *uniform sampling* obtains better results in image categorization than interest point detectors. Moreover, Bosch et al (BZM08) show that patch sampling achieves higher classification performances than Harris-Affine also in scene categorization. Dense sampling works by placing a grid on the top of the image, randomly picking a number of interest regions from the multiscale grid patches. In our opinion, sampling achieves a good categorization performance when used in combination with a *bag-of-words* image representation because the vector quantization step produces stabler results when applied to densely sampled regions instead of sparse interest patches. Interest regions detectors, in fact, tend to produce more irregular data distributions characterized by a large number of low dimensional and highly informative clusters with many outliers producing consistent distortion. On the other hand, random sampling tends to generate denser clusters (due to the redundancy of the sampling of most frequent features) where outliers have a smaller distortive effect on the quality of vector quantization.

6.2.3 Image Representation

Global approaches to visual content description offer a straightforward representation of the whole image content. For instance, global color and texture histograms can be used to compare and rank different images in a collection as in the earlier CBIR model such as QBIC (FSN+95), Virage (GJ97)) and Candid (KCH95).

Local descriptors, on the other hand, pose the additional challenge of devising a suitable representation for the whole image content: in fact, by themselves, local descriptors provide only information on a portion of the portrayed scene. In the following, we overview some of the most

relevant models for image representation based on local content.

One of the earliest models in the literature is based on image segments information: in particular, (GGR01) adopts a probabilistic image representation relying on the color, texture and position information of the detected segments. Each image is represented as a mixture of Gaussians, where each component describes the local distribution of the attributes in a segment. Images are then compared based on the Kullback-Leibler distance (GGR01). The Blobword system takes a slightly different approach: similarly to (GGR01) it fits a Gaussian Mixture to each image in order to determine the segments. Then each detected segment is represented as an histogram of color distributions and texture moments and the whole image is represented by a collection of segments' histograms. The matching between images is performed on a region level by confronting the segments' histograms using the Mahalanobis distance.

The representation of images based on local patches, either densely sampled or sparsely identified by interest point detectors, typically goes through a first normalization step that resembles the word stemming phase in text-based systems. In particular, the local descriptors collected for the whole image collection, or for a consistent portion of it, are feed to an unsupervised vector quantization algorithm in order to determine a set of K local patch classes, corresponding to the K codewords identified by the vector quantizer. The most popular algorithm for visual word quantization is k-means, although some authors have explored more sophisticated vector quantization approaches: (JT05), for instance, pointed out k-means' tendency to allocate codevectors in denser regions of the input space, producing a perceptually distorted representation of the descriptors, and proposed to address this issue by a quadratic Meanshift-like algorithm. In (LMS06) the authors introduce an optimized agglomerative clustering algorithm for automatically estimating the size of the codebook, while (MLS06) exploits hierarchical clustering to provide a multilevel image representation. The practical application of these advanced clustering models is limited by their run-time complexity and the use of k-means is, in general, preferred.

The *bag-of-words* (BOW) representation (CDF+04), also known as *bag-*

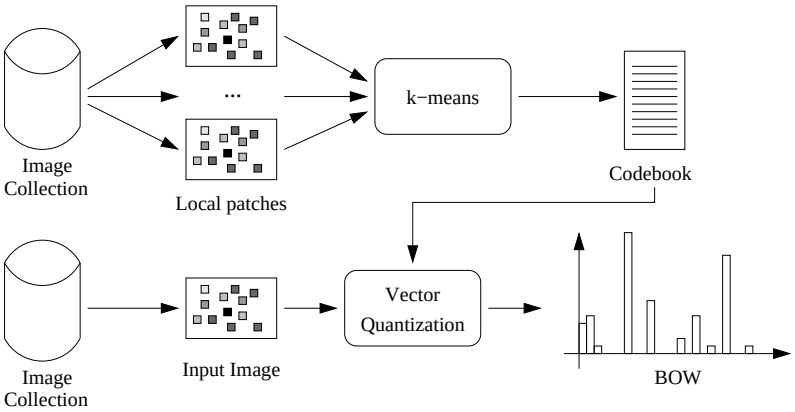


Figure 57: The bag-of-words (BOW) generation process: interest patches are extracted for the images in the collection; then, a codebook is generated for a set of *training* images (upper part of the graph). Each image can thus be represented by a word occurrence histogram, i.e. the BOW, computed using the previously extracted codebook (lower part of the graph).

of-visterms (BOV) and *bag-of-keypoints*, exploits the vocabulary of visual words (visterms) built by the vector quantizer to compute an histogram of codevectors' occurrences for each image in the collection (SZ03). BOV takes the local patches associated with each image and substitutes each local descriptor with the closest codevector with respect to a given distance measure, typically Euclidean or Mahalanobis. Then, it builds an histogram that counts the number of occurrences of each codeword in the image (see Fig.57 for a graphical description of the process). The resulting representation describes an image by its local patches distribution, so that pictures can be confronted by means of any bin-to-bin histogram measure. As with any vector quantization task, this process introduces noise in the image representation. However it has also the positive effect of *standardizing* the local patches description in such a way that the common visual root of similar descriptors is hopefully brought out, likewise with word stemming in text retrieval.

The BOV representation has been successfully applied to object recognition, scene classification and image retrieval using histograms as image

signatures that are feed to k-Nearest Neighbors (k-NN) (GD05; DKN05), Naive Bayes (CDF$^+$04; DKN05) and Support Vector classifiers (CDF$^+$04; NJT06). Moreover, the BOV representation is the basic building block for latent topic models (SRE$^+$05; FFP05; FFFPZ05) and generative approaches (MLS06; FFFP07) to visual content discovery (see Section 6.3 for a broader discussion).

The bag-of-words representation assumes that any information concerning the arrangement of local patches as well as the spatial relationships between visterms can be neglected. In other words, the probability of finding a visterm w_i given and image I is independent from the other visterms that are possibly present in the scene. This BOV description can be generated with a small computational overhead and is inherently robust with respect to partial occlusion, since the lack of a few local patches does not produces a substantial change in the histogram distribution. Moreover, given invariant interest point detectors, BOW is robust to scene translation, scaling and rotation since the spatial relationships between patches are ignored.

The bag-of-word assumption comes from the text retrieval community, where word ordering is assumed to contain a negligible informative content regarding the document semantics. However, in the visual domain this assumption contains an extreme simplification, since the semantics of visual entities, e.g. objects, is mostly determined by the way their constituent parts, possibly represented by local patches, relate spatially. Moreover, the bag-of-words model is particularly prone to a problem that is known from text retrieval literature, that is the semantic ambiguity introduced by *polysemy*. Polysemous words are characterized by different meanings and, typically, their actual semantics has to be inferred from the context, that is the surrounding terms. However, in a pure BOW approach each word is independent from the other so the semantic ambiguity cannot be easily solved.

Recently, there have been a number of works trying move past the bag-of-word assumption by introducing spatial information into the image histograms. For instance, (OB05) introduces the *localized feature histograms* representation, where a perceptual organization principle inspired

by the Gestalt theory is used to form compositions of local patches descriptors. In (MS06), the spatial relations between features are used to weight the BOW histogram by enhancing the relevance of those descriptors that match the objects layout, while irrelevant features are suppressed. The resulting representation is richer and more robust to clutter, although requires ground-truth object segmentations to learn the histogram weights. *Spatial pyramid matching* (LSP06) introduces a multi-resolution histogram that captures the aggregation of local descriptors at different spatial resolutions: an image is partitioned into in increasingly finer sub-regions and a BOW representation is built for each sub-region and finally aggregated in a unique histogram. This model is reported to perform extremely well for object categorization on the challenging Caltech101 dataset (FFFP04). In (SRE+05), the authors seek to increase to spatial specificity of object description by augmenting the visual vocabulary with *doublets*, that are pairs of visual words that co-occur often within a spatial neighborhood. A similar approach is taken by (ZWG06; YWY07), where feature proximity or co-occurrence are analyzed to build a highly discriminative image representation based on *visual phrases*. In (WLL+07), statistical language modeling techniques are used together with dense-sampling feature detection to extend the unigram representation, i.e. bag-of-words, to bigrams, where each visual word is conditionally dependent on its left neighbor, as well as to trigrams, where visual words are conditionally dependent on their left and upper neighbors. In (JDSW06), on the other hand, local patches are aggregated into compound terms using image annotations to drive perceptual grouping.

Some approaches have tried to get past the vector quantization step by introducing a continuous modeling of the descriptor distribution. Quelhas (dSQ06) proposes to model the descriptors distribution as a Gaussian mixture model: each local feature is represented by an histogram of Gaussian activations that describe the probability of the feature been generated by each of the GMM components. This approach has the advantage of producing a soft clustering of the local descriptors, enriching the image representation with a measure of how much each particular feature belongs to the corresponding cluster center. In (LJ08), the visual vocabulary

is learned simultaneously with the object recognition model in a latent topics approach that considers visual words as Gaussian mixtures of local descriptors. Perronnin (Per08), on the other hand, introduces adapted vocabularies that combine a universal visual dictionary, describing the visual content of all the image collection, with a specific vocabulary, that contains class specific visterms and is obtained through the adaptation of the universal vocabulary using class-specific data.

An alternative approach to image representation that founds on the BOW model uses latent aspects learned from the local descriptors distributions by means of probabilistic Latent Semantic Analysis (pLSA) (Hof01). The full details of this model will be given in Section 6.3 in the context of scene and object recognition models. The fundamental idea is to represent each image as an histogram of latent topics expression: each latent topic captures the co-occurrence of visual words within an image collection, hence can address the semantic ambiguity introduced by polysemous words: (QMO+07; BZM08) show how the latent topics representation can be used in combination with k-NN and support vector classifiers to achieve effective image ranking and scene categorization.

6.3 Models for Scene and Object Categorization

Once that a suitable description for the visual content has been obtained, it can be processed to discover semantically meaningful entities that are present in the image. In this section, we review models, mostly founding on the BOW representation, that address the tasks of object and scene recognition. The former task, in its most recent and most general interpretation, can be better interpreted as *object categorization*, that implies the recognition of general object classes from non specialized image sources. Earlier approaches to object recognition, on the other hand, focused more on recognizing the occurrence of a specific object in several images. Scene classification, likewise object recognition, tries to recognize the visual content of the image. However, objects are usually characterized by the presence of a limited number of specific parts organized in strongly structured configurations, while scenes are composed of several semantically

meaningful entities (e.g. sky, mountain, people,etc.) organized in weekly structured layouts (e.g. sky is typically above mountains, but people can appear almost everywhere in the picture). Scene classification attempts to organize such variable visual content by assigning an high level category label summarizing the overall scene information (e.g. urban, countryside, beach, etc,).

In the following we briefly review the main contributions to the object and scene recognition field: in Section 6.3.1, we focus on *weakly structured* approaches based on the BOW assumption and on latent aspect discovery; in Section 6.3.2 we describe structured approaches exploiting generative and *part-based* modeling of the objects and scenes.

6.3.1 Latent Aspect Discovery

Section 6.2.3 has introduced the bag-of-words image description that founds on the representation of textual documents as vectors of terms occurrences. By taking further this similitude, several works have applied computational models for text understanding with the intent of discovering semantically meaningful visual content from collections of BOW images.

In particular, the latent topic models has been widely applied to discover latent aspects in image collections. Sivic et al (SRE$^+$05), for instance, first proposed to exploit probabilistic Latent Semantic Analysis (pLSA) for the unsupervised discovery of object categories. The pLSA model has been introduced by Hoffmann (Hof01) to address a fundamental limitation of the earlier latent topic models such as the *mixture of unigrams*: this model assumes that each document can be assigned to a unique topic and, as a consequence, all the words in a document are constrained to belong to a single topic. The pLSA model relaxes this assumption by allowing words in a document to belong to different topics, obtaining a multi-topic representation for the documents in the collection.

The typical pLSA setting includes a collection of documents $\mathcal{D} = \{d_1, \ldots, d_D\}$ defined over a vocabulary of terms $\mathcal{W} = \{w_1, \ldots, w_W\}$. The document data is summarized by a rectangular integer matrix $n \in \mathbb{N}^{W \times D}$ such that each row $n(\cdot, d_i)$ contains the bag-of-words representation for

document d_i. The variables identifying words and documents, i.e. w_j and d_i, are observed, in contrast with a set of *latent* terms $\mathcal{Z} = \{z_1, \ldots, z_K\}$, that are hidden or unobserved variables. In pLSA, every observation $n(w_j, d_i)$ is associated to a latent topic k by means of the hidden variable z_k. The fundamental probabilities associated with this model are $P(d_i)$, that is the document probability, $P(w_j|z_k)$, that is the probability of word w_j conditioned on the latent variable z_k, and $P(z_k|d_i)$, that is the conditional probability of topic z_k given document d_i. Using this definitions, the pLSA scheme determines a (quasi) generative model for the word/document co-occurrences given by the following process

1. select a document d_i with probability $P(d_i)$;
2. pick a latent topic z_k with probability $P(z_k|d_i)$;
3. generate a word w_j with probability $P(w_j|z_k)$.

The same generative process can be described in a very compact way using the plate notation introduced by Buntine (Bun94) (see Fig. 58.a): each rectangular plate denotes the replication of its content for a number of times given by term on the bottom right (e.g. D and W_d for the outer and inner plates in Fig 58.a, respectively); each shaded circular item denotes an observed variable, while empty circles identify hidden variables. The direction of the arrows denote conditional dependence: e.g. z is dependent from d in Fig 58.a.

The pLSA data generation process can be translated in formulas starting from the joint probability distribution $P(w_j, d_i)$ and decomposing it by following the conditional dependencies in in Fig. 58.a as follows

$$P(w_j, d_i) = P(d_i)P(w_j|d_i) = P(d_i)\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k). \tag{6.1}$$

The second equality in (6.1) is given by the marginalization of the latent variables z_k and by the conditional independence assumption of the pLSA model, stating that word w_j and document d_i can be considered independent given the state of the latent variable z_k. In other words, the word distribution of a document is modeled as a convex combination of

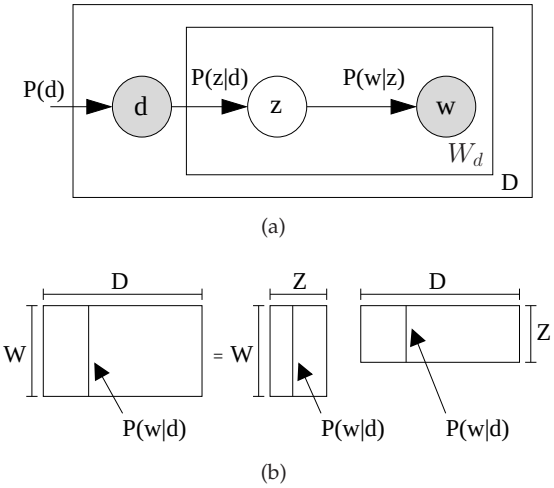


Figure 58: Probabilistic Latent Semantic Analysis: (a) graphical model, (b) matrix decomposition.

K aspect-specific distributions $P(w_j|z_k)$. Figure 58.b shows an interpretation of the pLSA model as a matrix factorization: essentially, the joint distribution matrix is decomposed in a document-topic matrix and and a word-topic matrix, such that the word-histogram of a document d_i becomes a mixture of word-topics weighted by the topics contribution to the document d_i, i.e. $\{P(z_1|d_i), \ldots, P(z_K|d_i)\}$. In this sense, pLSA can be interpreted as a special instance of a *Non-negative Matrix Factorization* (NMF) problem (LS99). In general, NMF relates to finding a factorization of an input matrix into two non-negative matrix, minimizing a cost function that measures the divergence between the original matrix and its non-negative factorization. In (GG05), it is shown how pLSA solves a non-negative matrix factorization problem with a Kullback-Leibler divergence.

The pLSA parameters, i.e. $P(w_j|z_k)$ and $P(z_k|d_i)$, are determined by maximization of the log-likelihood of observation $\mathcal{N}$ given the model pa-

rameters θ, that is

$$\begin{aligned}\log \mathcal{L}(\mathcal{N}|\theta) &= \log \prod_{i=1}^{D} \prod_{j=1}^{W} P(w_j, d_i)^{n(w_j,d_i)} \\ &= \sum_{i=1}^{D} \sum_{j=1}^{W} n(w_j, d_i) \log P(w_j, d_i),\end{aligned} \tag{6.2}$$

where $P(w_j, d_i)$ is expanded using the formulation in (6.1).

The parameter fitting process needs to infer the two multinomial distributions $P(w_j|z_k)$ and $P(w_j|z_k)$: the Expectation Maximization (EM) (DLR77) algorithm is used to this end, since it applies naturally to models comprising hidden variables. The derivation of the inference process is described in (Hof01). Essentially it consists of two steps: first, the *E-step* estimates the probability of the topic z_k given word w_j in document d_i as

$$P(z_k|w_j, d_i) = \frac{P(z_k|d_i)P(w_j|z_k)}{\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)}; \tag{6.3}$$

then, the *M-step* derives the $P(z_k|d_i)$ and $P(w_j|z_k)$ distributions from the estimated conditional distribution of topics $P(z_k|w_j, d_i)$ as follows

$$P(z_k|d_i) = \frac{\sum_{j=1}^{W} n(w_j, d_i)P(z_k|w_j, d_i)}{\sum_{j=1}^{W} n(w_j, d_i)}, \tag{6.4}$$

and

$$P(w_j|z_k) = \frac{\sum_{i=1}^{D} n(w_j, d_i)P(z_k|w_j, d_i)}{\sum_{j=1}^{W} \sum_{i=1}^{D} n(w_j, d_i)P(z_k|w_j, d_i)}. \tag{6.5}$$

This two-step process is iterated until a likelihood convergence criterion is met: often a validation set, or a *tempered* version of the EM are used in order to avoid model overfitting (Hof01).

The pLSA scheme maps naturally to image collection analysis by considering each visual word as a term w_j in an image document d_i. The pioneering work in (SRE+05) exploited pLSA for the unsupervised discovery of object categories in image collections: in particular, they showed how single latent topics learn to denotes image patches corresponding to a particular object class. Hence, by determining the most likely aspect

in the image based on $P(z_k|d_i)$, one can determine the most likely object depicted in the scene. Moreover, (SRE$^+$05) provides evidence that the knowledge acquired by applying the inference procedure on a training set, can be used to estimate the image content in a set of *fresh* images. This pLSA approach has been further extended in (FFFPZ05) and (LCC06) to include some partial spatial information concerning visual patches localization. Alternatively, the pLSA analysis has been seen as an intermediate feature extraction step, where the document-topic distribution vector $[P(z_1|d_i), \ldots, P(z_K|d_i)]^T$ can be used as a descriptor of the visual content in image d_i. Based on this idea, (QMO$^+$07) and (BZM08) independently developed an hybrid approach to scene categorization that uses pLSA to extract a latent topic characterization of the images that is in turn feed to a supervised k-NN or support vector classifier. In (QMO$^+$05), on the other hand, the document-topic distribution is used to rank the images with respect to a given latent aspect z_k, allowing to browse the image collection according to a sought visual theme.

Although pLSA describes a generative process for the word-document co-occurrences, it is not a full generative model. In fact, the document specific mixing weights for the topics are not sampled from a distribution, rather they are selected from $P(z|d)$ based on the document index d. Hence, pLSA indexes only the documents in the training set and cannot directly model unseen documents; furthermore it is prone to overfit the parameters to the training data. The Latent Dirichlet Allocation (LDA) (BNJ03) has been proposed to extend pLSA by treating the multinomial weights $P(z|d)$ as latent random variables, sampling them from a Dirichlet distribution, that is the conjugate prior of a multinomial distribution. As with pLSA, the goal is to maximize the data likelihood

$$P(w|\phi, \alpha, \beta) = \int \sum_z P(w|z, \phi) P(z|\theta) P(\theta|\alpha) P(\phi|\beta) d\theta \qquad (6.6)$$

where $P(w|z, \phi)$ is the word-topic distribution with parameter ϕ sampled from the Dirichlet distribution $P(\phi|\beta)$. The term $P(z|\theta)$ is the topic distribution having θ as document-specific multinomial parameter being sampled from the Dirichlet $P(\theta|\alpha)$. The terms α and β are the hyperparame-

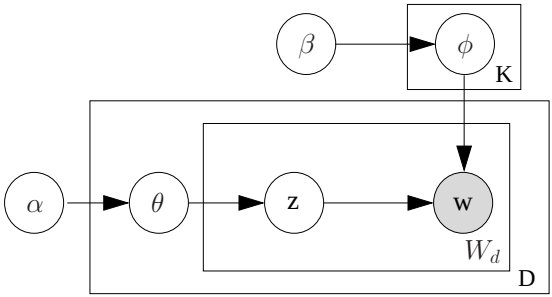


Figure 59: Graphical model for the Latent Dirichlet Allocation.

ters of the Dirichlet distribution: see Fig.59 for a graphical description of the model. Direct EM inference is impossible for the LDA model, since the integral in (6.6) is intractable due to the couplings between the parameters within the topic marginalization. Hence, approximate Bayesian inference methods such as variational expectation maximization (BNJ03), expectation propagation (ML02) and Gibbs sampling (GS04) have been used to fit the LDA parameters.

With respect to image content analysis, (RFE$^+$06) and (CFF07) apply LDA-based models to discover object categories from image segments. In (WZFF06), the authors apply a Bayesian extension of the LDA model, i.e. the Hierarchical Dirichlet Process (HDP), to categorize images in the challenging Caltech-101 dataset (FFFP04). Recently, (LJ08) proposed a simplified Gaussian-Multinomial LDA (BJ03) model that embeds visual vocabulary creation in the object model construction. This model does not need local descriptor quantization as in other latent aspects model based on the BOW representation; rather, it consider words as being Gaussian distributions over the SIFT descriptor space. These Gaussian distributions are enclosed in the LDA model and their parameters are fitted as part of the LDA inference process. In (FFP05), on the other hand, the authors extend LDA to supervised learning by introducing an explicit category variable and the resulting model is used to learn to categorize images into scenes.

Overall, there is no consensus on which of the two models, i.e. either LDA of pLSA, offers the best performance. Theoretically, LDA is

less prone to overfitting than pLSA, although experimental results have shown that pLSA can achieve better results than LDA (SRE+05). In general, pLSA is often preferred for its simplicity and the lower computational load: the approximate inference procedures used by LDA higher the time required for parameter fitting and might result in poorly fitting results.

6.3.2 Part-based and Generative Models

Latent aspect models try to learn object categories using mostly a *weakly* structured approach based on the bag-of-word assumption. Other approaches have explored more structured models where object categories are represented as collections of connected parts characterized by distinctive appearance and spatial position. A remarkable *part-based* approach is the *constellation* model (BMP98), a probabilistic scheme that describes objects as random constellations of parts and each part has an appearance, relative scale and can be be occluded or not. This model explicitly accounts for shape variations, i.e. modification in the mutual position of the parts, and for the randomness in the presence/absence of features due to occlusion; however, since the model is supervised, it requires consistent effort to provide sufficient ground truth data for the object categories. To address this issue, Weber et al. (WWP00) propose an unsupervised learning algorithm based on the maximum likelihood criterion. In (FPZ03), the authors extend further the constellation model by enforcing its invariance properties with respect to appearance variations and by adopting a saliency-based (KZB04) interest region operator (see Section 6.2.2) to detect object parts together with their characteristic scale.

Constellation is a generative object model, hence it represents appearance, scale, shape and occlusion by means of probability density functions (e.g. Gaussians). First, it detects regions and relative scales by means of interest point operators; then, it fits the density parameters by

maximizing the likelihood

$$P(X,S,A|\theta) = \sum_h P(X,S,A,h|\theta) = \\ = \sum_h \underbrace{P(A|X,S,h,\theta)}_{appearance} \underbrace{P(X|S,h,\theta)}_{shape} \underbrace{P(S|h,\theta)}_{scale} P(h|\theta) \tag{6.7}$$

where X denotes the interest regions locations, S the scales and A the appearances. This equation essential states that the likelihood can be factorized into appearance, shape and scale dependent terms given a latent variable h, called the hypothesis, that determines how detected regions are allocated to model parts.

Another generative part-based approach has been proposed in (RZ06) with application to human faces modeling. Given an image collection, parts correspond to fixed regions of the image that contain meaningful face sub-entities. In this approach, the range of appearances of each part is described by a set of probabilistic models, known as factors. The generative scheme dictates that an observed data vector is generated by stochastically selecting one appearance for each factor and one factor for each dimension of the data. The former selection allows each part to independently choose its appearance, while the latter describes how the parts combine to produce the observed data vector. Given a set of training examples, this model infers by variational algorithms the association of data dimensions with factors, as well as the appearance of each factor.

A simplified part-based approach is presented in (WaTM05). This model consider parts as being arbitrarily arranged, hence losing shape information, but occurring in particular proportions in objects, e.g. leaves on trees and windows on buildings. In particular, object classes are represented as conglomerates of visual words using a supervised generative model based on Gaussian distributions of visual words histograms. This model is faster than "full" part-based approaches and the experimental results in (WaTM05) highlight a notable performance in segmenting images into multiple objects. However, since it is a supervised model, it requires consistent amounts of ground-truth data, i.e. object segmentation, in order to be able to reliably estimate the object categories. The *Hierar-*

chical Pattern model (PCM07) addresses scene categorization by means of generative approach that is based on a hierarchical organization of the visual content. Images are clustered at different description levels: first, they are considered as whole entities and organized based on global patterns alone. This allows to group together images with an overall scene similitude, disregarding variations at the local level. Then, those images showing local typicality patterns are analyzed at increasingly finer scales so to obtain a structured categorization of the visual features.

Perceptual grouping is used in the Bayesian compositionality approach in (OSB06) (see Fig. 60.a): as in the constellation model, objects are represented based on their components and the relations between them; however, instead of modeling a flat constellation of parts, the compositionality approach learns intermediate groupings of parts forming a hierarchy of recursive compositions. In particular, this model first groups local patches that lie on common curves; then, curves are agglomerated if this increases the discriminative power of the compound with respect to the original constituents.

In (KK07) is presented a hierarchical approach that represents indoor scenes as undirected graphical models comprising place, object and part nodes. This model proposes a unified framework for describing contextual information, that is represented as interactions/links between nodes in the hierarchy. In particular, spatial context is introduced as intra-layer interactions (e.g. between part nodes), while top-down context or bottom-up influence is represented by inter-layer connections (see Fig. 60.b). The resulting graphical model is quite complex and needs to resort to several variational approximations in order to fit the parameters to the data.

In general, part-based models offer a structured representation of the visual content, being able to capture the spatial and contextual relationships between the object/scene components. However, they require a large number of parameters to be able to cope with the multiple variations in appearance of the object/scene categories they are seeking to model, thus requiring consistent computational efforts in order to fit the model parameters. Moreover, their performance is strongly influenced by the results of the interest point detection phase, since failure to de-

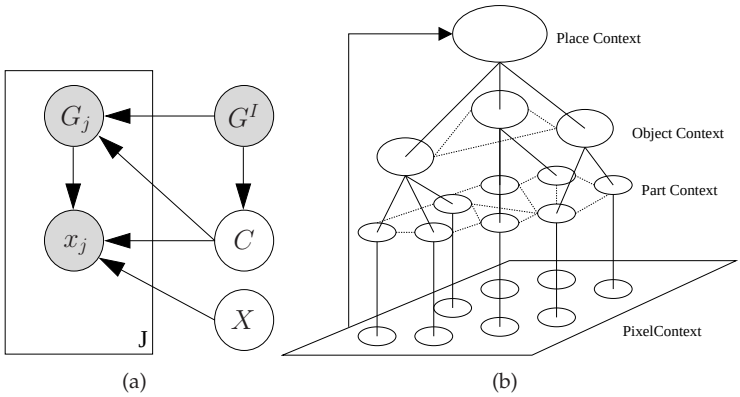


Figure 60: Examples of part-based hierarchical models. (a) Compositionality approach: Bayesian network coupling the J compositions G_j based on their locations X_j, the location of the object center X, a bag of features G_I and the image categorization C. (b) Hierarchical graphical model for the unified context framework: dotted lines represent spatial context while solid lines represent bottom-up and top-down hierarchical contextual relationships.

tect a key part of the object model might result in failure to detect the whole object. Hence, the use of part-based models has been limited, so far, to problems comprising a small/medium number of object categories or small variability in the scene description (e.g. see (KK07)).

6.4 Image Annotation

Automatic image annotation systems address the problem of learning the association between image content and text by relying on previously annotated data to learn the connection between words and visual features. Image annotation is a complex task and, in a sense, can be seen as the generalization of multi-object and scene characterization, since it tries to bind semantic information, i.e. the annotations, to the whole image or to significant portions of it. However, image annotation does not seek for explicit object presence, rather it estimates the probability of an image containing a particular concept. Image annotations allows to linguistically index

pictorial content, hence providing a powerful tool for retrieving images from large collections by expressing queries using textual terms. In the following, we review the main contributions to this field: following the presentation scheme in (Yav07), we differentiate approaches that follow a probabilistic modeling of the task from non-probabilistic models.

6.4.1 Non-Probabilistic Image Annotation

Non-probabilistic image annotation models are typically rooted in the information retrieval community and often use a bag-of-words representation of the image content. In (HLES06), for instance, it is exploited a cross language information retrieval system based on an algebraic version of the pLSA, named Latent Semantic Indexing (LSI). Likewise pLSA, the LSI algorithm produces a decomposition of the word-document matrix ensuring that frequently co-occurrent terms are mapped to the same topic. If the word-document matrix contains terms from two different languages, then LSI produces a decomposition that correlates words from the different idioms. Documents that are available in a single language are projected in the LSI topic-space, so that they can be associated with terms from the other language. Hare et al. (HLES06) have exploited this model within an image annotation framework by considering the image-documents as being represented by two idioms: a visual language, describing the visual content in a BOW representation, and a textual language containing the image annotations. The projection of an unseen image on the resulting LSI space produces automatic annotations of the visual content at the level of the whole image.

In (TL07), on the other hand, annotations are produced at the level of individual image regions. They exploit an image description inspired by the popular *term vectors* representation from the information retrieval community (SWY75). In their approach, a training set of annotated images is used as vocabulary to represent both the visual and the textual terms. In particular, each training image is a dimension in the vocabulary vector space such that an annotation a is represented by a vector t_a whose components correspond to the images in the training set. The j-th

component t_{aj} of this vector is set to 1 if the j-th image in the training set contains the annotation a, otherwise is set to 0. An image region r, on the other hand, is first represented as a bag of local patches; this low level description is then confronted with the regions in the training set to produce a refined vector w_r. In particular, each component w_{rj} contains the cosine similarity of the BOW descriptor of region r with the best matching region in the j-th image of the training set. Since the term vectors t_a and the region descriptors w_r are defined on the same vocabulary, a new region r_1 can be annotated by confronting its descriptor w_{r_1} with the t_a vectors, selecting the best matching terms as image annotations.

In (LCZ+06), the image annotation problem is seen as a *search and mine* task: the un-captioned images are first confronted with a large scale database of pictures crawled from the Web and characterized by a rich annotation content. A set of best matching pictures is selected and, in the mining phase, a clustering algorithm is used to find the most representative keywords for the annotations of the retrieved image set. These keywords are then ranked based on a saliency measure and used to annotate the un-captioned image.

Discriminative approaches have also been used to address image annotation. In (CCS04), a multi-class Support Vector Machine (SVM) is trained to perform semantic segmentation by classifying image pixels into one of the annotation categories: however, this model has been tested only with a few classes (i.e. $7/8$), it is thus debatable whether it has to be considered an image annotation system rather than an object recognition model. *Multiple-instance learning* (MIL) has been proposed as a natural tool for modeling image annotation (MR98): this learning strategy builds classifiers form bags of labeled examples, such that one typically knows only whether the bag contains or not a positive example for the class, but cannot relate this information to the bag's content. In the image annotation context, a picture is represented as a bag of regions or local patches (i.e. the typical BOW description), while the captions represent the labels that have to be attached to the bag content. In (QH07), is presented an image annotation system that integrates MIL with SVM. The former algorithm is used to generate an intermediate representation for the visual

content that is in turn feed to a set of SVMs that learn to associate textual annotations.

6.4.2 Probabilistic Image Annotation

The majority of the existing works in image annotation relies on a probabilistic formulation of the problem, that focuses on estimating the probability that a set of annotations is emitted by some particular image feature. The problem of learning joint probabilities between text and image features has been addressed at three levels of visual content representation, that is global, regional and local level. In (CSS98), for instance, is proposed a global descriptor approach where visual features are computed over the whole image.

However, the image annotation approaches in the literature are mostly based on the quantization of interest patches or whole regions. One of the earliest approaches in this sense, is the *co-occurence* model in (MTO99): images are partitioned into local segments by means of a fixed grid. Image segments are then described by their color and texture content and vector quantized to obtain a finite set of visual terms w and a bag-of-words representation for the image content. Given a set of annotated training images, the co-occurrence model estimates the empirical distribution of each textual caption t given the visual terms w as

$$P(t|w) = \frac{N(t,w)}{\sum_t N(t,w)} \tag{6.8}$$

where $N(t,w)$ is the number of co-occurrences of term t with the visterm w. Based on the probability in (6.8), an un-captioned image is annotated by computing the probability of each annotation given the visual terms extracted from the picture.

Besides fixed-grid partitioning, image segments, or *blobs*, have been widely used to represent visual content in image annotation. In (Bar01), for instance, it is proposed a probabilistic generative model in which words and image regions are independently emitted at different levels of a fixed hierarchy of image clusters. In particular, each node in the hierarchy has some probability of generating terms and image regions, rep-

resented by means of Blobword (CBGM02) features. Images are assigned to clusters, that are the leaves of the hierarchy tree, and their words and segments are modeled as being generated from a distribution over the nodes on the path from the root to the leaf corresponding to the cluster. Mathematically, this is modeled as a sum over the clusters weighted by the probability that the image d is in the cluster, that is

$$P(D|d) = \sum_{c} P(c) \prod_{t \in D} \sum_{l} P(t|l,c) P(l|c,d) \tag{6.9}$$

where D is the set of visual and textual terms t in image d, c indexes clusters and l indexes levels. The EM algorithm is used to fit the model, comprising two hidden variable sets: the former indicates the membership of a training document d to a given cluster c, while the latter indicates that term t is generated at level l of the hierarchy.

The *Translation* model in (DBdFF02) uses Brown's machine translation framework (BPPM94) for associating image segments (that are words in one language) to annotations (that are the translated words): blobs are identified by Normalized Cuts (SM00) segmentation and are represented by vector quantized color and texture features. The EM algorithm is used to learn the translation probabilities (*translation table*) linking a blob b within image d with the annotations t: given a a trained translation table, an un-captioned image is annotated by picking the most likely word for each of its blobs. Brown's model is used also in (JDSW06) with local patches extracted by interest point detectors. In this approach, textual annotations serve also as an aggregating means for detecting visterms compounds, that are sets of frequently co-occurring patches with respect to single textual terms. Given an un-captioned image, compound visual terms are detected and annotated with the most likely term estimated using the translation table.

Similarly, the *Cross Media Relevance Model* (CMRM) (JLM03) considers images as sets of words and blobs, that are terms from two different lexicons. However, CMRM loosens the translation model's assumption that considers annotations and blobs to be in a one-to-one correspondence: instead of computing a translation table, CMRM estimates the probability

of observing an annotation t together with a blob b. An un-captioned image is annotated by choosing the most probable terms t_j under the joint probability

$$P(t_j, b_1, \ldots, b_I) = \sum_{d \in D_T} P(d) P(t_j|d) \prod_{i=1}^{I} P(b_i|d) \tag{6.10}$$

where b_i are the blobs describing the un-captioned image and D_T is the set of training images. Equation (6.10) essentially considers words and blobs as being generated independently given an image d in the training set. Given this assumption, the probability of seeing an annotation t_j together with the blobs b_i is estimated based on statistics on the occurrence of annotations and blobs in the training images. In particular, the probability of a term, either an annotation or blob, occurring in a training image d is given by a multinomial distribution smoothed by the background probability of the term with respect to the whole training set. The CMRM is reported to improve consistently the performance of translation model: however, rather than annotating blobs, CMRM provides captions for the whole image. In (THL06), the CMRM model is adapted to a salient regions: image segmentation is avoided by representing pictures as bags of local SIFT patches, while the multinomial probabilities $P(w_i|d)$ are estimated for each quantized visterm w_i.

So far, we have described models that rely on vector quantized descriptors of the image regions: in (LMJ03) it has been proposed the *Continuous Relevance Model* (CRM) that extends CMRM by abandoning the quantization of the blobs. Instead of computing the probability $P(b_i|d)$ as a smoothed multinomial, the CRM model computes it using a non-parametric Gaussian kernel density estimate of its un-quantized blob feature vector. In (FML04), it has been proposed an extension to the CRM model that replaces the multinomial image-annotation distribution $P(t_j|d)$ with a multiple Bernoulli distribution.

A non-parametric kernel density estimate approach is exploited in (YSR05) using global image features in place of a blob representation. This approach models the annotation process using a simple Bayesian scenario, where the probability of labeling an image d with a word t is

given by

$$P(t|d) = \frac{f(d|t)p(t)}{f(d)} \tag{6.11}$$

where $f(d|t)$ is the density of d conditional to the assignment of the label t. The image d is represented either by color and texture features or by a *signature* $s = \{(c_1, m_1), \ldots, (c_I, m_I)\}$, that models the centroids c_i of I k-means clusters in CIE-Lab space as well as the number of points m_i belonging to each cluster. The function $f(d|t)$ is then approximated by a Gaussian non-parametric kernel density estimator for each annotation t: additionally, the *Earth Movers Distance* (EMD) (Rub98) is used in place of the Euclidean measure in the Gaussian kernel, when images are represented by signatures. The model is shown to perform as well as the CRM when using image signatures with the EMD measure.

Besides scene and object recognition, latent variable models have found application also in automatic image annotation: (BJ03), for instance, describes several generative models based on a blob-like image representation. The simplest approach presented in (BJ03) is a finite mixture model named *Gaussian-multinomial mixture* (GM-MIXTURE), where a single latent aspect z is used to bind the visual modality, i.e. the blobs b, with the annotations t. Likewise in the CRM, the distribution of blobs and words are modeled by Gaussian and multinomial distributions, respectively. The generative process in Fig. 61.a essentially states that images and captions are generated by first choosing a topic z and then repeatedly sampling B_d region descriptors and M_d labels, conditional on z. This model is limited to a single latent aspect for each image, that is sampled once and held fixed during the generation of the blobs and labels. Moreover, the correspondence between regions and words is totally ignored, since blobs and annotations are independent conditioned on the latent variable z.

To address the limitations of the GM-MIXTURE approach, (BJ03) first introduced a multimodal extension to the LDA model (BNJ03), named *Gaussian-Multinomial LDA* (GM-LDA). As discussed in Section 6.3.1, LDA allows multiple topics to be allocated for a single image: likewise, the GM-LDA permits to generate regions and words from different latent as-

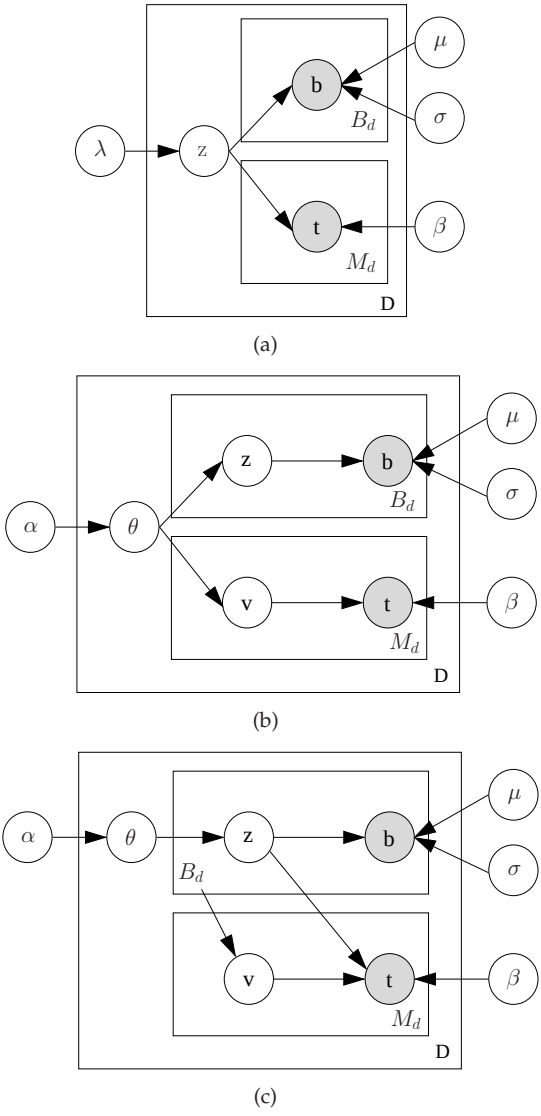


Figure 61: Latent topic image annotation (BJ03): (a) GM-Mixture model, (b) GM-LDA model and (c) CORR-LDA model. See the text for further details.

pects within the same image. The GM-LDA generative process in Fig. 61.b states that the the Dirichlet variable θ is sampled once for each image, while blob and word topics (i.e. z and v, respectively) are selected for each region and annotation but from the same Dirichlet variable. In a sense, the parameter θ describes how blob and word topics can be assembled within an image. However, this increased flexibility with respect to the GM-MIXTURE does not produce a performance improvement in the GM-LDA model, as detailed in the experimental evaluation in (BJ03). This is due to the fact that the latent factors z and r are not directly dependent on each other, hence GM-LDA cannot model the relationship between the visual content and the image captions. The same authors in (BJ03) introduce a third model, named *Correspondence LDA* (CORR-LDA), that extends the flexible GM-LDA approach by modeling the conditional correspondence between blob and word topics (see Fig. Fig. 61.c). The key assumption of this model is that region aspects are generated first, while captions are generated conditional on their topics v as well as the region aspects z. In other words, after having generated B_d blob descriptors b, we then select one blob b and its generating topic z for each annotation, and we finally sample a term t conditioned on z. Essentially, in CORR-LDA, image regions can be sampled from any visual topic, while image annotations are conditioned to be selected from the visual aspects that are present in the image. This representation is flexible enough to allow caption words to be drawn from a subset of the image regions with multiple terms eventually coming from the same blob. The experimental results in (BJ03) show that CORR-LDA improves significantly the GM-MIXTURE model. In (BDdF$^+$03), the CORR-LDA is further extended by the *Mixture of Multi-Modal LDA* (MoM-LDA) to model independent image collections. Each collection is modeled by a randomly generated mixture over latent factors: in particular, a multinomial collection variable c is added as a prior to the Dirichlet term θ and is sampled once for each image, providing a means for constraining the choice of the topics mixture for images within the same corpus.

Three models inspired by the pLSA scheme are presented in (MGP07) to deal with annotated image collections. The performance of the three

models is evaluated with different image representation, that are a global descriptor of the pixel color distribution, a blob-like representation based on image regions and a localized description based on SIFT and local interest points. The simpler latent aspect scheme for annotated images, named PLSA-MIXED, learns a standard pLSA model on a concatenated representation of text and visual content within an image. In other words, each image descriptor $x = (v, t)$ is the concatenation of the visual descriptor v with the bag of annotations t, and the word-topic distribution $P(x|z)$ captures the co-occurrence of visual features and words together. An uncaptioned image d_{new} is then represented by the visual part alone, i.e. $x_{d_{new}} = (v_{d_{new}}, 0)$, and the conditional probability $P(x|d_{new})$ is generated using the standard *folding-in* (SRE$^+$05) procedure. Finally, the conditional distribution over the captions $P(t_{d_{new}}|d_{new})$ is extracted from $P(x|d_{new})$ and used to annotate the image. This approach implicitly assumes that both modalities have the same influence in determining the latent space decomposition and, likewise the GM-LDA model, does not allow to relate the visual content with the annotations (recall that the pLSA approach assumes that the elements in the bag are independent conditioned on the latent aspect z). The second and third pLSA-based models (MGP07) address these issues by first generating features with one modality and then sampling the other modality from the subset of aspects that generated the first features, in a process that closely resembles that of the CORR-LDA approach. The PLSA-FEATURES model, for instance, allows to estimate the document-topic distribution $P(z|d)$ from the visual modality alone, while learning also the distribution $P(v|z)$ of the visual features conditioned on the latent aspects. When learning converges, the conditional distribution of annotations given the latent topics, i.e. $P(t|z)$, is estimated while keeping $P(z|d)$ fixed for the textual modality, likewise in the folding-in process. Conversely, the PLSA-WORDS first estimates the topic distribution for the textual modality and then keeps it fixed for the visual information. The annotation of a new image d_{new} is then performed by probabilistic inference using the folding-in procedure to esti-

mate the conditional distribution over captions

$$P(t|d_{new}) = \sum_k P(t|z_k)P(z_k|d_{new}). \tag{6.12}$$

The experimental results in (MGP07) show that PLSA-WORDS has a better retrieval performance than PLSA-FEATURES: this is not un-expected since, in general, annotations are more semantically discriminative than visual features; hence, they produce a document space decomposition that separates more sharply the images with respect to their visual semantics.

Other relevant probabilistic approaches to image annotation include the *2D Multiresolution Hidden Markov Model* (2D-MHMM) (LW03) where images are segmented by a fixed grid at progressively higher resolutions and a 2D-MHMM model is trained for each image category to capture segment dependencies. Each category is associated with multiple annotations and the likelihood of an un-captioned image (based on the learned 2D-MHMM category models) is used to annotate it with the category labels. A supervised generative/discriminative approach is presented in (FGLX04), where SVM classifiers are trained on labeled image regions to recognize and annotate salient objects, while a Gaussian mixture model is used to estimate the scene caption, achieving a multi-level annotation of the image. In (WFCX06), contextual information is integrated in a region-based probabilistic annotation model. In particular, spatial context is modeled in such a way that the choice of a region label is influenced by the annotations of its neighboring blobs. In (ZWZ+07), on the other hand, it is explored the contextual relationships between the terms in the image annotations: the CMRM model is extended by incorporating co-occurrence statistics for the textual terms, achieving substantial improvements with respect the multiple Bernoulli relevance model (FML04). The *Collaterally Confirmed Labelling* model in (ZB07) integrates both the spatial context given by region co-occurrence as well as the annotation context given by term co-occurrence statistics gathered from external sources (e.g. web pages, linguistic corpora, etc.). In (WG07), it is presented an hybrid approach that exploits the translation model to attach labels to the topics in a probabilistic latent semantic analysis model fitted with the vi-

sual information from an annotated image collection. words. In a sense, compared to the original translation model, this approach provides an intermediate level of description for the visual content based on the learned topics: however, likewise the original model, it assumes a one-to-one correspondence between topics and words.

Chapter 7

Image Region Annotation By Hierarchical Region Topic Discovery

7.1 Introduction

In the last ten years, there has been a notable strive to realize new tools for the semantic categorization of image content, motivated both by the availability of consistent collections of weakly annotated data (e.g. an image with its surrounding text in a web page) as well as by the recent advancements in image processing models. In the previous chapter, it has been noted that there is an increasing attention towards new machine vision models that overtake flat region-based and keypoint-based approaches in favor of multi-layered representations conveying richer semantics and discriminative power than a BOW model, but without the rigidity of a part-based approach.

In this chapter, we present a model that takes further the idea of stratifying the representation of the visual content. In particular, we propose to exploit a multi-layered description both at the level of image representation as well as the *semantic* level of latent topic discovery. To address the former issue, we define a multi-resolution representation of the visual

content that draws both on region-based as well as on keypoint-based models. The proposed image representation approach is integrated into a hierarchical latent variable model that is used to discover visual aspects at different levels of image resolution. By exploiting the *asymmetric pLSA learning* proposed by (MGP07), we extend the hierarchical latent variable algorithm to allow representing multimodal information collections comprising images and related textual captions. Such captions act as a source of weakly supervised information, that we exploit to learn to annotate the single image regions based on the distribution of their latent aspects.

The remainder of the chapter is organized as follows: Section 7.2 introduces the multi-resolution image representation, while in Section 7.3.1 we lay down the foundations of the Hierarchical Region-Topic Probabilistic Latent Semantic Analysis (HRPLSA), that is extended in Section 7.3.2 to address automatic image annotation. Section 7.4 discusses the relationships of the proposed approach with the state-of-the-art latent aspect models. Finally, in Sections 7.5 and 7.6 we present a performance evaluation in unsupervised visual classes segmentation and image annotation, respectively.

7.2 Multi-resolution Image Representation

The definition of a suitable representation of the visual information content is an essential prerequisite for developing competitive image understanding models. In this section, we introduce a *multi-resolution* description for the pictorial information that integrates within the region-based hierarchical latent topic model that we describe in Section 7.3. Since the aim of this latent model is to learn to recognize and segment semantically meaningful portions of a picture, we need a representation scheme that provides sufficient information concerning the appearance of such image regions.

The term *multi-resolution* stems exactly from the need of describing visual information at different spatial granularity, by borrowing both from the BOW and the region-based image representations. In Section 6.2 we have pointed out that, on the one hand, region-based approaches do not

offer sufficient warrantees with respect to robustness to occlusion and affine transformations of the visual entities; moreover, they rely strongly on the performance of segmentation algorithms, that is not sufficient to ensure the isolation of relevant objects. On the other hand, the BOW representation seems to be too strongly biased by the assumption that spatial distribution of visual keywords can be neglected when determining the overall image appearance. In our opinion, such an approach has the best performance on image corpora characterized by small variability of the pictorial content, especially when applied to the characterization of single-object images (e.g. the Caltech101 data (FFFP04)) or to high level scene characterization, such as in scene recognition. However, when dealing with varied visual information and with complex tasks encompassing the identification of multiple relevant entities within the same picture, such as in image annotation, we need a more structured approach that can isolate semantically different image portions and process them with little noise coming from the visual keywords of the surrounding regions.

Our intention is to devise a flexible representation that can account for the spatial structuring of the visual information content, while retaining the tolerance of a BOW description. To do so, we shift the application of the BOW representation to a lower spatial granularity: in particular, we first extract a set of visual keywords, either by dense sampling or interest point detection, that are in turn aggregated based on their spatial contiguity as determined by a coarse segmentation of the image into perceptually homogenous portions. Figure 62 exemplifies the proposed multiresolution representation: an image d is, essentially, a bag of R_d regions which are, themselves, bags of visual keywords w_j. Hence, each region is represented by an histogram counting the occurrences of visual codewords within the segment. Seen from the perspective of textual latent semantic analysis, this representation essentially assumes that a document can be characterized by the unordered paragraphs in the text, whose general concept is, in turn, independent from the disposition of the words in the paragraph.

Image regions are obtained by performing CoRe segmentation on the image pixels, such that each image position is represented by the corre-

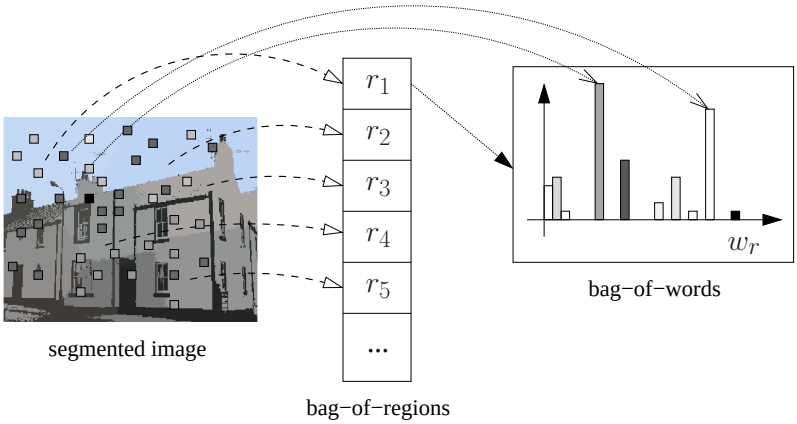


Figure 62: Multi-resolution representation: images undergo both visual keypoints extraction and region segmentation. Each image is characterized by its composing regions r_l; each region is, in turn, described as a bag of those visual keypoints w_j that fall into its portion of the image.

sponding three-dimensional color descriptor in the perceptually uniform CIE-LAB (WS82) space as well as by the coordinates of the pixel. CoRe segmentation is employed because, given the results in Chapter 2 and 3, it produces reliable estimates of the underlying pixel distribution with a linear computational complexity. Moreover, as shown in Sections 3.2 and 3.3, it is robust noise and outliers and its repetition-suppression mechanism allows to detect small dimensional clusters (i.e. corresponding to small homogeneous image regions) even in presence of consistently denser datapoint concentrations (i.e. corresponding to larger visual entities, such as sky, grass, etc.). The settings for the CoRe algorithm are those described in Section 2.6.1 and the number of initial units is chosen equal to 20, that offers a fair tradeoff between quality of the estimate and computation effort. Figure 63 shows an example of segmented images from the Microsoft Research Cambridge dataset (MRSC-B1) (WaTM05). This dataset provides ground truth segmentations for a number of visual classes: in Section 7.5 we exploit this information to evaluate the segmentation performance of the hierarchical latent topic model proposed

in Section 7.3.1.

Visual keypoints can be extracted based on dense sampling or using one of the interest point detectors described in Section 6.2.2. In this chapter, we focus on densely sampled keypoints since they have been shown to be more effective than sparse interest point detectors for scene categorization (BZM08) and image annotation (MGP07), that are both problems dealing with articulated scenes comprising multiple visual entities. Sparse interest points, on the other hand, have shown to perform well when dealing with object recognition, while they do not seem to capture well the informative portions of large scenes comprising landscapes and large perceptually uniform regions. Following the approach in (BZM08), we select gray-level patches with radius $r = 4, 8, 12$ and 16, that lie on a multi-resolution grid with spacing $\delta = 5, 10$ and 15. In general, we randomly sample 2000 or 3000 patches from each image and we represent them by SIFT descriptors using the VLFeat package (Ved07). Standard k-means clustering is then performed on the SIFT descriptors to extract a codebook of visterms (typically 1000–3000), that is in turn used to vector quantize the local patches within each image.

As a final step, regions and keywords are brought together: each region r_l is represented by the histogram of visterms occurring with their centers in the image portion assigned to the segment. Notice that CoRe provides a description of the detected image segments in terms of Gaussian functions: in principle we can exploit this to devise an alternative image representation approach, where keypoints are not assigned selectively to a single image region; rather, they can be shared between those segments whose Gaussian response (to the keypoint center) exceeds a minimum threshold.

Compared to the work in (RFE$^+$06; SRZ$^+$08), our representation offers the advantage of avoiding to tear the image into pieces: in their work, segments are considered as standalone entities that are supposed to represent objects in their entirety. In order to find *good* objects they perform multiple segmentations of the same image, obtaining a pool of alternative, possibly overlapping, regions among which they choose the candidate solution. In our approach, every picture is represented in its entirety, and each region

Figure 63: Segmentation examples for the MRSC-B1 Dataset (WaTM05).

is in the context of its originating image. A similar approach has been developed by (CFF07): they perform an over-segmentation of the images to extract a large number of image portions (i.e. $30 - 50$). Then, for each segment they extract a set of informative keypoints by the scale invariant saliency detector (see Section 6.2.2) and represent them as a bag of visterms. Notice that we take a different approach to the creation of the regional bag-of-words: *first*, we extract keypoints from the *whole* image and *then* we assign them to the regions. In a sense, we seek keypoints that describe the visual content of the whole image and we pool them together based on the perceptual concepts of closure and spatial contiguity. From a computational point of view, this allows to parallelize the image segmentation process and the keypoint extraction phase; moreover, a variation in the segmentation of the image does not require recomputing the whole image descriptor. This later aspect poses also a robustness issue for the model in (CFF07): some image segmentations, in fact, might lead to the extraction of mis-informative keypoints; our representation, on the other hand, is more stable since the set of keypoints used to compute the region BOW does not depend on the outcome of the segmentation.

7.3 Region Annotation by Hierarchical Probabilistic Latent Semantic Analysis

In this section, we introduce our proposal for a hierarchical latent topic model for image segment annotation. In particular, in Section 7.3.1, we describe the details of the *Hierarchical Region-Topic Probabilistic Latent Semantic Analysis* (HRPLSA) that we use to learn a multilayered and multiresolution representation of the visual content. In Section 7.3.2, we show how to extend HRPLSA to model the association between visual content and related textual information.

7.3.1 Hierarchical Region-Topic Probabilistic Latent Semantic Analysis (HRPLSA)

The hierarchical region-topic model extends the probabilistic latent semantic analysis by introducing a multilayered representation of the visual content both on the *spatial* and on the *semantic* level. The former aspect refers to the fact that images are no longer represented as bag-of-keywords; rather, they are considered as an unordered collection of coarse regions, which are themselves described as bags of visterms. The semantic aspect, on the other hand, refers to the fact that HRPLSA introduces a hierarchy of latent topics, such that the appearance of each region is associated to a mixture of high level aspects that, in turn, determine the mixing proportions of low level topic associated to the visterms in the region.

More formally, the HRPLSA setting is defined by a collection of image documents $\mathcal{D} = \{d_1, \ldots, d_i, \ldots, d_D\}$, such that the i-th image is represented as a bag-of-regions $d_i = \{r_1, \ldots, r_l, \ldots, r_{R_i}\}$ where each region r_l is itself a bag-of-visterms $r_l = \{w_1, \ldots, w_j, \ldots, w_{W_l}\}$. The generative probabilistic model of HRPLSA is described in Figure 64: observed variables include the documents d_i, the regions r_l and the visual words w_j, while the latent variables include the region topics $\mathcal{Z} = \{z_1, \ldots, z_k, \ldots, z_K\}$ as well as the word aspects $\mathcal{T} = \{t_1, \ldots, t_p, \ldots, t_T\}$. The observation matrix, that in the original pLSA model is a two-dimensional matrix of word-document occurrences, is now a three-dimensional matrix $n(w_j, r_l, d_i)$, containing the number of occurrence of word w_j in region r_l of image d_i.

The fundamental probabilities associated with the HRPLSA model can be better understood by taking a top-down approach, starting from the coarser image representation (region level) to the finer (word level). Let's first consider the occurrence of a region r_l within an image d_i; the associated co-occurrence probability $P(d_i, r_l)$ can be decomposed by marginalizing the latent topics z_k as in the original pLSA model, that is

$$P(d_i, r_l) = P(d_i) \sum_{k=1}^{K} P(z_k|d_i) P(r_l|z_k) \tag{7.1}$$

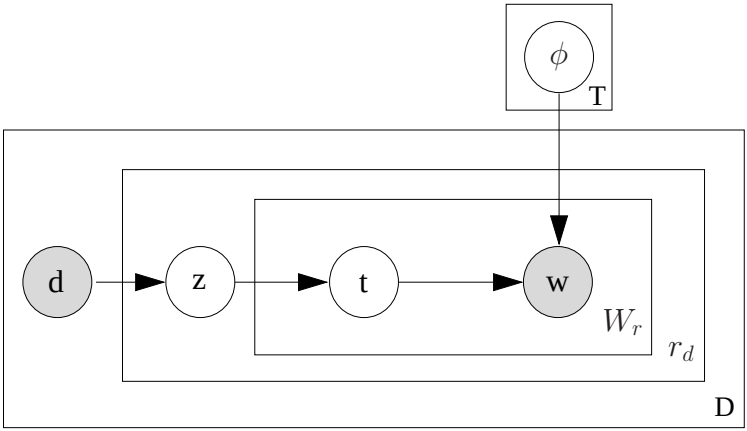


Figure 64: The Hierarchical Region-Topic Probabilistic Latent Semantic Analysis as a generative model: an image document d contains a set of regions r_d, that are themselves bags of W_r visual words. Notice that each region is assigned to a single topic z and that each region aspect z is characterized by a different proportion of word topics t. The ϕ term denotes the multinomial parameter of the word-topic distribution.

where $P(z_k|d_i)$ denotes the mixing proportions of the region topics given the document. The probability of a region in d_i being generated by an aspect, i.e. $P(r_l|z_k)$, can be further expanded by applying the BOW assumption to the regions, yielding

$$P(r_l, z_k) = P(w_1, \ldots, w_{W_l}|z_k) = \prod_{j=1}^{W_l} \left(\sum_{p=1}^{T} P(t_p|z_k) P(w_j|t_p) \right)^{n(d_i, r_l, w_j)} . \tag{7.2}$$

Equation (7.2) expresses the BOW assumption, stating that visual words in a region r_l are generated independently conditioned on the choice of a visterm topic t_p. The probability $P(t_p|z_k)$ determines the mixing proportions of word topics given the region aspects, while $\phi_{jp} = P(w_j|t_p)$ is the multinomial word-topic distribution. For the sake of notational compactness, in the following, we will refer to the observation $n(w_j, r_l, d_i)$ as n_{jli}. Given the model probabilities, the generative process of HRPLSA can be stated as follows

1. Select a document d_i with probability $P(d_i)$.

2. For each region $r_l \in d_i$,

   (a) pick a topic z_k with probability $P(z_k|d_i)$;

   (b) for each word $w_j \in r_l$,

      i. pick a topic t_p with probability $P(t_p|z_k)$;

      ii. choose a word w_j from the multinomial ϕ_{jp}.

In order to estimate the model parameters, i.e $\theta = \{P(z_k|d_i), P(t_p|z_k), \phi_{jp}\}$, we need to formalize the log-likelihood of the collection $\mathcal{N}$, that is

$$\mathcal{L}(\mathcal{N}|\theta) = \sum_{i=1}^{D}\sum_{l=1}^{R_i} \log \left[P(d_i) \sum_{k=1}^{K} P(z_k|d_i) \prod_{j=1}^{W_l} \left(\sum_{p=1}^{T} P(t_p|z_k)\phi_{jp} \right)^{n_{ilj}} \right]. \tag{7.3}$$

The log-likelihood maximization cannot be performed directly on the formulation in (7.3), due to the sums in the logarithm. However, the *Hierachical Latent Variable Model* (BT98) provides a hierarchical extension of the Expectation-Maximization algorithm, that can be adapted to fit the HRPLSA model. First, we introduce two sets of *indicator* variables that specify which component of the mixture is responsible for generating the single data samples. For instance, Z_{ilk} determines which topic z_k is responsible for the generation of the l-th region in the i-th image; similarly, T_{kjp} specifies which topic t_p generated the word w_j given the region aspect z_k. Essentially, indicator variables are such that $Z_{ilk} = 1$ (or $T_{kjp} = 1$) only for the generating component k (respectively p) and they are equal to zero for all the other topics. Hence, the complete likelihood of the HRPLSA model can be restated as

$$L_C(\mathcal{N}|\theta) = \prod_{i=1}^{D}\prod_{l=1}^{R_i}\sum_{k=1}^{K} P(z_k|d_i)^{Z_{ilk}} \prod_{j=1}^{W_l} \left[\sum_{p=1}^{T} P(t_p|z_k)(\phi_{jp})^{T_{kjp}} \right]^{n_{ilj}} \tag{7.4}$$

where the document probability $P(d_i)$ has been neglected since we assume that documents in the collection are equiprobable. The log-likelihood

corresponding to (7.4) is

$$\mathcal{L}_{\mathcal{C}}(\mathcal{N}|\theta) = \sum_{i=1}^{D}\sum_{l=1}^{R_i}\sum_{k=1}^{K} Z_{ilk}\left[\log(P(z_k|d_i)) + \sum_{j=1}^{W_l} n_{ilj}\sum_{p=1}^{T} T_{kjp}\log\left(P(t_p|z_k)\phi_{jp}\right)\right]. \tag{7.5}$$

Indeed, we do not have full knowledge concerning the responsibilities of the single components in generating the data. However, we have a probabilistic estimate of the responsibility of each latent aspect, that is given by the posteriors of the indicator variables Z_{ilk} and T_{kjp}. Such posteriors are estimated in the E-Step of the EM algorithm: for the region-topics, this yields to

$$\begin{aligned} E[Z_{ilk}] = P(z_k|d_i, r_l) &= \frac{P(z_k|d_i)P(r_l|z_k)}{\sum_{k'=1}^{K} P(z_{k'}|d_i)P(r_l|z_{k'})} \\ &= \frac{P(z_k|d_i)\prod_{j=1}^{W_l}[\sum_{t=1}^{T} P(t_p|z_k)\phi_{jp}]^{n_{ilj}}}{\sum_{k'=1}^{K} P(z_{k'}|d_i)\prod_{j=1}^{W_l}[\sum_{t=1}^{T} P(t_p|z_{k'})\phi_{jp}]^{n_{ilj}}} \end{aligned} \tag{7.6}$$

where the region-topic probability $P(r_l|z_k)$ has been expanded using the BOW assumption. Similarly, we compute the word-topic posterior as

$$E[T_{kjp}] = P(t_p|z_k, w_j) = \frac{P(t_p|z_k)P(w_j|t_p)}{\sum_{p'=1}^{K} P(t_{p'}|z_k)P(w_j|t_{p'})}. \tag{7.7}$$

The posteriors in (7.6) and (7.7) are used in the M-Step to compute the current estimates of the model parameters θ. To obtain the learning equations for θ, one has to maximize the expected log-likelihood

$$E[\mathcal{L}_{\mathcal{C}}(\mathcal{N}|\theta)] = \sum_{i=1}^{D}\sum_{l=1}^{R_i}\sum_{k=1}^{K} P(z_k|d_i, r_l)\left[\log(P(z_k|d_i)) + \sum_{j=1}^{W_l} n_{ilj}\sum_{p=1}^{T} P(t_p|z_k, w_j)\log\left(P(t_p|z_k)\phi_{jp}\right)\right] \tag{7.8}$$

with respect to the model parameters. In order to enforce the probability normalization constraints, one needs to extend the formulation in (7.8)

with proper Lagrange multipliers ρ_i, λ_k and τ_p, yielding

$$\begin{aligned}\mathcal{H} = E[\mathcal{L}_{\mathcal{C}}(\mathcal{N}|\theta)] + \sum_{i=1}^{D} \rho_i \left(1 - \sum_{k=1}^{K} P(z_k|d_i)\right) \\ + \sum_{k=1}^{K} \lambda_k \left(1 - \sum_{p=1}^{T} P(t_p|z_k)\right) + \sum_{p=1}^{T} \tau_p \left(1 - \sum_{j=1}^{W} \phi_{jp}\right).\end{aligned} \tag{7.9}$$

Maximization of $\mathcal{H}$ is obtained by differentiating (7.9) with respect to the model parameters θ, which yields to the following system of stationary equations

$$\frac{\partial \mathcal{H}}{\partial P(z_k|d_i)} = 0 \Longleftrightarrow \sum_{l=1}^{R_i} P(z_k|d_i, r_l) - \rho_i P(z_k|d_i) = 0, \qquad \begin{matrix} 1 \leq i \leq D \\ 1 \leq k \leq K \end{matrix}$$

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial P(t_p|z_k)} = 0 \Longleftrightarrow \sum_{i=1}^{D} \sum_{l=1}^{R_i} P(z_k|d_i, r_l) \sum_{j=1}^{W_l} n_{ilj} P(t_p|z_k, w_j) \\ - \lambda_k P(t_p|z_k) = 0, \quad \begin{matrix} 1 \leq p \leq T \\ 1 \leq k \leq K \end{matrix}\end{aligned}$$

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial \phi_{jp}} = 0 \Longleftrightarrow \sum_{i=1}^{D} \sum_{l=1}^{R_i} \sum_{k=1}^{K} n_{ilj} P(z_k|d_i, r_l) P(t_p|z_k, w_j) \\ - \tau_p \phi_{jp} = 0, \quad \begin{matrix} 1 \leq p \leq T \\ 1 \leq j \leq W \end{matrix}.\end{aligned}$$

After eliminating Lagrange multipliers, resolving the system above yields the M-step update equations for the model parameters, that are

$$P(z_k|d_i) = \frac{\sum_{l=1}^{R_i} P(z_k|d_i, r_l)}{\sum_{k'=1}^{K} \sum_{l'=1}^{R_i} P(z_{k'}|d_i, r_{l'})} = \frac{\sum_{l=1}^{R_i} P(z_k|d_i, r_l)}{R_i} \tag{7.10}$$

where we have used the equality $\sum_{k'=1}^{K} P(z_{k'}|d_i, r_{l'}) = 1$ from (7.6),

$$\begin{aligned}P(t_p|z_k) &= \frac{\sum_{i=1}^{D} \sum_{l=1}^{R_i} P(z_k|d_i, r_l) \sum_{j=1}^{W_l} n_{ilj} P(t_p|z_k, w_j)}{\sum_{p'=1}^{T} \sum_{i'=1}^{D} \sum_{l'=1}^{R_{i'}} P(z_k|d_{i'}, r_{l'}) \sum_{j'=1}^{W_{l'}} n_{i'l'j'} P(t_{p'}|z_k, w_{j'})} \\ &= \frac{\sum_{i=1}^{D} \sum_{l=1}^{R_i} P(z_k|d_i, r_l) \sum_{j=1}^{W_l} n_{ilj} P(t_p|z_k, w_j)}{\sum_{i'=1}^{D} \sum_{l'=1}^{R_{i'}} P(z_k|d_{i'}, r_{l'}) n_{i'l'.}}\end{aligned} \tag{7.11}$$

given that $\sum_{p'=1}^{T} P(t_{p'}|z_k, w_{j'}) = 1$ and where $n_{i'l'\cdot}$ is the number of visterms in region $r_{l'}$ of image $d_{i'}$; finally,

$$\phi_{jp} = \frac{\sum_{i=1}^{D}\sum_{l=1}^{R_i}\sum_{k=1}^{K} n_{ilj} P(z_k|d_i, r_l) P(t_p|z_k, w_j)}{\sum_{i'=1}^{D}\sum_{l'=1}^{R_{i'}}\sum_{k'=1}^{K}\sum_{j'=1}^{W_{l'}} n_{i'l'j'} P(z_{k'}|d_{i'}, r_{l'}) P(t_p|z_{k'}, w_{j'})}. \quad (7.12)$$

Parameter fitting proceeds as in the original pLSA model (Hof01), alternating between posterior estimation in the E-Step and model update in the M-Step. Algorithm 4 summarizes the HRPLSA parameter learning: notice that the stopping condition is based on the convergence of the likelihood estimate, which can be computed either on training data $\mathcal{N}$ or with respect to hold-out samples $\mathcal{N}_{valid}$ to prevent model overfitting (*early stopping*) (Hof01).

The probability estimates determined by the HRPLSA model can be used to perform Bayesian inference on the visual content. As regards the estimation of the most likely topic within an image d_i, this can be easily done as in (SRE+05), by finding the predominant aspect z^i as

$$z^i = \arg\max_{z_k \in \mathcal{Z}} P(z_k|d_i). \quad (7.13)$$

Compared with pLSA, the HRPLSA model allows also to determine a semantic segmentation of the image regions by estimating the most likely latent aspect z^* for each segment r_l, that is

$$z^* = \arg\max_{z_k \in \mathcal{Z}} P(z_k|d_i, r_l). \quad (7.14)$$

On the other hand, in order to determine the most likely aspect t^* of a visterm w_j in d_i, HRPLSA needs first to determine the topic z^* of the region r_l that encloses w_j, so that it can infer t^* by seeking

$$t^* = \arg\max_{t_p \in \mathcal{T}} P(t_p|z^*, w_j). \quad (7.15)$$

In other words, different region topics might induce different word topics associations even within the same image. Seen from the point of view of textual semantic analysis, this implies that the meaning of a given word can potentially vary depending on the context of the surrounding text,

Algorithm 4 HRPLSA Model Fitting

Randomly initialize $P(z_k|d_i)$, $P(t_p|z_k)$ and ϕ_{jp}
repeat
for all $n_{ilj} \in \mathcal{N}$ and $k \in \{1, \ldots, K\}$ **do**

$$P(z_k|d_i, r_l) = \frac{P(z_k|d_i) \prod_{j=1}^{W_l} [\sum_{t=1}^{T} P(t_p|z_k)\phi_{jp}]^{n_{ilj}}}{\sum_{k'=1}^{K} P(z_{k'}|d_i) \prod_{j=1}^{W_l} [\sum_{t=1}^{T} P(t_p|z_{k'})\phi_{jp}]^{n_{ilj}}}$$

end for
for all $j \in \{1, \ldots, W\}$, $k \in \{1, \ldots, K\}$ and $p \in \{1, \ldots, T\}$ **do**

$$P(t_p|z_k, w_j) = \frac{P(t_p|z_k)P(w_j|t_p)}{\sum_{p'=1}^{K} P(t_{p'}|z_k)P(w_j|t_{p'})}$$

end for
for all $i \in \{1, \ldots, D\}$ and $k \in \{1, \ldots, K\}$ **do**

$$P(z_k|d_i) = \frac{\sum_{l=1}^{R_i} P(z_k|d_i, r_l)}{R_i}$$

end for
for all $p \in \{1, \ldots, T\}$ and $k \in \{1, \ldots, K\}$ **do**

$$P(t_p|z_k) = \frac{\sum_{i=1}^{D} \sum_{l=1}^{R_i} P(z_k|d_i, r_l) \sum_{j=1}^{W_l} n_{ilj} P(t_p|z_k, w_j)}{\sum_{i'=1}^{D} \sum_{l'=1}^{R_{i'}} P(z_k|d_{i'}, r_{l'}) n_{i'l'.}}$$

end for
for all $p \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, W\}$ **do**

$$\phi_{jp} = \frac{\sum_{i=1}^{D} \sum_{l=1}^{R_i} \sum_{k=1}^{K} n_{ilj} P(z_k|d_i, r_l) P(t_p|z_k, w_j)}{\sum_{i'=1}^{D} \sum_{l'=1}^{R_{i'}} \sum_{k'=1}^{K} \sum_{j'=1}^{W_{l'}} n_{i'l'j'} P(z_{k'}|d_{i'}, r_{l'}) P(t_p|z_{k'}, w_{j'})}$$

end for
Compute the new likelihood $\mathcal{L}_{\mathcal{C}}(\mathcal{N})_t$
until Increase in likelihood $\mathcal{L}_{\mathcal{C}}(\mathcal{N})_t - \mathcal{L}_{\mathcal{C}}(\mathcal{N})_{t-1} < \epsilon$
return $P(z_k|d_i)$, $P(t_p|z_k)$, ϕ_{jp}, $P(z_k|d_i, r_l)$ and $P(t_p|z_k, w_j)$

i.e. the region topic. Hence, the HRPLSA model can naturally account for the presence of polysemous words within the same document, by disambiguating the topic depending on the context of the term. The pLSA model, on the other hand, determines the word-topic distribution on a per-document basis: hence, the same word recurring multiple times in a document will always be associated with the same topic. In Section 7.5, we will show how HRPLSA can effectively be used for the unsupervised discovery and segmentation of multiple visual classes within the same image.

Probabilistic inference can be extended to images outside the training corpora by using the *folding-in* technique described in (SRE+05). Given an unseen image d_{test} described as a bag of regions, we can determine the document specific mixing coefficients $P(z_k|d_{test})$, as well as the posteriors $P(z_k|d_{test}, r_l)$ and $P(t_p|z_k, w_j)$, by running Algorithm 4 using fixed word-topic and topic-topic distributions, i.e. ϕ_{jp} and $P(t_p|z_k)$, estimated using the training data $\mathcal{N}_{train}$. In other words, ϕ_{jp} and $P(t_p|z_k)$ are kept fixed through the whole EM process while the other distributions are re-estimated for the new image.

The generative model described in (7.1) reflects an *asymmetric* formulation of the latent semantic model (Hof99). However, by straightforward considerations on the Bayes'rule, (7.1) can be rewritten following a *symmetric* description (Hof99), that is

$$P(d_i, r_l) = \sum_{k=1}^{K} P(z_k)P(d_i|z_k)P(r_l|z_k). \tag{7.16}$$

This re-parametrization of the original problem highlights the symmetry of word-region generation process, since both d_i and r_l are generated from the latent topic z_k. Some authors (GGPC02; VG02) have argued that one formulation might offer advantages with respect to the other, although there is no consensus regarding which could be the best performing solution. For the sake of completeness, we point out that the asymmetric HRPLSA model described in Algorithm 4 can be straightforwardly

transformed into a symmetric one by estimating the region posterior as

$$P(z_k|d_i, r_l) = \frac{P(z_k)P(d_i|z_k)\prod_{j=1}^{W_l}[\sum_{t=1}^{T} P(t_p|z_k)\phi_{jp}]^{n_{ilj}}}{\sum_{k'=1}^{K} P(z_{k'})P(d_i|z_{k'})\prod_{j=1}^{W_l}[\sum_{t=1}^{T} P(t_p|z_{k'})\phi_{jp}]^{n_{ilj}}} \quad (7.17)$$

while the M-Step computation for $P(z_k|d_i)$ is substituted by

$$P(d_i|z_k) = \frac{\sum_{l=1}^{R_i} P(z_k|d_i, r_l)}{\sum_{i'=1}^{D}\sum_{l'=1}^{R_{i'}} P(z_k|d_{i'}, r_{l'})} \quad (7.18)$$

and

$$P(z_k) = \frac{\sum_{i=1}^{D}\sum_{l=1}^{R_i} P(z_k|d_i, r_l)}{\sum_{k'=1}^{K}\sum_{i'=1}^{D}\sum_{l'=1}^{R_{i'}} P(z_{k'}|d_{i'}, r_{l'})}. \quad (7.19)$$

Besides, in the remainder of the chapter, we will deal solely with the asymmetric HRPLSA formulation described in Algorithm 4.

7.3.2 Modeling Images and Words with Hierarchical Region-Topic Probabilistic Latent Semantic Analysis

In the previous section, we have introduced HRPLSA as a tool for modeling the visual content of an image collection as a multi-resolution hierarchical mixture of latent aspects. In this section, we extend it to model the association between visual content and related textual information. In particular, we show how HRPLSA learns to attach textual labels to the image regions by relying only on weak annotations available solely at the level of the whole image.

The HRPLSA annotation model, HRPLSA-ANN in short, takes inspiration from the *Asymmetric PLSA Learning* (APL) introduced by Monay and Gatica-Perez (MGP07) and described in brief in Section 6.4.2. Essentially, APL fits a pLSA model to the visual modality alone, obtaining a document-topic distribution $P(z|d)$ as well as a distribution of visual features conditioned on the latent aspects, i.e. $P(w|z)$. Then, it estimates the conditional distribution of annotations a given the topics z, i.e. $P(a|z)$, while keeping $P(z|d)$ fixed from the visual modality, likewise in the folding-in process. The annotation of an un-captioned image is obtained by probabilistic inference and provides textual labels at the level of

the whole scene. Notice that the same process can be applied in a reversed fashion, e.g. by first fitting the textual annotations and then folding in the visual content (i.e. the pLSA-WORDS model in (MGP07)).

The HRPLSA-ANN model uses an asymmetric learning approach, but learns to generate annotations for multiple levels of the topic hierarchy and of the spatial pyramid. For instance, lets consider the output of Algorithm 4 when applied to an image collection $\mathcal{N}$: at convergence, HRPLSA provides a document-topic distribution $\overline{P}(z_k|d_i)$, a topic-topic distribution $\overline{P}(t_p|z_k)$, a word-topic distribution $\overline{\phi}_{jp}$, a region-topic posterior $\overline{P}(z_k|d_i, r_l)$ and a word-topic posterior $\overline{P}(t_p|z_k, w_j)$. The overline sign on a distribution, e.g. in $\overline{\phi}_{jp}$, indicates that this term is held fixed for the textual modality. Now, represent each image in $\mathcal{N}$ as a vector of annotation occurrences m_{fi}, that is the number of instances of term a_f within document d_i. By asymmetric learning, we can fit the HRPLSA-ANN model to provide the probability of an annotation being generated by a region topic, that is $P(a_f|z_k)$. To do so, we perform an EM optimization to maximize the following *constrained annotation likelihood*

$$\mathcal{L}_{cal}(\mathcal{N}|\theta') = \sum_{i=1}^{D}\sum_{f=1}^{F} m_{fi} \log \sum_{k=1}^{K} P(a_f|z_k)\overline{P}(z_k|d_i). \tag{7.20}$$

The term *constrained* refers to the fact that the document topic distribution $\overline{P}(z_k|d_i)$ is inherited from the visual modality and is not updated by the M-step of the textual modality.

Algorithm 5 summarizes the optimization process for attaching annotations to the region-topics z_k. Given the estimated annotation-topic probability $P(a_f|z_k)$, we can annotate an un-captioned image d_{new} by means of the following probabilistic estimate

$$P(a_f|d_{new}) = \sum_{k=1}^{K} P(a_f|z_k)P(z_k|d_{new}),\ 1 \leq f \leq A \tag{7.23}$$

where the word-document distribution $P(z_k|d_{new})$ is generated by the visual modality and the text-topic probability $P(a_f|z_k)$ is the outcome of the EM process in Algorithm 5. Notice that the annotations provided by

Algorithm 5 HRPLSA-ANN model fitting at the region-topic level

Given $\overline{P}(z_k|d_i)$ from visual modality
Randomly initialize $P(a_f|z_k)$
repeat
E-STEP
for all $m_{fi} \in \mathcal{N}$ and $k \in \{1, \ldots, K\}$ **do**

$$P(z_k|d_i, a_f) = \frac{\overline{P}(z_k|d_i)P(a_f|z_k)}{\sum_{k'=1}^{K} \overline{P}(z_{k'}|d_i)P(a_f|z_{k'})} \tag{7.21}$$

end for
Constrained M-STEP
for all $f \in \{1, \ldots, A\}$ and $k \in \{1, \ldots, K\}$ **do**

$$P(a_f|z_k) = \frac{\sum_{i=1}^{D} P(z_k|d_i, a_f)}{\sum_{f'=1}^{A}\sum_{i'=1}^{D} P(z_k|d_{i'}, a_{f'})} \tag{7.22}$$

end for
Compute the constrained likelihood $\mathcal{L}_{cal}(\mathcal{N})_t$
until Increase in likelihood $\mathcal{L}_{cal}(\mathcal{N})_t - \mathcal{L}_{cal}(\mathcal{N})_{t-1} < \epsilon$
return $P(a_f|z_k)$ and $P(z_k|d_i, a_f)$.

(7.23) are still generated at the level of the whole image. However, by applying a similar inference process to the region-topic posterior, we can estimate the following

$$P(a_f|d_{new}, r_l) = \sum_{k=1}^{K} P(a_f|z_k)P(z_k|d_{new}, r_l),\ 1 \leq f \leq A \tag{7.24}$$

where $P(z_k|d_{new}, r_l)$ is estimated by HRPLSA from the visual modality. The estimate in (7.24) allows to determine the most likely annotation a^* for region r_l in image d_{new}, that is

$$a^* = \arg\max_{a_f \in \mathcal{A}} P(a_f|d_{new}, r_l). \tag{7.25}$$

The region annotation process described in (7.24) can be taken further by devising a generative annotation-topic model for the visterm aspects t_p. In other words, we consider the problem of attaching captions to a

visual word w_j characterized by a latent context (i.e. region topic) z_k; this is equivalent to perform the following probabilistic inference

$$P(a_f|z_k, w_j) = \sum_{p=1}^{T} P(a_f|t_p)P(t_p|z_k, w_j),\ 1 \leq f \leq A. \tag{7.26}$$

where $P(t_p|z_k, w_j)$ is estimated from the visual modality. Such word-topic posterior allows to performs a contextualized prediction of the most likely annotation for w_j since, coherently with the HRPLSA model, each visterm is associated to a latent aspect depending on the topic of the region it belongs to.

In order to compute (7.26), we need to estimate $P(a_f|t_p)$, that is the distribution of annotations given visterm-topics. To do so, we exploit asymmetric learning to fit the textual modality to the visual information. Algorithm 6 describes the EM optimization used to fit the $P(a_f|t_p)$ distribution. Notice that in (7.27) we have got rid of the $P(t_p|d_i)$ term by exploiting the HRPLSA independency assumption between word-topics and documents, i.e. given latent region aspects z_k, then visterm topics t_p and documents d_i are independent. Hence, by marginalizing the z_k variables we can substitute $P(t_p|d_i)$ with $\sum_{k=1}^{K} P(z_k|d_i)P(t_p|z_k)$.

Summarizing, the HRPLSA-ANN model allows to represent multimodal collections of images and associated text. In particular, annotations can be attached at different spatial resolutions (i.e. the whole scene, the single regions or the visual words) by varying the level of the topic hierarchy where captions are generated. Therefore, several approaches can be taken when modeling the association between visual and textual modality. For instance, we can learn solely coarse-level associations between regions (or images) and captions. Alternatively, we can learn the association between annotations and both regions and visterms: this can be achieved, for instance by using the same captions to estimate both $P(a|z)$ and $P(a|t)$. A more refined strategy, can exploit a word ontology such as *WordNet* (Fel98), to decide whether to generate annotations at the region-topic or at the word-topic level. The *LabelMe* collection and annotation tool (RTMF08) already provides a means for organizing the image annotations with respect to the WordNet ontology: however, this knowledge is

Algorithm 6 HRPLSA-ANN model fitting at the visterm-topic level

Given $\overline{P}(z_k|d_i)$ and $\overline{P}(t_p|z_k)$ from visual modality
Randomly initialize $P(a_f|t_p)$
repeat
E-STEP
for all $m_{fi} \in \mathcal{N}$ and $p \in \{1, \ldots, T\}$ **do**

$$P(t_p|d_i, a_f) = \frac{P(a_f|t_p) \sum_{k=1}^{K} \overline{P}(z_k|d_i)\overline{P}(t_p|z_k)}{\sum_{p'=1}^{T} P(a_f|z_{p'}) \sum_{k'=1}^{K} \overline{P}(z_{k'}|d_i)\overline{P}(t_{p'}|z_{k'})} \quad (7.27)$$

end for
Constrained M-STEP
for all $f \in \{1, \ldots, A\}$ and $p \in \{1, \ldots, T\}$ **do**

$$P(a_f|t_p) = \frac{\sum_{i=1}^{D} P(t_p|d_i, a_f)}{\sum_{f'=1}^{A} \sum_{i'=1}^{D} P(t_p|d_{i'}, a_{f'})} \quad (7.28)$$

end for
Compute the constrained likelihood $\mathcal{L}_{cal}(\mathcal{N})_t$
until Increase in likelihood $\mathcal{L}_{cal}(\mathcal{N})_t - \mathcal{L}_{cal}(\mathcal{N})_{t-1} < \epsilon$
return $P(a_f|t_p)$.

not used for driving image annotation, which is still performed manually.

Figure 65 shows an example of how WordNet can be employed in annotating an image from the LabelMe dataset. In particular, the ontology suggests that `window` is a *part meronym* of `building`: hence, we can hypothesize that the visual representation of the window is more likely to be generated by a visterm rather than a region. On the other hand, a building is more likely to be captured by a whole region. Therefore, we generate the `window` label from the visterm-topics while the region-topics take responsibility for the building caption.

In the remainder of the chapter, we evaluate the performance of the HRPLSA and HRPLSA-ANN models in image understanding tasks. Before delving into the details of the experimental evaluation, in the next section, we briefly review the relationships of HRPLSA with relevant re-

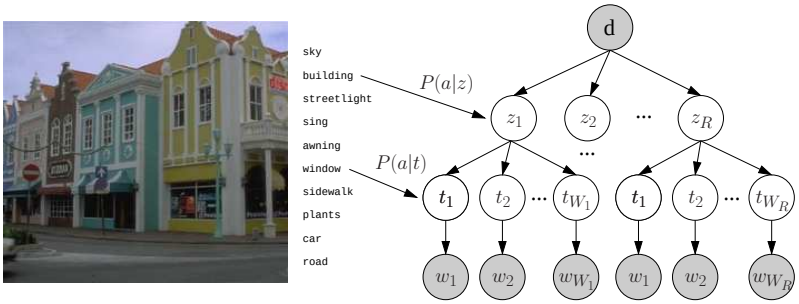


Figure 65: Example of hierarchical image annotation with the HRPLSA-ANN model: given an image and its associated annotation (both available on LabelMe (RTMF08)) we can use a word ontology (i.e. WordNet (Fel98)) to determine whether to assign captions to the region-aspects z or to the visterm-topics t. In this figure, the `window` caption is in a part-of (*part meronym*) relation with the `building`. Hence, we can hypothesize that the latter annotation is generated by a region topic (coarse scale), while the former is associated to a visterm topic (finer scale).

lated work in the literature.

7.4 Related Work

In the recent years, weakly structured approaches to image content understanding have acknowledged a notable increase in popularity, mostly due to the surprising performance of pLSA applications built on the top of a bag-of-words image representation. After this initial burst, attention has started to slowly drift towards more structured approaches, that are capable of retaining the flexibility and low computational complexity of BOW approaches, while introducing explicit spatial and contextual information into the visual model. The HRPLSA model fits into this scenario, since it tries to get past a purely BOW assumption by introducing a multilevel representation of the image comprising regions and visual words.

Between the earliest approaches introducing spatial information into latent semantic analysis are the ABS-pLSA and TSI-pLSA models (FFFPZ05). The former approach defines a discrete grid of image locations that is

used to quantize the visterm positions in a set of spatial bins; then, pLSA density estimate is performed jointly on visual word appearance and location. The TSI-pLSA model, instead, introduces a latent variable that provides an estimate of the object position under the form of a bounding box. Inside this box, the image is partitioned as in ABS-pLSA, while the outer part is considered as a unique bin. In the context of object recognition, (LC06) propose the *Semantic Shift* model, that is a modified form of pLSA that extends the BOW representation by adding localization information regarding the visual words position. Essentially, the document-word co-occurrence matrix becomes a document-word-position table $n(d; w; x)$, where x is the position of word w in document d. As a result, this model provides location and scale estimates of foreground objects, allowing to better separate background latent aspects. The models described so far, essentially focus on determining a suitable isolation of the object topics from the background: to do so, they resort to a joint modeling of word appearance and position in an absolute coordinate system. HRPLSA, on the other hand, does not assume any absolute positioning since this, for instance, might impair the detection of an object presented at consistently different scales. Moreover, HRPLSA accommodates for the presence of multiple objects within the same scene.

Recently, segmentation has been re-discovered as an useful object recognition tool. Russell et al (RFE$^+$06) introduced the idea of performing multiple segmentations of the same image, representing each detected region as a bag of visual words identified by interest point operators. In their approach, each segment is considered as a document d in a visual corpus $\mathcal{D}$ that is used to fit an LDA model. Therefore, the estimated latent topics can be used to label the single regions in the collection and a candidate image segmentation is chosen based on the similarity of the visual word distribution in the alternative segments with respect to the corresponding word-topic multinomial. This model *breaks down* images by treating each segment as an isolated entity, i.e. a document of the collection; in contrast, HRPLSA treats each image as one document characterized by a topic distribution that is inferred at the region-level rather than at word level. Moreover, HRPLSA does not need to perform multiple segmenta-

tions of the same image, since the final segmentation is produced straightforwardly by the region-topic assignment (in Section 7.5 we show an experimental comparison between the two approaches). The *spatially coherent latent topic* model (Spatial-LTM) takes a rather similar approach, where images are modeled as bag of regions within an LDA model and the latent topics are estimated at the level of picture segments. However, Spatial-LTM does not define a hierarchy of latent topics since the region aspects are estimated directly from the visual word distribution. In other words, Spatial-LTM *forces* all visual words within the same regions to share a common latent aspect, whereas HRPLSA allows the co-existence of multi-topic words, determining the region aspect based on the mixture of word-topics in the segment. In a sense, the HRPLSA approach accommodates naturally the semantical differences between visterms and regions. Consider, for instance, a urban scene: visterm topics can, eventually, learn to isolate window portions from concrete patches, while the region topic can isolate buildings; with the Spatial-LTM approach, window and building visterms would be forced to belong to the same topic.

Hierarchical latent semantic approaches have initially found wide application in text clustering, given that they provide a means for organizing documents in a hierarchy of clusters, e.g. HPLSA (GGPC02) and MASHA (VG02). Recently, there has started to be an increasing attention towards hierarchical representation of the visual content. The *Dependent Hierarchical Dirichlet Process* (DHDP) (WZFF06), for instance, exploits a hierarchical structure over the latent topics to encode visterms co-occurrence, facilitating better clustering of the latent aspects. With respect to HRPLSA, the DHDP models only the inter-dependencies between visual words, whereas topic correlation is totally disregarded. The *Pachinko Allocation Model* (DPAM) (LWG07), on the other hand, explicitly models the correlation between visual topics by means of a four level hierarchy of latent aspects: the leaves of the hierarchy are responsible for the generation of the local patches, while the internal nodes learn the correlation between the subtopics (represented as their child nodes): within this model, a local patch is generated by sampling a path from the root to the leaves of the hierarchy. The HRPLSA model, compared to DPAM, defines a less

articulated aspect hierarchy, but allows to model both the topic correlation (via the topic hierarchy) as well as the visterm co-occurrence (by the multi-layered image representation). DPAM, on the other hand, needs to be augmented with prior knowledge regarding the visual words interdependence, so that co-occurrent visterms are more likely to share the same topic.

Recently, (SRZ+08) adapted the *hierarchical Latent Dirichlet Allocation* (hLDA) model to the visual domain, with the intent of unsupervisedly discovering a hierarchy of object aspects, such that topics become increasingly specialized on the path from the root to the leaves of the hierarchy. This work extends the multiple segmentation framework in (RFE+06), so that the hLDA model learns not only the object category but also its segmentation. With respect to HRPLSA, the hLDA model still treats each image segment as an isolated document of the collection. The comparative analysis between HRPLSA, hLDA and the *flat* multiple segmentation model (RFE+06) is further developed in the following section, where we evaluate the performance of the models on a visual class segmentation task.

As a general comment, HRPLSA-ANN seems to be, to the best of our knowledge, the first hierarchical latent semantic model exploiting a multi-layered image representation for modeling visual content and related textual information. As we have already pointed out in Section 7.3.2, the asymmetric learning model (MGP07) provides only a *flat* approach, where captions are attached to the whole image and latent topics are responsible solely for visterm generation. We recall that in (MGP07) it is proposed an alternative annotation model, named pLSA-WORDS, that first fits a pLSA model to the textual modality and then folds-in the visual modality. The experimental results in (MGP07) show that this model achieves a slightly superior precision/recall performance with respect to the approach that we adopted for the HRPLSA-ANN. However, the pLSA-WORDS does not not fit naturally in our multi-resolution scenario since it allows to predict annotations only for the whole scene. On the other hand, with the HRPLSA-ANN model we aim at inferring a region-caption distribution that can be used to generate annotations for the sin-

gle regions of an image; hence, in the remainder of the chapter, we concentrate solely on the pLSA-FEATURE approach.

7.5 Unsupervised Discovery and Segmentation of Visual Classes

Before discussing the application of the HRPLSA-ANN model to automatic region annotation, we study the performance of the hierarchical latent topic model on visual information alone. In particular, we focus on evaluating HRPLSA performance on the unsupervised discovery and segmentation of visual classes from a small collection of images comprising articulated scenes depicting multiple object instances. To this end, we apply our approach to the Microsoft Research Cambridge dataset (MSRC-B1 (WaTM05)), containing 240 images with 9 visual classes, that are *airplanes*, *bicycles*, *buildings*, *cars*, *cows*, *faces*, *grass*, *tree* and *sky*. Each image in the dataset is provided with a, manually obtained, ground truth segmentation into the semantic classes. The dataset is reported to contain 4 additional visual entities, i.e. *horses*, *water*, *mountain* and *sheep*, which, however, have insufficient data support to yield to reasonable recognition.

In this section, we test HRPLSA performance with respect to the segmentation of the 9 MRSC-B1 classes. In particular, we study if the region-topics can successfully isolate the single visual classes, refining the coarse CoRe segmentation introduced in Section 7.2, by joining regions that share a common semantic interpretation (i.e. the same region aspect). The performance of the HRPLSA model is compared with two state of the arts methods, that are the flat multiple segmentation LDA (RFE$^+$06) and the hierarchical hLDA model (both discussed in Section 7.4).

The multi-resolution image representation is constructed following the indications given previously in the chapter: local patches are obtained by densely sampling 3000 points from the multi-resolution grid described in Section 7.2; a 2000 dimensional visual codebook has been constructed by running k-means clustering on the keypoint descriptors of 120 randomly selected images. Regions have been generated by CoRe clustering

initialized with 20 units, resulting in an average of 11 units per image.

The HRPLSA model has been fitted to the 240 images using different choices for the number of word and region topics (i.e. t_p and z_k, respectively). The results for flat LDA are provided for 10, 15, 20 and 25 word topics, while hLDA is based on a 5-level topic hierarchy (see (SRZ$^+$08) for the details.). Given GT_o, i.e. the ground truth segmentation for object o, and R_r the retrieved candidate region, the discovery/segmentation performance is computed by means of the region overlap

$$\rho_{or} = \frac{GT_o \cap R_r}{GT_o \cup R_r}. \tag{7.29}$$

Following the approach in (SRZ$^+$08), we obtain the object overlap ρ_o by averaging ρ_{or} over the top-5 retrieved images and for each object class we reported the score of the best performing topic. Table 13 shows the segmentation scores for each object/model averaged over multiple runs: LDA_{10}, stands for a flat LDA model with 10 topics, while $HPRLSA_{25}^{10}$ identifies an HRPLSA model with 25 word-topics and 10 region aspects. For the sake of completeness, in the last two rows of Table 13 we report the overlap score obtained by $\overline{HPRLSA}_{25}^{10}$ and $\overline{HPRLSA}_{25}^{15}$ (i.e. the best scoring models) over all the segments assigned to a single visual class. Figure 66 to Figure 68 show a mosaic of the top-5 image segmentations retrieved by $HPRLSA_{25}^{10}$.

On average, the HRPLSA models obtain the best results: in particular, the most consistent performance gain seems to be achieved for those aspects, i.e. *building*, *grass* and *sky*, that typically cover larger areas of the image. Due to their size, these regions tend to include within their boundaries visterms characterized by different topic assignments. Consider, for instance, a building region: it typically contains visual words that can be characterized by either a concrete/brick topic or by a glass/window aspect. This latter topic can be shared, for instance, with the *car* or even with *sky* classes. Since LDA assigns the region topic based on the most likely word aspect, the whole building segment has an high probability of being assigned to a shared glass/window topic, hence resulting in a poor LDA segmentation. HRPLSA, on the other hand, can isolate both the concrete and the glass topics at the word level, while modeling the

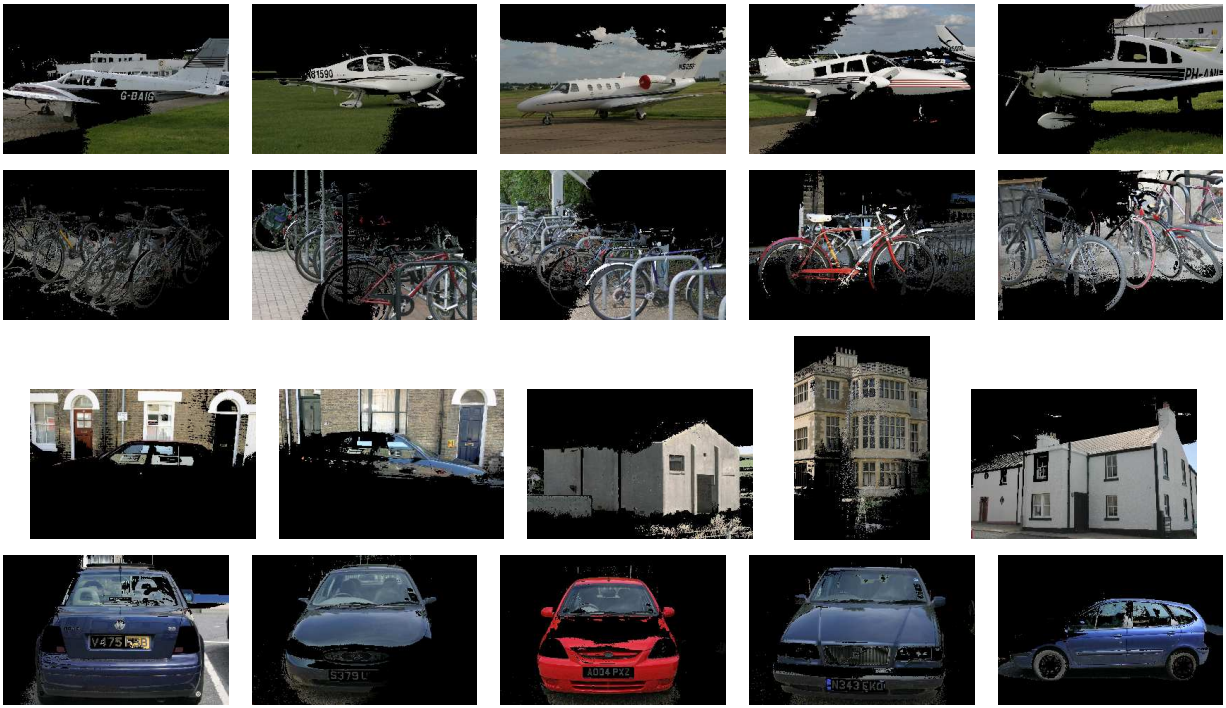


Figure 66: Example of retrieved segments for the MSRC classes: airplanes, bicycles, buildings and cars.

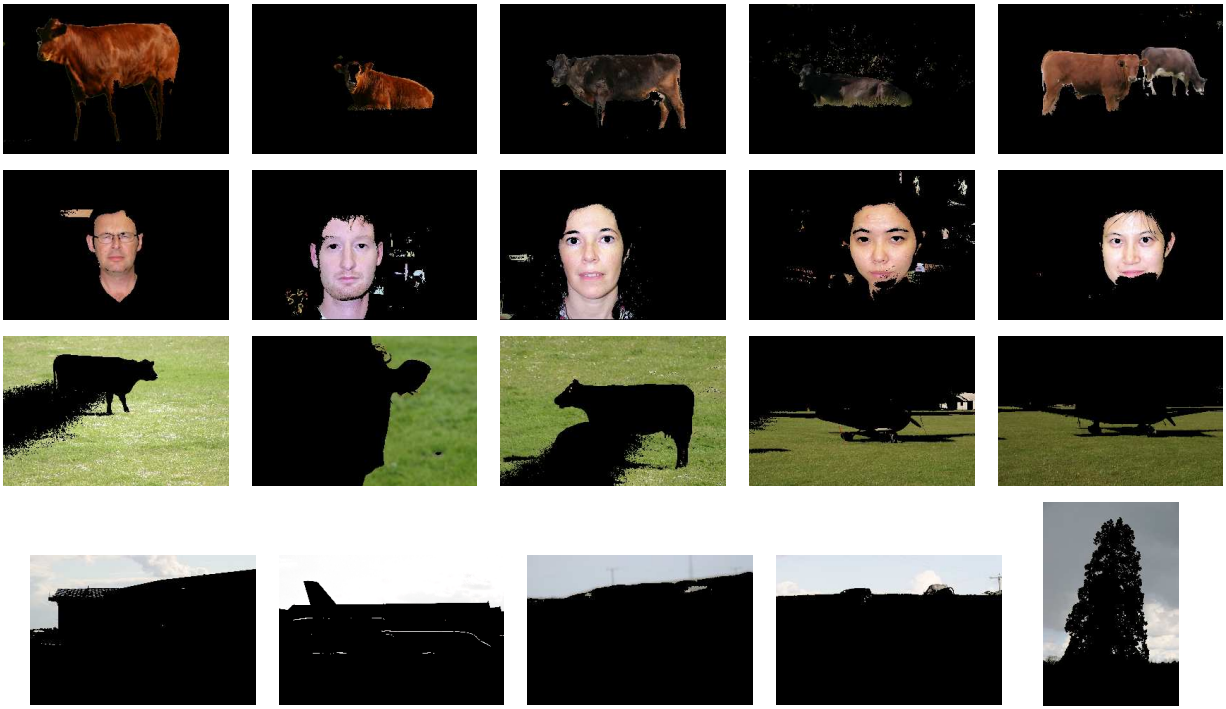


Figure 67: Example of retrieved segments for the MSRC classes: cows, faces, grass and sky.

Figure 68: Example of retrieved segments for the tree MSRC class.

higher-level building aspect as a mixture of concrete and glass topics. Notice that the hLDA topic hierarchy is conceptually different from that of HRPLSA: hLDA assigns the whole documents (i.e. the single image regions in the multi-segmentation framework) to the single topic/nodes of the hierarchy; hence, all the latent topics in the hierarchy are still generated at the level of the visterms. The superior performance of HRPLSA on segments covering large portions of the image is also motivated by the different approach to image representation. Both hLDA and flat LDA need to select one region from a soup of segments generated by multiple segmentations: if a region depicting a particular object class is split into multiple pieces in all the alternative segmentations, then hLDA and LDA cannot recover its original boundaries since they consider each segment as a separate document. HRPLSA, on the other hand, allows image regions to be split into multiple segments before merging them back into a unique region once that it has inferred the topic segmentation for the image. This motivates why hLDA, although hierarchical, has a poor overlap score for the building class. Regions containing building instances are typically fragmented, including several visual entities (e.g. doors, windows, signs, etc) that worsen the chances of obtaining a unique segment including the whole building. On the other hand, *grass sky* regions can be well isolated by color segmentation, resulting in a good overlap score

also for the hierarchical LDA model.

The visual classes where HRPLSA obtains a limited performance gain are *airplane*, *bicycle*, *face* and *tree*. The *airplane* class seems to be particularly difficult to detect for the flat LDA model that fails to discover good segment-topic assignments: hLDA and HRPLSA achieve consistently better results, although none of the two models obtains a notably superior performance with respect to the other. The visual impression in Figure 66 confirms that airplane segments discovered by $HRPLSA_{25}^{10}$ do not sharply isolate the object; whereas a notable performance is obtained for *building* and *cars* (third and fourth row in Figure 66), as well as for *cows*, *grass* and *sky* (see Figure 67). In particular, the results for the *cars* and *cows* classes (see Table 13) show that HRPLSA achieves better topic-segmentation quality with respect to hLDA and LDA: it is worth noting how the overlap score obtained for the totality of the object segments (i.e. $\overline{HRPLSA}_{25}^{10}$ and $\overline{HRPLSA}_{25}^{15}$) is very close to the results obtained by hLDA and LDA on the top-5 segments, denoting the fact that HRPLSA can stably detect good cars and cows instances.

By comparing the performance of HRPLSA for different numbers of latent topics (see Table 13), we notice that some aspects, such as *sky* and *grass* are well isolated independently from the size of the latent spaces $\mathcal{Z}$ and $\mathcal{T}$. These classes, together with the *cows* and *cars* categories, ensure that the average overlap score remains high for different choices of the initial topic number. The *airplanes* class, on the other hand, seems to be discovered best when the number of region topics z_k is in between 15 and 20. Airplane segments typically include noisy visterms from the background, e.g. grass, building and trees, which results in HRPLSA isolating spurious aspects where airplanes cannot be well isolated from the surrounding (see Figure 66). This issue affects primarily those models characterized by fewer region-topics, since the model is less likely to cluster airplane segments and noisy regions into two different aspects. Segmenting out the airplane class is, in general, an hard task also for those HRPLSA models using $15 - 20$ region topics since, typically, some parts of the airplane tend to be confused with background aspects. Figure 69 shows an example of topic segmentation obtained by $HRPLSA_{25}^{15}$ on im-

Figure 69: An example of topic segmentations of images containing airplanes ($HRPLSA^{15}_{25}$): left column shows the original images, center the CoRe segmentation and right the topic segmentation. The airplane aspect is shown in purple, while brown and light-blue identify sky and grass, respectively. Notice how consistent portions of the airplane are assigned to a background cluster (in yellow).

ages containing airplane instances: it can be seen that the model is able to isolate well the sky and grass aspects (shown in light-blue and brown, respectively), while part of the airplane is assigned to the background. A possible way to address this issue requires forcing CoRe to produce a fine-grained segmentation of the image (e.g. similarly to (CFF07)), hence producing smaller regions that are less likely to contain mixed airplane-background visterms.

As regards the *faces* class, it achieves the best isolation when the HRPLSA model comprises at least 20 word topics. By looking at the mixing proportions $P(t|z_{face})$ of the visterm-topics t in the face aspect z_{face}, we discover that topic z_{face} is characterized by at least 3 different word-topics t. Therefore, if the visterm collection is modeled with few t aspects, it can result in HRPLSA running out of word-topics, consequently failing to isolate a word-topic that is important for representing the z_{face} aspect.

By means of the HRPLSA model, we can produce semantically meaningful segmentations of an image collection. In a sense, CoRe learning generates low-level hypotheses concerning the existence of meaningful

Figure 70: An example of topic segmentation obtained by $HRPLSA^{10}_{25}$: each color identifies a different image segment (center) or topic (right).

local entities, while the hierarchical topic model refines them by providing an unsupervised top-down signal that aggregates semantically homogenous regions. Consider for instance the topic segmentation in Figure 70: the HRPLSA model is able to aggregate the perceptually uniform regions generated by CoRe (shown in the center) into meaningful topic-segments (shown on the right). The dark-red color, for instance, isolates the grass topic while the gray color identifies cows. Notice how the sheep instance in the second row is assigned to the cow topic, since HRPLSA has insufficient information for discriminating between the two animal kinds. Moreover, in both images, part of the grass segments (typically corresponding to the weed) are erroneously assigned to the tree topic (brown color in Figure 70). Also, part of the animals in the top-row image is assigned to the face topic (light-red color): this is not unexpected since the original CoRe region (center) includes all the cow heads; hence, its visterm distribution includes a large percentage of face-related keypoints (e.g. eyes, mouth, etc.).

So far, we have studied the performance of our model as a fully unsupervised approach to visual content discovery on a dataset characterized by a small number of visual classes. Indeed, since the whole process is completely unsupervised it cannot infer the full semantics underlying

the visual information (especially as the number of aspects increases), although HRPLSA achieves a segmentation performance that is superior to both hLDA and flat LDA, that are the state-of-the-art models in this respect. In the next section, we approach the problem of exploiting weakly supervised information provided by image captions to extend the visual content discovery to more articulated image collections comprising hundreds of visual aspects.

7.6 Image Annotation

The experiments in this section aim at evaluating the effectiveness of the multi-resolution hierarchical extension of pLSA in modeling collections of annotated images. The performance of the flat and hierarchical latent aspect models is first tested on the University of Washington Ground Truth dataset[1], which contains 857 publicly available images that have been manually annotated with 392 words. Each image caption contains between 1 and 20 annotations, resulting in average of 5.8 words per picture. As noted in (THL06), the dataset contains several annotation mistakes as well as mixed singlurar/plural forms: after error correction and stemming we reduce the original vocabulary size to 245 words. The frequency distribution of the words across the dataset is shown in Figure 71: notice how the frequency is distributed quite unevenly between the words, with several terms occurring only once in the dataset (e.g. the *zen* word). As a general idea, we choose to keep the level of keyword preprocessing to the minimum, in order to test the robustness of latent space annotation with respect to noisy dictionaries: however, to allow a better comparison with the works in the literature (THL06; HSSST06), we provide also results for *cleaner* vocabularies including only terms occurring more than once (i.e. 188 words) and more than twice (i.e. 158 terms).

Images have been processed, as detailed in Section 7.5, to extract a set of 2000 SIFT descriptors using dense grid sampling : 427 randomly selected images have been used to build a 1000-dimensional codebook by k-means clustering. As noted in a recent review on image annotation

[1] `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`

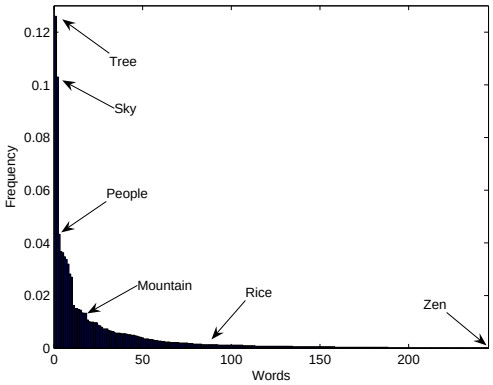


Figure 71: Word frequency distribution for the Washington Ground Truth dataset.

in semantic spaces (HSLN08), the codebook size, whenever is is larger than 500, has a negligible effect on the maximally achievable precision of the method. For practical purposes, we have chosen a 1000-dimensional codebook since it is characterized by a fair tradeoff between computational efficiency and discriminative power (in particular for the vector-space baseline). Regions have been generated by CoRe clustering initialized with 20 units, resulting in an average of 11 regions per image, with the minimum number being 5 and the maximum 18. Training and test images are selected by dividing the dataset into two equally-sized random partitions (427 images).

In the experiments, we perform a direct comparison between the proposed hierarchical HRPLSA-ANN model and the flat pLSA-FEATURE annotation approach introduced in (MGP07). Moreover, following the experimental setup in (MGP07), we validate the performance of the two models against a baseline vector space model, that is *Annotation Propagation* (AP). This approach is intuitively very simple: it computes the similarity of an un-captioned image d_{new} with the images in the training set; annotations are *propagated* from the training images to d_{new} based on the cosine similarity between the visual descriptors (see (MGP07) for further

details). Besides its simplicity, this method is reported to perform well, especially on datasets, like Washington, characterized by images with a large degree of visual similitude (THL06; MGP07).

The performance of the methods is evaluated based on the *precision* and *recall* measures, defined as

$$pre_i = \frac{r_i}{(r_i + w_i)} \tag{7.30}$$

and

$$rec_i = \frac{r_i}{n_i} \tag{7.31}$$

where r_i is the number of correctly predicted words for image i, w_i is the number of wrongly predicted captions and n_i is the actual number of words in the image. Indeed, the best model will be characterized by the highest precision, i.e. the ratio of correct annotations with respect to the number of predicted words, at the highest recall, i.e. the total ratio of correctly predicted captions. The *normalized score* proposed in (BDdF$^+$03) is another popular measure for evaluating the quality of keyword prediction: however, (MGP03) show that it does not sufficiently weight incorrect word responses, since it tends not to take into consideration the added noise given by incorrect guesses once that all the right captions has been generated.

Figure 72 shows the average precision/recall curves obtained by the three models for different sizes of the annotation vocabulary: the curves have been obtained by increasing the number of predicted annotations from 1 to 9. Overall, the HRPLSA-ANN model obtains the best precision-recall results when predicting $1-5$ captions, with $\sim 79\%$ of correct predictions when a single term is generated for each image. On the other hand, the performance of the two latent models is essentially equivalent when generating $7-9$ annotations. In particular, in this latter interval, the best performance is obtained by the annotation propagation method: this is not unexpected since, as we have noted before, the Washington dataset contains very similar images, that is the best scenario for the annotation propagation method. Additionally, the particularly good performance of this method is coherent with the results presented in (THL06), where a

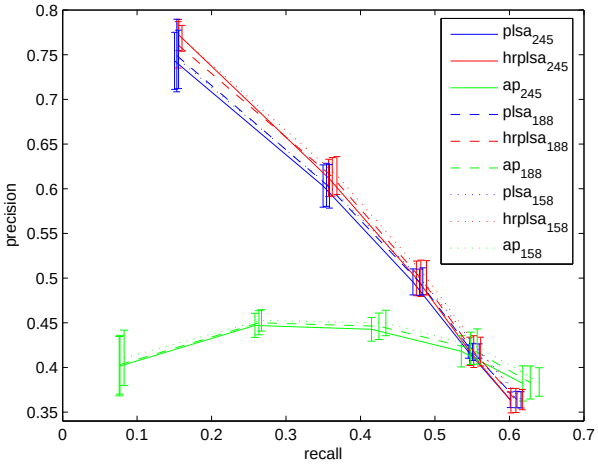


Figure 72: Precision/recall curves for the Washington dataset: the plot shows the performance of annotation propagation (ap), pLSA-FEATURES (plsa)and HRPLSA-ANN (hrplsa) for different sizes of the vocabulary (shown as footer number).

closely related annotation propagation approach yields similar results on the Washington data. The size of vocabulary does not seem to influence much the performance of the three models, given that the variations of the precision/recall remain bounded within the respective error bars.

Table 14 summarizes the precision/recall results obtained on the Washington dataset by relevant annotation models in the literature. For instance, (HL05) performs image annotation using Latent Semantic Indexing (LSI) on images represented as bags of salient regions, while (HSSST06) applied Kernel Canonical Correlation Analysis (KCCA) to retrieve and annotate pictures, again represented as histograms of visterms. The Washington dataset has also been used in (THL06) to confront two Cross Media Relevance Models (CMRM), one using a blob-based image representation and the other one relying on salient regions extracted by Lowe's interest point detectors (Low99). Notice that a direct comparison of the results in Table 14 needs to take into consideration the fact that results

have been obtained for different vocabulary sizes (i.e. given by different clean-up processes); hence, Table 14 reports the name of the models together with the size of the word vocabulary used in the experimentation. A comparison of the results obtained by our latent approach shows that HRPLSA-ANN achieves a comparable performance with respect to the saliency-based CMRM (THL06), that is the state-of-the-art model on the Washington dataset. In particular, HRPLSA-ANN yields to an higher precision at price of a lower recall, which can be motivated by the difference in the term vocabulary: for instance, the average caption size in our experiments is 5.8, while (THL06) reports an average annotation length of 4.8 keywords. With respect to the region-based CMRM, the HRPLSA-ANN model obtains a consistent performance increase, irrespectively of the size of its vocabulary of captions.

Figure 73 shows an example of Washington images annotated by the HRPLSA-ANN, pLSA-FEATURES and annotation propagation. In general, both latent topic models can recall with an high precision those words associated to latent aspects that are predominant in the image collection, e.g. *tree*, *building*, *sky*, etc, and that, in a sense, are those isolating the broader visual concepts in the data. This ability provides them with an high baseline performance when predicting few terms, since they can correctly guess the most general words describing the images. Since these words are those most likely to occur across several pictures, latent topic models results in a high overall performance. Annotation propagation, on the other hand, can recall words that have a low overall frequency but a sufficient presence within a cluster of similar images: for instance, the *lines* term (i.e. denoting electric lines) detected for the sixth image in Figure 73 is not frequent in general, but appears sufficiently often in a number of house images. The HRPLSA-ANN seems to have an advantage over the flat pLSA model since it can recall less general captions that are connected with peculiar distributions of visterms within a spatial locality, i.e. that can be well isolated by region aspects. An example is the *waterfall* theme shown in the fourth image in Figure 73, or the *cherry* aspect depicted in the fifth image. In pLSA, the latent aspect connected with these words might not stand out from the crowd of topics associ-

ated to the visterms present in the whole image, while in HRPLSA it will be characterizing the spatial locality, i.e. the picture segment, where it occurs.

As described in Section 7.3.2, the HRPLSA-ANN model provides a caption-topic distribution $P(a|r,d)$ that can be used to label the single image regions r with the most likely caption a^*. Figure 74 and 75 show an example of the annotated segments discovered by HRPLSA-ANN on the Washington images. In some cases, the model is able to learn to separate semantically related captions such as *sky* (Figure 75.b) from *clear sky* (Figure 74.c), and *clouds* (Figure 74.d) from *overcast* (Figure 75.h). In other cases, such as for *building* (Figure 75.1) and *house* (Figure 74.e), there is a certain overlap between the visual concepts captured by the two annotations. Figure 76 shows some good examples of HRPLSA-ANN region annotation, while Figure 77 points out typical labeling mistakes. Some limitations to the model performance rise as a consequence of the SIFT image representation: for instance, segments labeled as containing trees may also contain instances of flowers, given the difficulty of distinguishing between the two visual concepts relying solely on gray level SIFT information (see Figure 77.a). Similarly, aspects relating to sky and water tend to overlap (see 77.b), resulting in water regions being labeled as containing sky. In this cases, the keyword water is usually the second most likely caption with respect to $P(a|r,d)$). Some dataset objects, e.g. *people* and *cars*, are too small for being detected and segmented out as isolated entities. Typically, they tend to be included into cluttered regions comprising portions of their background, resulting in noisy aspects that produce less reliable annotations. In Figure 77.c, for instance, some image portions are labeled as *people* since this keyword tends to co-occur frequently with the *house* term: hence, the discovered word-topic association is likely to capture the occurrence of human figures on a house background. Conversely, the *woman* keyword (see segments in 75.d) is associated to nicely isolated regions containing man/woman occurrences, thanks to the well defined contrast between the human figures and the flat background.

The second set of experiments is performed on a LabelMe (RTMF08)

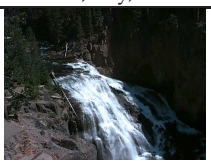
			
Original	boat, clouds, mast, sky, small tree, water	building, bush flower, ground, sidewalk, tree	clouds, sky tree, water
HRPLSA	boat, clouds, sky, tree, water	building, bush flower, grass, tree	clouds, sky mountain, tree, water
plSA	boat, clouds sky, tree, water	bush, flower grass, sky, tree	boat, clouds sky, tree, water
AP	boat, clouds mast, sky, tree	building, bush clear, flower, sky	building, clouds sky, small, tree
			
Original	tree, rock waterfall	cherry, clouds house, sky, tree	car, flower, house lines, overcast, pole sky, structures, white
HRPLSA	mountain, rock, sky, tree, waterfall	cherry, clouds, grass, sky, tree	clouds, house, people, sky, tree
plSA	clear, mountain, rock, tree, sky	bush, clouds, grass, sky, tree	clouds, people, sky, tree, water
AP	canyon, clouds, rock, sky, tree	building, bush, cherry, grass, tree	clouds, lines, house, flag, overcast

Figure 73: Annotation examples from the Washington datatset.

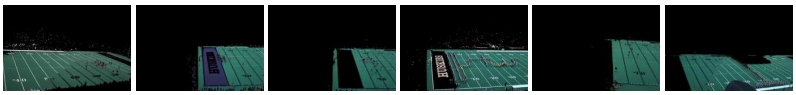
(a) Building

(b) Bush

(c) Clear Sky

(d) Clouds

(e) House



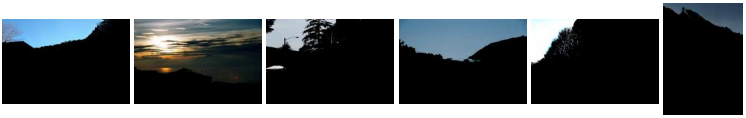
(f) Husky Field

(g) Mountain

(h) Overcast

Figure 74: An example of annotated Washington segments.

(a) Rocks

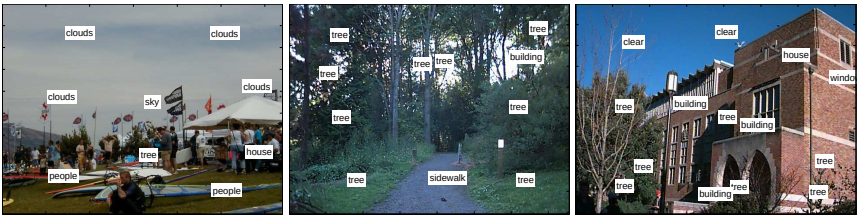


(b) Sky

(c) Tree

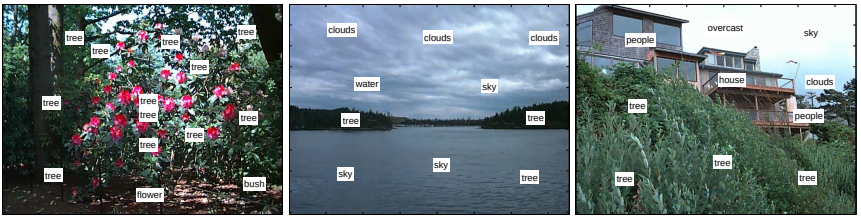
(d) Woman

Figure 75: An example of annotated Washington segments.



(a) (b) (c)

Figure 76: Examples of good HRPLSA annotations.



(a) (b) (c)

Figure 77: Examples of bad HRPLSA annotations.

benchmark dataset[2] comprising 2688 color images (256×256 pixels in size) from eight outdoor scene categories, i.e. coast, mountain, forest, open country, street, inside city, tall buildings and highways. The challenge proposed by the authors of the collection requires to try to recognize and segment as many object categories as possible, randomly selecting 100 images for each of the 8 scene categories to form the training set (giving a total of 800 training images), while using the remaining 1888 as a test-set. This is a challenging task that, to the extent of our knowledge, has never been undertaken by any of the object recognition models in the literature, although this dataset has been widely used to evaluate the performance of scene recognition approaches (BZM08). Given that figures are provided with manual annotations, we can approach it as an image annotation task: first, we cleaned up the initial 400 terms by removing annotation errors and merging plurals, obtaining a final vocabulary of 309 keywords. With respect to the Washington dataset, this image collection has a challenging number of images with a relatively large vocabulary size (see (Han08) for a recent survey on annotated datasets). Figure 78 shows the word distribution for the dataset: notice how the the frequency is distributed even more unevenly than in the Washington dataset, with a long tail of terms with a single occurrence within the collection ($\sim 28\%$). On the other hand, the distribution of the caption length in Figure 79 shows that most of the images are labeled by ~ 10 terms, hence confirming that there are certain *general* captions that occur very often across the

[2] `http://labelme.csail.mit.edu/`

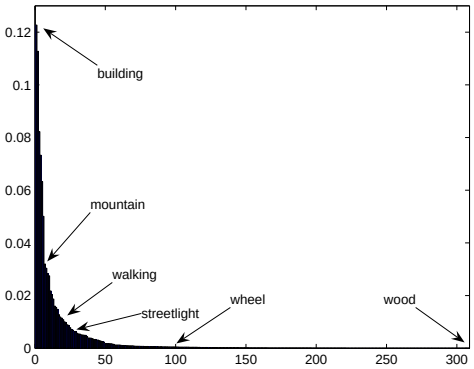


Figure 78: Word frequency distribution for the LabelMe dataset.

image collection.

As for the Washington dataset we densely sampled 2000 SIFT descriptors per image, randomly selecting 100 images from each of the 8 scene categories, to generate a 1000-dimensional codebook. Regions have been generated with the process described previously.

As required by the benchmark, the experiments have been performed by randomly sampling 100 images from each of the 8 scene categories to form the training set, while using the remaining 1888 as a test-set. Figure 80 shows the precision/recall results for HRPLSA-ANN, the flat pLSA-FEATURE model as well as annotation propagation. Both latent methods are tested on different sizes of the latent spaces: i.e. $hrplsa_{25,50}$ denotes an HRPLSA-ANN with 25 region aspects and 50 word aspects. The curves in Figure 80 show that HRPLS-ANN achieves better results with respect to the other models: in particular, it obtains higher precison/recall using less region topics than pLSA-FEATURES (confront the $hrplsa_{25,50}$ result with $plsa_{60}$). Overall, latent topic models confirm their superiority over vector space models. With respect to the results obtained for the Washington data, the gap between the performance of annotation propagation and latent topic approaches has grown notably. This might be due to the

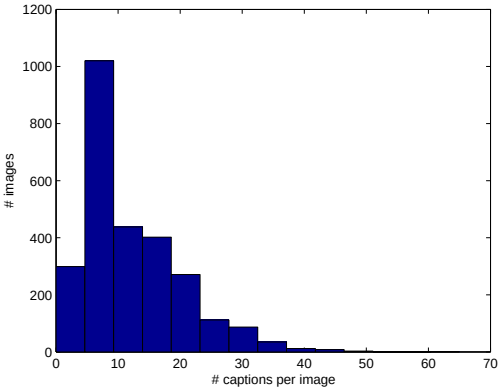


Figure 79: Distribution of caption length for the LabelMe dataset.

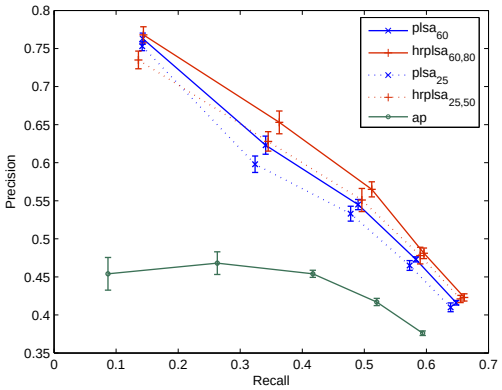


Figure 80: Precision/recall curves for the LabelMe dataset: the plot shows the performance of annotation propagation (ap), pLSA-FEATURES with 25 and 60 topics (e.g. $plsa_{25}$) as well as HRPLSA-ANN with 25 and 60 region aspects, with corresponding 50 (e.g. $hrplsa_{25,50}$) and 80 word topics, respectively.

fact the this experimentation is characterized by a low ratio between the number of training images and the size of the testing set: this aspect can have particularly affected the annotation propagation model, since it relies solely on image comparisons in order to determine the annotation of an un-captioned image. An example of images annotated by the three methods is shown in Figure 81. As with the Washington dataset, both latent topic models recall with good precision those annotations that are associated to predominant latent aspects of the image collection, such as *sky*, *building* and *mountain*. Notice how, in the second and sixth images, HRPLSA-ANN infers labels that are not present in the original ground-truth data but are, nevertheless, coherent with the visual content. In comparison to pLSA-FEATURE, the HRPLSA-ANN model seem to maintain a fair precision on low-recall terms, such as the *balcony* and *sign* captions in the third and fourth images of Figure 81: the bar plot in Figure 82 quantitatively shows the difference between the models in terms of the *per-word* precision and recall.

The challenge proposed by the authors of the LabelMe benchmark requires to recognize and segment as many objects as possible from the image collection. To some extent, we can interpret HRPLSA-ANN as an object recognition model that learns to segment visual entities based on weak labels available only at the level of the whole image. Hence, by exploiting the region labeling probability $P(a|r,d)$ as in the Washington data, we can approach the proposed object recognition challenge using the HRPLSA-ANN model.

Figure 83 plots the distribution of the vocabulary terms with recall> 0 generated by HRPLSA-ANN region labeling on the LabelMe collection. Essentially, HRPLSA-ANN seems to associate the region topics only to those captions that are the most frequent terms for the eight scene categories. This behavior can be better understood by observing the distribution of term occurrences for each scene class: Figure 84 shows that there are few words (often a single one) occurring for all the images in a scene class. These captions are likely to be co-occurring with the region-aspects that characterize the scene categories: as a consequence, the $P(a|z)$ distribution tends to associate such frequent terms to the region topics charac-

			
Original	lighthouse, tree, rocks, sea, sky, water	brush, mountain	car, hill central reservation, highway, road sign, sky, tree
HRPLSA	mountain, sea, sky, tree, water	brush, field stone, tree, trunk	car, road sign, sky, tree
plSA	car, mountain sky, tree, water	car, person sky, tree, trunk	car, mountain road, sky, tree
AP	bridge, building mountain, river, sky	brush, field stone, tree trunk	building, road sign, sky streetlight
			
Original	balcony, brand, car building, door, name person, road, walking, window, shop, sidewalk	branch, sky mountain, rocks, snow, tree	building, sky, tree
HRPLSA	balcony, car, building, tree window	mountain, sky, rocks, snow, tree	building, tree, window, sky, skyscraper
plSA	building, window, tree, trunk, car	car, mountain, person, sky, tree	building, sidewalk, car, window, skyscraper
AP	awning, balcony, brand, building, car	cloud, mountain, rocks, sky, snow	building, car, hedge, sign, sky

Figure 81: Annotation examples from the LabelMe datatset.

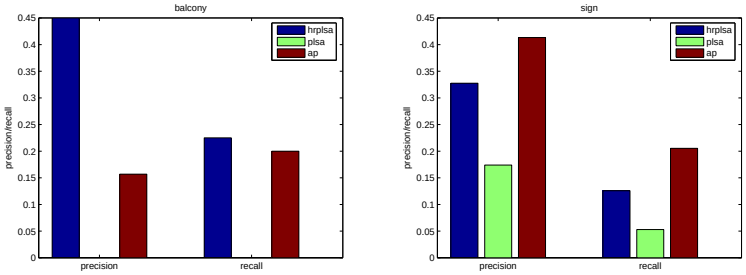


Figure 82: Precision and recall for the *balcony* and *sign* terms in the LabelMe dataset.

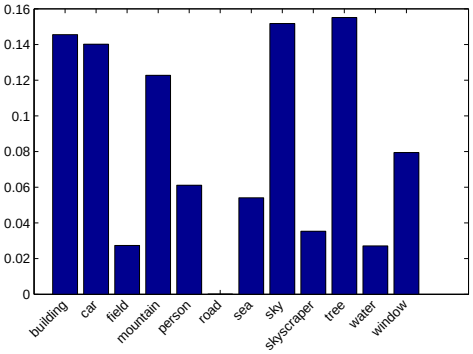


Figure 83: Distribution of word frequency for the LabelMe region labels generated by HRPLSA-ANN.

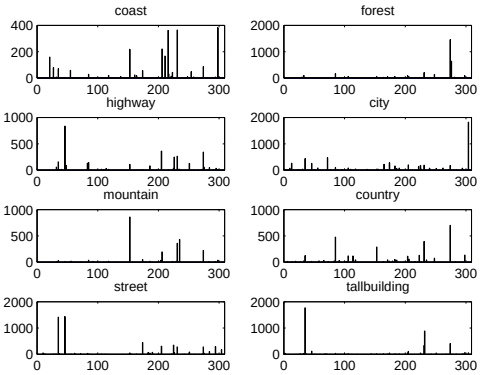


Figure 84: Distribution of word occurrences within the 8 scene classes of the LabelMe dataset. The frequently occurring word for each category are: *water* (231), *sky* (231) and *sea* (216) for coast; *tree* (274) and *trunk* (276) for forest; *car* (46) for highway; *window* (304) for city; *mountain* (153) and *snow* (235) for the mountain category; *tree* (274) and *field* (85) for country; *building* (35) and *car* (46) for street; *building* (35) and *skyscraper* (232) for tallbuildings. The number within brackets denotes the term index in the vocabulary.

terizing the scene class. Since the region labeling process recalls a single caption for each image segment (i.e. the one with highest $P(a|r,d)$ probability), it is very likely that the caption will be chosen among the most frequent terms for the image category.

In other words, the unbalanced distribution of term occurrences introduces a bias in the generation of the segment caption that yields to distinct region aspects being associated to the same frequently occurring word. Consider, for instance, the topic distribution $P(a|z)$ related to the *mountain* keyword, that is a frequently occurring term for the mountain scene category: Figure 85 shows that the *mountain* term is primarily generated by four aspects, that are number 11, 20, 40 and 55. Topic 55 characterizes noisy regions comprising, typically, sky and mountains fading in the background, while the former three topics isolate three different visual themes, that are a general mountain topic, i.e. topic 20, a snow aspect, i.e. topic 11, and a rock aspect, corresponding to topic 40 (the right side

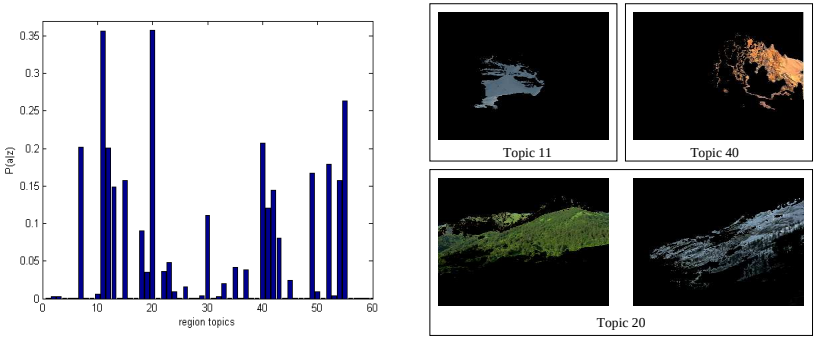


Figure 85: Distribution of topics for the *mountain* caption.

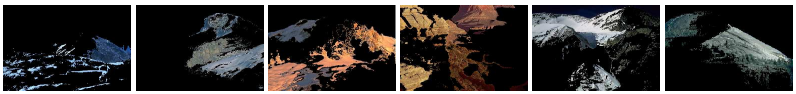
of Figure 85 shows a example of characteristic segments for the three aspects). Since the mountain keyword occurs for every image labeled with the snow caption, both keywords will have similar probabilities $P(a|z)$ of being generated by the 11-th aspect. In other words, the annotation process cannot determine which caption should be given more credit for the topic because the two annotations co-occur always with the snow aspect. On the other hand, the mountain caption is likely to be generated also by other region aspects (for instance topic 40), hence its overall region probability $P(mountain|r, z)$ will, typically, be higher than $P(snow|r, z)$, resulting in the *mountain* keyword being generated before the *snow* annotation when performing region labeling. In a sense, this issue is caused by the "ANN" part go the HRPLSA-ANN model: Figure 85, for instance, shows that the visual HRPLSA model is capable of differentiating between snow and mountain region aspects; however, at the topic annotation stage, i.e. the ANN part of the model, both aspects end-up being associated to the same *mountain* caption, thus producing the issue discussed so far. Notice that this problem appears in region labeling but it doesn't show when annotating the whole image: this is due to the fact that the former task generates a single label based on the topics distribution within the region; the annotation of the whole image, on the other hand, typically recalls multiple words, allowing more specialized captions to be generated together with frequent keywords.

Figure 86 shows an example of segments retrieved by HRPLSA-ANN for the *building, car, mountain, sea, sky, skyscraper, tree* and *window* annotations. Given the results in Figure 86, the HRPLSA-ANN model confirms the good performance obtained on the Washington data when isolating *sky* and *tree* segments, although the use of sole gray-level information still limits HRPLSA's ability in discriminating between perceptually similar visual aspect such as *sky* and *sea*. Notably, the *skyscaper* annotation is linked to a well defined distribution of region aspects, resulting in skyscraper occurrences being well segmented in the image collection (see the examples in Figure 86.f). With respect to the Washington data, there are a number of small sized visual entities, such as cars, person and windows, that have sufficient training support for being isolated in regions labeled with the corresponding keywords. However, given the small scale of these objects with respect to the image size, they tend to be isolated into coarse regions comprising also portions of their background: this results in the corresponding textual labels being associated both to the visual aspects describing the objects as well as to the topics identifying the background theme (e.g. road and buildings). As it can be seen in Figure 87, this issue is particularly evident for the *car* keyword, that is prone to be associated both to car and road segments, and for the *person* keyword, that is typically associated to cluttered segments, often without a clear visual interpretation.

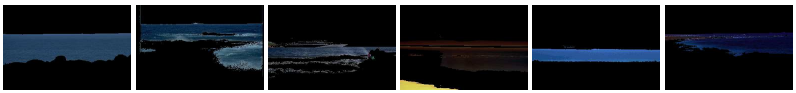
Overall, the number of segment classes discovered in the LabelMe dataset is lower than that obtained for the Washington collection: this is not unexpected, given the small training support with respect to the test data in an image collection characterized by a notable number of visual aspects (which often need to be inferred from very few occurrences). However, with this experimentation we wanted to evaluate HRPLSA-ANN performance on a challenging task to outline those limitations that will need to be addressed in future developments. The experimental results obtained on the two datasets suggests that we need to provide HRPLSA with an enhanced description of the visual content comprising color information. Possible extensions in this sense might take inspiration from the work in (BZM08), where SIFT descriptors are extended to convey color infor-

(a) Building

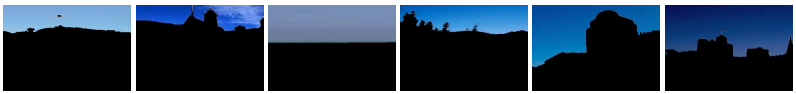
(b) Car



(c) Mountain

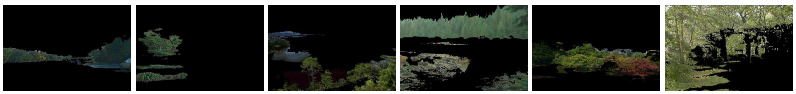


(d) Sea



(e) Sky

(f) Skyscraper



(g) Tree

(h) Window

Figure 86: An example of annotated LabelMe segments.

Figure 87: An example of LabelMe regions comprising mixed visual content: the car-labelled regions include consistent portions of the road and surrounding, while the person-labeled segments contain significant background clutter. The latter class of segments, in particular, has only a weak association to the person visual concept.

mation into a flat pLSA model; on the other hand, in (CFF07), the BOW region representation is complemented with an histogram describing the distribution of color intensities within the image regions, which is in turn added as an additional observed variable in the latent topic model. An additional issue highlighted by the experimentations relates to the spatial resolution at which captions are generated; in particular, small-sized visual entities, e.g. human figures, tend to be included into large image segments, often comprising a large portion of background. As a consequence, these segments tend to produce noisy region aspects, that are in turn associated with the vocabulary term identifying the small-sized visual entity, e.g. *people*, resulting in this caption being incorrectly attached to cluttered image regions. To address this issue, we can consider taking further the hierarchical labeling approach, by allowing labels to be generated at different levels of the topic hierarchy, e.g. associating captions related to small-sized visual entities to the visterm aspects. Finally, the experimental results on the LabelMe dataset point out the strong relationship between the quality of image annotations and the resulting region labeling performance. In particular, unbalanced vocabularies produce catastrophic effect on region annotation, biasing the caption generation process to predict only frequently occurring terms. Since this issue influences predominantly the outcome of region annotation, we can consider introducing a limited amount of supervised training information in the

HRPLSA-ANN model. In particular, we can extend the generative model with an observed variable providing the actual region labels: by these means we can bias the model so that it is more likely to predict the correct topic-caption association, instead of being biased towards the most frequent terms in the vocabulary. Notice that this effect can be obtained by introducing only a few ground truth segmentations into the training collection, whereas the rest of the training can be performed, as in the original model, on weak labels available only at the level of the whole image.

7.7 Conclusion

We have presented a novel hierarchical latent aspect model that exploits a multi-layered representation of the pictorial information to discover latent visual aspects at different levels of spatial resolution. The proposed model has been complemented with a topic annotation procedure, derived from asymmetric pLSA learning (MGP07), that allows to process and represent multimodal collections comprising images and related textual captions.

Throughout experimentation on pictorial data alone, we have shown the effectiveness of the HRPLSA model in determining a semantic segmentation of the image content based on latent visual aspects that are discovered in a completely unsupervised fashion. The experimental results show that HRPLSA reaches a superior segmentation performance with respect to the state-of-the-art models based on latent dirichlet allocation.

The multimodal extension of the HRPLSA model has been tested on two collections of annotated images, showing that the hierarchical region-based approach yields to a performance improvement with respect to flat pLSA annotation when dealing with the automatic generation of captions at the level of the whole image. Further, the experimental results have shown that the proposed model reaches a precision/recall performance that is comparable to that obtained by the state-of-the-art saliency-based CMRM approach (THL06). Besides automatic image annotation,

the HRPLSA-ANN model is capable of segmenting object occurrences by exploiting its generative model to infer associations between captions and the topics distribution within the regions. In particular, the model learns to attach textual labels to perceptually coherent portions of the image, using captions that are available only at the level of the whole picture. In this sense, image annotations serve as a source of weakly-supervised information within an, otherwise, unsupervised model. The experimental results point out how the quality of the discovered object labeling depends strongly on the quality of the annotating vocabulary: for instance, the presence of frequently occurring keywords can negatively influence the annotation process by preventing associations between regions and medium/low frequency terms.

Overall, HRPLSA-ANN is the first latent space model that addresses image annotation by exploiting a multi-resolution representation of the image content that mixes a saliency-based description of the visual information with a region-based organization of the perceptual knowledge. This multi-level representation is developed in parallel to a hierarchical structuring of the semantic space, where latent topics are discovered at spatial resolutions corresponding to visterms and image segments. The proposed topic annotation process can be seamlessly applied to any level of this topic hierarchy, potentially yielding to a multi-resolution annotation of the visual content.

Table 13: Segmentation Overlap on the MSRC-B1 Dataset.

Algorithm	Airplanes	Bicycles	Buildings	Cars	Cows	Faces	Grass	Tree	Sky	Average
dLDA	0.43	0.50	0.16	0.45	0.52	0.44	0.60	0.71	0.74	0.51
LDA10	0.11	0.06	0.09	0.14	0.11	0.40	0.41	0.69	0.41	0.27
LDA15	0.08	0.56	0.06	0.14	0.14	0.43	0.57	0.62	0.41	0.33
LDA20	0.10	0.50	0.40	0.15	0.58	0.45	0.54	0.46	0.50	0.40
LDA25	0.14	0.52	0.21	0.17	0.48	0.44	0.45	0.59	0.50	0.39
$HPRLSA_{15}^{10}$	0.32	0.37	0.59	**0.68**	**0.74**	0.44	0.87	**0.81**	0.90	0.62
$HPRLSA_{20}^{10}$	0.35	**0.56**	0.65	0.67	**0.74**	0.47	0.89	0.47	0.91	0.63
$HPRLSA_{25}^{10}$	0.35	0.52	**0.75**	0.63	0.73	0.50	0.89	0.71	0.86	**0.66**
$HPRLSA_{20}^{15}$	0.36	0.49	0.56	0.60	0.68	**0.52**	0.89	0.77	0.91	0.64
$HPRLSA_{25}^{15}$	0.42	0.52	0.61	**0.68**	0.69	0.45	**0.90**	0.61	**0.92**	0.65
$HPRLSA_{25}^{20}$	**0.44**	0.43	0.48	0.52	0.67	0.48	0.88	0.75	0.91	0.61
$\overline{HPRLSA}_{25}^{10}$	0.16	0.32	0.15	0.40	0.43	0.22	0.28	0.33	0.38	0.30
$\overline{HPRLSA}_{25}^{15}$	0.24	0.31	0.13	0.38	0.37	0.19	0.25	0.21	0.45	0.28

Table 14: Comparative precision/recall results on the Washington dataset: the KCCA results have been obtained by predicting the actual number of keywords in each image caption (HSSST06).

Model	Pred. Words	Precision	Recall
HRPLSA-ANN	3	0.612	0.357
	5	0.50	0.47
245 terms	7	0.418	0.547
	9	0.363	0.602
HRPLSA-ANN	3	0.613	0.362
	5	0.50	0.481
188 terms	7	0.418	0.552
	9	0.363	0.609
HRPLSA-ANN	3	0.615	0.368
	5	0.50	0.489
158 terms	7	0.418	0.561
	9	0.364	0.617
Saliency-based CMRM	3	0.584	0.371
	5	0.489	0.500
170 terms	7	0.412	0.576
	9	0.351	0.628
Region-based CMRM	3	0.541	0.336
	5	0.448	0.458
170 terms	7	0.383	0.541
	9	0.333	0.604
LSI $K = 40$	~ 4.8	0.490	0.480
	~ 7.42	0.414	0.588
170 terms	~ 9.70	0.356	0.648
KCCA (132 terms)	-	0.381	0.381

Chapter 8

Conclusion

The dissertation discussed a hybrid neural-probabilistic model for the unsupervised discovery of hierarchical latent structures within image collections. The proposed approach is founded, on the one hand, on a novel computational learning algorithm inspired by a perceptual learning mechanism of the visual cortex and, on the other hand, on a probabilistic model extending pLSA (Hof01) to hierarchies of localized latent topics.

The main contributions of this research can be summarized into two broad areas, i.e. competitive neural networks and image understanding, corresponding to the two parts in which the thesis has been organized. The first part of the dissertation introduced our proposal for a novel competitive learning model, named CoRe learning, inspired by the repetition suppression mechanism of the visual cortex. Experimental results show that CoRe learning provides an effective mean for estimating the unknown cluster number from the data, achieving performance results that are comparable with that of state-of-the-art kernel based models, while retaining a computational complexity that is linear with respect to the size of the dataset. The CoRe model introduces a generalized formulation for rival penalized learning that allows to train articulated neural network models, e.g. comprising neurons with several different activation functions, with a single unsupervised learning algorithm. On the practical side, we have shown that the repetition suppression mechanism over-

comes several limitations of rival penalized learning algorithms: first, it provides an adaptive mechanism that modulates the effect of neural inhibition depending on the temporal distribution of the neural activations, resulting in accurate estimates of the cluster number, irrespectively of the initial network size and without needing to customize the learning rates to the peculiarities of the dataset. Additionally, the repetition suppression mechanism enforces a parsimonious neuron allocation policy, such that densely populated regions of the input space don't get over-represented in the neural coding. As a consequence, CoRe can detect small-sized, low density clusters even in asymmetrical scenarios characterized by clusters containing a large portion of the samples of the data set. This information compression mechanism reflects the typical repetition suppression behavior, where more frequent patterns tend to be coded by fewer units, sharply tuned to their preferred stimuli.

As part of the dissertation, we have underlined the limitations of the CoRe learning formulation with respect to high-dimensional data and redundant or noisy input components. To address such limits we have proposed a *feature-wise* extension of the CoRe model, that allows to propagate the repetition suppression competition to the single input covariates. As a by-product of this extension, we derived a novel relevance measure that is capable of ranking the input features with respect to their ability in determining cluster membership. Throughout experimentations on gene expression data from DNA microarray, we have shown how the feature-wise CoRe model is able to estimate the latent structure of an high dimensional dataset, while developing a feature ranking that is consistent with the state-of-the-art knowledge regarding the functional role of genes' expression in disease realization. The experimental results on labeled data suggest that feature-wise CoRe obtains performances that are comparable with that of computationally intensive wrapper approaches, while retaining the low computational complexity of an incremental filter model. In addition to DNA microarray data, we have applied the extended CoRe model to a small dimensional dataset from breast cancer research, with the intent of estimating its performance as an exploratory tool for biomedical data analysis, in comparison with advanced Bayesian cluster-

ing algorithms. The experimental results showed that the performance of the model-based algorithms can be seriously affected by the nature of the data. On the other hand, the bio-inspired CoRe approach has proved to be a valuable tool for discovering new knowledge concerning tumor profiles, providing a reliable measure of feature relevance that can be used to infer significant markers in the discovered bio-profiles.

Besides experimental performance, we have also analyzed two important theoretical aspects of the CoRe learning model, namely robustness to noise and convergence behavior. By resorting to the theory of robust statistics we have shown that, given certain choices for the units' activation function, CoRe cluster estimation is a robust process. In particular, we have pointed out the interpretation of CoRe as a robust version of RPCL, where the rival penalization mechanism is strengthened by exploiting robust M-estimators to produce a learning process that is resistent to noise and outliers in the data. By experimental validation, we have showed the fundamental role played by robust estimation when addressing cluster number identification from noisy data, that is a key aspect when dealing, for instance, with unsupervised image segmentation. As regards CoRe's convergence behavior, we have presented a sound analysis showing how the minimization of the CoRe error satisfies the properties of separation nature, correct division and location defined in (MW06). Moreover, we have given a sound optimality criterion suggesting that the repetition suppression penalization produces a local minima escape term that drives the CoRe prototypes to the global minimum of a kernel vector quantization loss. In other words, the repulsion produced by repetition suppression does not introduce a distortion in the cluster identification process, whereas it offers a means for escaping local optima, providing theoretical motivations for CoRe's fast convergence rate. As a by-product of CoRe's convergence analysis, we have also derived a family of kernel-based learning vector quantizers that comply with the aforementioned global minimum condition. Experimental results have shown the effectiveness of the proposed kernel-based algorithms in minimizing the distortion error in unsupervised image quantization tasks.

The second part of the dissertation introduced a novel multi-layered

latent aspect model that extends probabilistic Latent Semantic Analysis to the unsupervised discovery of hierarchies of latent visual topics from image collections. The proposed model, named Hierarchical Region-topic Latent Semantic Analysis (HRPLSA), integrates with a multi-resolution image representation that exploits CoRe learning to pool together local visual patches extracted from the whole image, organizing them into perceptually meaningful intermediate structures, called regions. The resulting representation provides a stratified description of the visual content both on the spatial level, i.e. through the region-visterm organization, as well as on the semantic level, i.e. through the topic hierarchy. Experimental results on a benchmark dataset have shown the effectiveness of the model in determining a semantic segmentation of the image content, based on latent visual aspects that are discovered in a completely unsupervised fashion. In particular, HRPLSA achieves a superior segmentation performance with respect to the state-of-the-art models based on Latent Dirichlet Allocation (SRZ$^+$08). Further, by applying a multi-modal learning scheme developed for the flat pLSA model (MGP07), we have extended the HRPLSA approach to model collections of images and associated textual information. The multi-modal HRPLSA has been tested on two collections of annotated images, showing that the proposed hierarchical region-based approach yields to a performance improvement with respect to flat pLSA learning when addressing automatic image annotation. Moreover, the experimental results pointed out that HRPLSA-ANN achieves a precision/recall performance that is comparable to that obtained by the state-of-the-art saliency-based CMRM model (THL06). In addition to image-level annotation, we have shown that HRPLSA-ANN can exploit its hierarchical region-based generative model to learn to attach textual labels to the image regions using captions that are available only at the level of the whole scene.

To the extent of our knowledge, HRPLSA-ANN is the first latent space model that addresses multi-resolution annotation by exploiting a layered representation of the image content that organizes the low-level saliency-based description of the visual information into perceptually homogenous image regions. Elaborating further on the simile with text retrieval,

we can interpret the proposed approach as a means to organize documents (i.e. the images) into an orderless collection of paragraphs (the regions) that are themselves bags of words (the visterms). Such a multilevel document representation is developed in parallel to a hierarchical structuring of the semantic space, where latent topics are discovered at both levels of document representation, i.e. paragraphs and words. Within this scheme, we infer the topic of the whole document based on the mixture of its paragraphs topics which are, in turn, described by the latent semantics of the composing words.

The most general message conveyed by this dissertation is that the interchange between bio-inspired neural processing algorithms and statistical machine learning models can lead to fruitful integrations, with positive outcomes on both sides. Indeed, the first part of the dissertation suggests that biological inspiration can effectively lead to robust computational models characterized both by competitive performances as well as by strong theoretical properties. On the other hand, the experience gained while developing this doctoral research has led us to the conclusion that purely cognitive approaches might not yield to the best solutions when applied to complex scenarios such as in automatic image annotation. With this respect, statistical learning algorithms have strong modeling capabilities that earned them a central role in the image understanding community. However, the complexity of their parameter fitting process often prevents their application to large scale problems while, as shown by the breast cancer case study, they might incur in catastrophic degenerations whenever they have to cope with peculiar data distributions. The experimental results provided in the dissertation suggest that neural processing approaches have the flexibility to deal with such data irregularities being, at the same time, characterized by limited computational requirements. Hence, a new research issue emerges from the strive to reach new forms of integrations between the flexibility and computational feasibility of neural processing, and the well-foundedness and modeling power of statistical machine learning approaches. In the context of this thesis, we have realized an integration where the neural algorithm intervenes to provide bottom-up information that *teaches* the hierar-

chical topic model how low level aspects are expected to co-occur within a spatial neighborhood. The model in (SRZ+08), for instance, achieves this by resorting to a-priori knowledge that is used to constrain the way their hierarchical model assigns visual words to a certain node in the topic hierarchy.

Indeed, there are several ways to extend the work presented in this dissertation. As regards the CoRe learning model, we are planning to reformulate it to account more closely for the sequential nature of the biological repetition suppression mechanism. We recall that, at the cortical level, repetition suppression rises as the consequence of the repeated presentation of similar stimuli within a sequence of perceptual inputs. In our intentions, the reformulation in terms on sequential processing should allow to extend further the model by including the effect of two additional correlates of perceptual learning, that are *enhancement* and *delayed activation* (see their description in Chapter 2). This model should result in a novel neural network paradigm for sequence learning, where repetition suppression is used to generate a compact coding of the single items in the sequence, while the enhancement and delayed activation mechanisms provide the means for learning contextual information from the input sequences. Possible applications of this model would range from validating attentional models of perceptual learning, to the processing of sequential information, comprising strings, trajectories and time series data. Another interesting area where CoRe learning can find effective application is within the so-called *third generation* or *spiking* neural networks (MB99). These models have been proposed as a refined approach to neurocomputing where information, rather than being represented under the form of weight vectors, is coded in the firing intervals of the neurons. The development of an efficient unsupervised learning mechanism is still an open challenge within this neural model: it is therefore our intention to explore possible applications of the repetition suppression mechanism in this context. Additionally, on the practical side, we plan to exploit the peculiarities of the feature-wise CoRe algorithm to develop a tool for the unsupervised exploration of bio-medical datasets: this tool should provide the user with a range of hypotheses concerning the most likely number of

functional classes within the data, together with an estimate of relevant markers/predictors for the discovered classes.

As regards the hierarchical image annotation model, there is still work to be done for validating the performance of HRPLSA-ANN in comparison with relevant region-based image annotation systems in the literature. The first step to provide a thorough validation of the model requires to build a real-life sized benchmark: this ought to be an open-access, standardized collection of annotated images depicting articulated scenes with multiple objects portrayed in different sceneries and providing high quality ground truth segmentations of the visual content. In our opinion, the LabelMe collection provides an interesting starting point for the establishment of such a benchmark; however, as noted in the experimental evaluation, there is a consistent amount of work to be done for selecting suitable images as well as for preparing an high-quality annotation vocabulary. From the point of view of possible developments of the model, we plan to study a generalized formulation of the HRPLSA scheme that allows to describe generic topic hierarchies that go beyond the two-layered structure described in the present thesis. For instance, the generalized model should comprise multiple layers of topics, each drawing information from data gathered at different spatial resolutions. This, on the one hand, requires to formulate a novel multi-resolution representation of the visual content that generalizes the region/visterms organization. On the other hand, it requires the development of a learning scheme that is capable of automatically determining the depth of the topic hierarchy (and, implicitly, of the multi-resolution representation) from the data. On a practical side, it would be interesting to exploit HRPLSA-ANN to develop a frontend for the LabelMe manual annotation tool. This tool should exploit annotated LabelMe pictures to learn to provide the user with suggestions concerning the most likely keywords for the regions of new, un-captioned images. Such a tool can potentially yield to two positive outcomes: on the one hand, it would allow the users to save annotation time by having at least some portions of the image being correctly segmented and annotated in a fully automated way. On the other hand, it would provide the system with a relevance feedback from the user that can be adaptively in-

corporated into the HRPLSA-ANN model to increase its region labeling performance.

References

[ABQ+06] F. Ambrogi, E. Biganzoli, P. Querzoli, S. Ferretti, P. Boracchi, S. Alberti, E. Marubini, and I. Nenci. Molecular subtyping of breast cancer from traditional tumor marker profiles using parallel clustering methods. *Clin Cancer Res.*, 12(1):781–790, 2006. 4, 161, 176, 177, 178, 179, 180, 181, 182

[ACFV03] G. Acciani, E. Chiarantoni, G. Fornarelli, and S. Vergura. A feature extraction unsupervised neural network for an environmental data set. *Neural Networks*, 16:427–436, 2003. 26

[Aka74] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, Dec 1974. 32

[AKCM90] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3(3):277–290, 1990. 24, 45

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950. 90

[ARS07] P. Arias, G. Randall, and G. Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007. 123

[ATRB95] A. P. Ashbrook, N. A. Thacker, P. I. Rockett, and C. I. Brown. Robust recognition of scaled shapes using pairwise geometric histograms. In *BMVC '95: Proceedings of the 6th British conference on Machine vision (Vol. 2)*, pages 503–512, Surrey, UK, UK, 1995. BMVA Press. 197

[BA02] A. Baraldi and E. Alpaydin. Constructive feedforward ART clustering networks. *IEEE Transactions on Neural Networks*, 13(3):645–677, May 2002. 63

[Bar01] D. Barnard, K.; Forsyth. Learning the semantics of words and pictures. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2:408–415, 2001. 201, 226

[Bau00] A. Baumberg. Reliable feature matching across widely separated views. *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 1:774–781, 2000. 196

[BDdF+03] Kobus Barnard, Pinar Duygulu, Nando de Freitas, David Forsyth, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, Feb. 2003. 231, 269

[BDG+03] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2:675–682, June 2003. 201

[BETG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008. 198, 199

[Bez81] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. 34, 64, 83

[BHHSV02] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2002. 62, 63, 65

[BJ03] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR'03)*, Aug. 2003. 219, 229, 230, 231

[BLO02] S. Brandt, J. Laaksonen, and E. Oja. Statistical shape features for content-based image retrieval. *Journal of Mathematical Imaging and Vision*, 17(2):187–198, Sep. 2002. 193

[BMP98] M. Burl, M.Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proceedings of the European Conference on Computer Vision (ECCV'98)*, pages 628–641, 1998. 220

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522, Apr 2002. 197

[BNJ03] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. 4, 218, 219, 229

[BPPM94] P. F. Brown, S. Della Pietra, V. J. Della Pietra, and R. L. Mercer. The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1994. 227

[BRS+01] A. Bhattacharjee, W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno R, M. Gillette, M. Loda, G. Weber, E.J. Mark, E.S. Lander, W. Wong, B.E. Johnson, T.R. Golub, D.J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Science USA*, 98(24):13790–13795, November 2001. 170

[BS06] D. Bacciu and A. Starita. Competitive repetition suppression learning. In *Proc. of the 2006 International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science. Springer, 2006. 41

[BS07a] D. Bacciu and A. Starita. Convergence behavior of competitive-repetition suppression clustering. In *Proc. of the 2007 International Conference on Neural Information Processing*. Springer, November 2007. 3

[BS07b] D. Bacciu and A. Starita. A robust bio-inspired clustering algorithm for the automatic determination of unknown cluster number. In *Proc. of the 2007 International Joint Conference on Neural Networks*. IEEE, August 2007. 67

[BS08] D. Bacciu and A. Starita. Competitive repetition suppression (CoRe) clustering: a biologically inspired learning model with application to robust clustering. *To Appear in IEEE Transactions on Neural Networks*, 2008. 3

[BT98] C.M. Bishop and M.E. Tipping. A hierarchical latent variable model for data visualization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):281–293, Mar 1998. 244

[Bun94] W. L. Buntine. Operations for learning with graphical models. *J. Artif. Intell. Res. (JAIR)*, 2:159–225, 1994. 215

[BWS03] G. H. Bakir, J. Weston, and B. Schlkopf. Learning to find pre-images. In *Advances in neural information processing systems*, 2003. 123

[BZM08] A. Bosch, A. Zisserman, and X. Muoz. Scene classification using a hybrid generative/discriminative approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):712–727, April 2008. 208, 213, 218, 239, 276, 284

[CB01] A. Corduneanu and C.M. Bishop. Variational bayesian model selection for mixture distributions. In T. Richardson and T. Jaakkola, editors, *Artificial Intelligence and Statistics*, pages 27–34. Morgan Kaufmann, Jul. 2001. 176

[CBGM02] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1026–1038, 2002. 201, 227

[CCS04] C. Cusano, G. Ciocca, and R. Schettini. Image annotation using svm. *Proceedings of SPIE*, 5304:330–338, 2004. 225

[CDF+04] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 209, 211

[CdFB04] P. Carbonetto, N. de Freitas, and K. Barnard. A statistical model for general contextual object recognition. In *Computer Vision - ECCV 2004, 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I*, volume 3021 of *Lecture Notes in Computer Science*, pages 350–362. Springer, 2004. 201

[CFF07] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007. 219, 241, 265, 286

[CG88] G.A. Carpenter and S. Grossberg. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88, Mar 1988. 30, 49, 152

[Cha05] D. Charalampidis. A modified k-means algorithm for circular invariant clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1856–1865, December 2005. 36

[Che05] YM. Cheung. On rival penalization controlled competitive learning for clustering with automatic cluster number selection. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1583–1588, 2005. 12, 48, 60

[cIBFS04] Ş. İlker Birbil, Shu-Cherng Fang, and Ruey-Lin Sheu. On the convergence of a population-based global optimization algorithm. *J. of Global Optimization*, 30(2-3):301–318, 2004. 114

[CL07] C. Constantinopoulos and A. Likas. Unsupervised learning of gaussian mixtures based on variational component splitting. *IEEE Trans. Neural Netw.*, 18(3):745–755, May 2007. 36, 176

[CM99] D. Comaniciu and P. Meer. Mean shift analysis and applications. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:1197–1203, 1999. 201

[Coh60] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960. 178

[CSS98] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*, page 24, Washington, DC, USA, 1998. IEEE Computer Society. 226

[CV05] F Camastra and A. Verri. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):801–805, May 2005. 64, 65

[DBdFF02] P. Duygulu, Kobus Barnard, J. F. G. de Freitas, and David A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, London, UK, 2002. Springer-Verlag. 227

[DD95] R. Desimone and J. Duncan. Neural mechanism of selective visual attention. *Ann. Rev. Neurosci.*, (18):193–222, 1995. 2, 16

[DeS88] D. DeSieno. Adding conscience to competitive learning. In *IEEE Annual International Conference on Neural Networks*, pages 1117–1124. IEEE Computer Society, July 1988. 24

[Des96] R. Desimone. Neural mechanisms for visual memory and their role in attention. In *Proceedings of National Academy of Sciences USA*, volume 93, pages 13494–13499, November 1996. 3, 11, 16, 17, 18, 19, 20

[DGK04] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *KDD '04: Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, New York, NY, USA, 2004. ACM Press. 93

[Din02] C. H. Q. Ding. Analysis of gene expression profiles: class discovery and leaf ordering. In *RECOMB '02: Proceedings of the sixth annual international conference on Computational biology*, pages 127–136, New York, NY, USA, 2002. ACM Press. 144, 167, 169

[Din03] C. H. Q. Ding. Unsupervised feature selection via two-way ordering in gene expression analysis. *Bioinformatics*, 19(10):1259–1266, 2003. 169

[DKN05] T. Deselaers, D. Keysers, and H. Ney. Discriminative training for object recognition using image patches. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:157–162, June 2005. 211

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 31, 217

[dM85] C. Von der Malburgs. Nervous structures with dynamical links. *Ber. Bunsenges. Phys. Chem.*, (89):703–710, 1985. 16

[DMBD04] M. Dugas, S. Merk, S. Breit, and P. Dirschedl. mdclust—exploratory microarray analysis by multidimensional clustering. *Bioinformatics*, 20(6):931–936, 2004. 183

[Dor06] G. Dorkó. *Selection of Discriminative Regions and Local Descriptors for Generic Object Class Recognition*. PhD thesis, Institut National Polytechnique de Grenoble, june 2006. 206

[DSH00] Y. Dufournaud, C. Schmid, and R. Horaud. Matching images with different resolutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 612–618, Hilton Head Island, South Carolina, USA, Jun 2000. 204

[dSQ06] P. M. da Silva Quelhas. *Scene Image Classification and Segmentation with Quantized Local Descriptors and Latent Aspect Modeling*. PhD thesis, Ecole Polytechnique Federale de Lausanne, Dec. 2006. 212

[EF98] T. Eltoft and R.J.P. De Figueiredo. New neural network for cluster-detection-and-labeling. *IEEE Transactions on Neural Networks*, 9(5):1021–1035, Sept 1998. 63

[ESI98] Y. El-Sonbaty and M. A. Ismail. On-line hierarchical clustering. *Pattern Recognition Letters*, 19(14):1285–1291, Dec 1998. 65

[FA91] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991. 196

[Fel98] C. Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998. 253, 255

[FFFP04] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop*, page 178, 2004. 212, 219, 237

[FFFP07] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.*, 106(1):59–70, 2007. 211

[FFFPZ05] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2:1816–1823, Oct. 2005. 4, 211, 218, 255

[FFP05] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society. 211, 219

[FGLX04] J. Fan, Y. Gao, H. Luo, and G. Xu. Automatic image annotation by using concept-sensitive salient objects for image content representation. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR'04)*, pages 361–368, July 2004. 233

[Fis36] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7(2):179 – 188, 1936. 61

[FJ02] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):381–396, 2002. 36, 37, 176

[FK97] H. Frigui and R. Krishnapuram. Clustering by competitive agglomeration. *Pattern Recognition*, 30(7):1109–1119, July 1997. 34, 35

[FK99] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450 – 465, May 1999. 35, 47, 64, 71, 74, 78, 79, 80

[FML04] S.L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:1002–1009, 2004. 201, 228, 233

[FMR05] M. Filippone, F. Masulli, and S. Rovetta. Unsupervised gene selection and clustering using simulated annealing. In *LNCS*, volume 3849, pages 229–235. Springer, 2005. 142, 145, 169, 183

[FN04] H. Frigui and O. Nasraoui. Unsupervised learning of prototypes and attribute weights. *Pattern Recognition*, 37(3):567–581, Mar 2004. 35

[FOPA04] M. Fussenegger, A. Opelt, A. Pinz, and P. Auer. Object recognition using segmentation for feature detection. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 3:41–44, Aug. 2004. 201

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03: Proceedings if the IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 264–271, Washington, DC, USA, 2003. IEEE Computer Society. 220

[FR98] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, August 1998. 32

[Fre92] R. M. French. Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science*, 4:365 – 377, 1992. 20

[Fri94] B. Fritzke. Fast learning with incremental RBF networks. *Neural Processing Letters*, 1(1):2–5, 1994. 28

[Fri95] B. Fritzke. A growing neural gas network learns topologies. In Gerald Tesauro, David S. Touretzky, and Todd K. Leen, editors, *Advances in Neural Information Processing Systems 7, [NIPS Conference, Denver, Colorado, USA, 1994]*, pages 625–632. MIT Press, 1995. 28

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995. 208

[FtHRKV94] L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, and M.A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4:171–187, 1994. 196

[Fuk03] K. Fukushima. Neocognitron for handwritten digit recognition. *Neurocomputing*, 51:161–181, April 2003. 49, 68

[GB00] B. Gabrys and A. Bargiela. General fuzzy min-max neural network for clustering and classification. *IEEE Transactions on Neural Networks*, 11(3):769–783, May 2000. 63

[GB01] A.B.A. Graf and S. Borer. Normalization in support vector machines. *Lecture Notes in Computer Science*, 2191:277–282, Jan 2001. 92

[GD05] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2:627–634, June 2005. 211

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. 145

[GG05] E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, New York, NY, USA, 2005. ACM. 216

[GGPC02] E. Gaussier, C. Goutte, K. Popat, and F. Chen. A hierarchical model for clustering and categorising documents. In *Proceedings of the 24th BCS-IRSG European Colloquium on IR Research*, pages 229–247, London, UK, 2002. Springer-Verlag. 249, 257

[GGR01] H. Greenspan, J. Goldberger, and L. Ridel. A continuous probabilistic framework for image matching. *Comput. Vis. Image Underst.*, 84(3):384–406, 2001. 201, 209

[Gir02] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, May 2002. 33, 62, 63, 64, 65

[GJ97] A. Gupta and R. Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, 1997. 208

[GMU96] L. J. Van Gool, T. Moons, and D. Ungureanu. Affine/ photometric invariants for planar intensity patterns. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 642–651, London, UK, 1996. Springer-Verlag. 198

[GP94] M.M. Gorkani and R.W. Picard. Texture orientation for sorting photos "at a glance". *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on*, 1:459–464 vol.1, Oct 1994. 193

[GS04] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101:5228–5235, 2004. 219

[GSHM06] K. Grill-Spector, R. Henson, and A. Martin. Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Sciences*, 10(1):14–23, Jan 2006. 20, 21

[GST+99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999. 165, 168, 169

[HA85] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 1985. 162

[Ham74] F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974. 76

[Han08] A. Hanbury. A survey of methods for image annotation. *Journal of Visual Languages and Computing (In press)*, 2008. 276

[Har79] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979. 194

[HHB05] R.J. Hathaway, J.M. Huband, and J.C. Bezdek. Kernelized non-euclidean relational fuzzy c-means algorithm. *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*, pages 414–419, 2005. 121

[HKM+97] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–768, June 1997. 193

[HKM+99] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Spatial color indexing and applications. *International Journal of Computer Vision*, 35(3):245–268, 1999. 193

[HL05] J.S. Hare and P.H. Lewis. Saliency-based models of image content and their application to auto-annotation by semantic propagation. In *Multimedia and the Semantic Web, European Semantic Web Conference 2005*, 2005. 270

[HLES06] J. Hare, P. Lewis, P. Enser, and C. Sandom. A linear-algebraic technique with an application in semantic image retrieval. In *Proceedings of the International Conference on Image and Video Retrieval*, pages 31–40, 2006. 224

[HNDA04] H.Muller, N.Michoux, D.Bandon, and A.Geissbuhler. A review of content-based image retrieval systems in medical applications: Clinical benefits and future directions. *Int J Med Inform.*, 1(73):1–23, February 2004. 2

[Hof99] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 50–57. ACM, 1999. 249

[Hof01] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001. 4, 213, 214, 217, 247, 291

[HR04] P. Howarth and S. M. Rüger. Evaluation of texture features for content-based image retrieval. In *Image and Video Retrieval: Third International Conference, CIVR 2004, Dublin, Ireland, July 21-23, 2004. Proceedings*, volume 3115 of *Lecture Notes in Computer Science*, pages 326–334. Springer, 2004. 194

[HSLN08] J. S. Hare, S. Samangooei, P. H. Lewis, and M. S. Nixon. Semantic spaces revisited: investigating the performance of auto-annotation and semantic retrieval using semantic spaces. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 359–368, New York, NY, USA, 2008. ACM. 268

[HSSST06] D. R. Hardoon, C. Saunders, S. Szedmk, and J. Shawe-Taylor. A correlation approach for automatic image annotation. In Xue Li, Osmar R. Zaane, and Zhanhuai Li, editors, *ADMA*, volume 4093 of *Lecture Notes in Computer Science*, pages 681–692. Springer, 2006. 267, 270, 290

[HSV05] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Process. Lett.*, 21(1):21–44, 2005. 122

[Hub81] P.J. Huber. *Robust Statistics*. Wiley, 1981. 3, 5, 73, 74, 76, 77

[Hun98] R. W. G. Hunt. *Measuring Color*. Fountain Press, 3 edition, 1998. 193

[HW62] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, (160):106–154, 1962. 13, 14

[JDSW06] M. Jamieson, S. Dickinson, S. Stevenson, and S. Wachsmuth. Using language to drive the perceptual grouping of local image features. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2102–2109, Washington, DC, USA, 2006. IEEE Computer Society. 212, 227

[JG05] T.W. James and I. Gauthier. Repetition-induced changes in bold response reflect accumulation of neural activity. *Human Brain Mapping*, 27(1):37–46, Jun 2005. 20

[JLM03] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th International Conference on Research and Development in Information Retrieval*, Aug. 2003. 227

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999. 31

[JT05] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 1:604–610, Oct. 2005. 209

[JTZ04] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386, Nov. 2004. 145

[KCH95] P. M. Kelly, T. M. Cannon, and D. R. Hush. Query by image example: the comparison algorithm for navigating digital image databases (CANDID) approach. In W. Niblack and R. C. Jain, editors, *Proc. SPIE Vol. 2420, p. 238-248, Storage and Retrieval for Image and Video Databases III, Wayne Niblack; Ramesh C. Jain; Eds.*, volume 2420 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 238–248, March 1995. 208

[KK07] S. Kim and I. S. Kweon. Robust model-based scene interpretation by multilayered context information. *Comput. Vis. Image Underst.*, 105(3):167–187, 2007. 222, 223

[KM58] E. L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481, 1958. 182

[Koh82] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59 – 69, Jan 1982. 26, 48, 113

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding groups in data. an introduction to cluster analysis*. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990, 1990. 176

[KS04] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:506–513, June-2 July 2004. 197

[KT04] J.T.-Y. Kwok and I.W.-H. Tsang. The pre-image problem in kernel methods. *Neural Networks, IEEE Transactions on*, 15(6):1517–1525, Nov. 2004. 123

[KvD87] KJ. J. Koenderink and A. J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, Mar. 1987. 196

[KZB04] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *In Proceedings of the 8th European Conference on Computer Vision*, pages 228–241, 2004. 206, 220

[LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, Jan 1980. 24, 112, 115

[LC06] D. Liu and T. Chen. Semantic-shift for unsupervised object detection. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 16, Washington, DC, USA, 2006. IEEE Computer Society. 256

[LCC06] D. Liu, D. Chen, and T. Chen. Latent layout analysis for discovering objects in images. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2:468–471, 0-0 2006. 218

[LCZ+06] X. Li, L. Chan, L. Zhang, F. Lin, and WY. Ma. Image annotation by large scale content-based image retrieval. In *Proceedings of the ACM Multimedia 2006 (MM'06)*, pages 607–610, 2006. 225

[LF98] M. P. Lee and A. P. Feinberg. Genomic Imprinting of a Human Apoptosis Gene Homologue, TSSC3. *Cancer Res*, 58(5):1052–1056, 1998. 171

[LG97] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure. *Image and Vision Computing*, 15(6):415–434, 1997. 200, 205

[LHB04] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2:II–97–104 Vol.2, 27 June-2 July 2004. 49

[Lin98] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998. 202, 203

[LJ08] D. Larlus and F. Jurie. Latent mixture vocabularies for object categorization and segmentation. *Journal of Image & Vision Computing*, 2008. 212, 219

[LL00] A. Luk and S. Lien. Dynamics of the generalised lotto-type competitive learning. In *Proc. of Int Joint Conf on Neur Netw 2000*, volume 4, pages 597–601, 2000. 114

[LMJ03] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Proc. Ann. Conf. Neural Information Processing Systems*, Dec. 2003. 228

[LMS06] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *British Machine Vision Conference (BMVC*, September 2006. 209

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society. 197, 270

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 202

[LS99] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999. 216

[LSDJ06] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006. 2

[LSP03] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2:319–324, June 2003. 195, 196

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society. 212

[LVV03] A. Likas, N. Vlassis, and J. J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, February 2003. 116

[LW03] J. Li and J.Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 25:1075–1088, 2003. 233

[LWG07] Y. Li, W. Wang, and W. Gao. Object recognition based on dependent pachinko allocation model. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 5:V –337–V –340, 16 2007-Oct. 19 2007. 257

[MB99] W. Maass and C. M. Bishop, editors. *Pulsed Neural Networks*. MIT Press, Cambridge, MA, USA, 1999. 296

[MBCCGF05] G. Musumarra, V. Barresi, D. F. Condorelli, and S. Scire C. G. Fortuna. Genome-based identification of diagnostic molecular markers for human lung carcinomas by PLS-DA. *Computational Biology and Chemistry*, 29(3):183–195, June 2005. 172

[MCUP02] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA. 203

[MD94] E.K Miller and R. Desimone. Parallel neuronal mechanisms for short-term memory. *Science*, (263):520–522, 1994. 20

[MF00] D. MacDonald and C. Fyfe. The kernel self-organising map. *Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000. Proceedings. Fourth International Conference on*, 1:317–320 vol.1, 2000. 121

[MGP03] F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 275–278, New York, NY, USA, 2003. ACM. 269

[MGP07] F. Monay and D. Gatica-Perez. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1802–1817, Oct. 2007. 4, 231, 232, 233, 236, 239, 250, 251, 258, 268, 269, 287, 294

[ML02] T. Minka and J. Lafferty. Expectation propagation for the generative aspect model. In *Proceedings of the Conference on Uncertainty in AI*, 2002. 219

[MLS06] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 1:26–36, June 2006. 209, 211

[MM96] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, Aug 1996. 193

[MP43] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, December 1943. 2

[MP00] Geoff J. McLachlan and David Peel. *Finite Mixture Models*. J. Wiley, New York, 2000. 32

[MPGRLRGB02] J. Munoz-Perez, J. A. Gomez-Ruiz, E. Lopez-Rubio, and M. A. Garcia-Bernal. Expansive and competitive learning for vector quantization. *Neural Process. Lett.*, 15(3):261–273, 2002. 71, 96, 101, 102, 113, 117, 118, 119

[MR98] O. Maron and A. Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *In Proc. 15th International Conf. on Machine Learning*, pages 341–349. Morgan Kaufmann, 1998. 225

[MS91] T. M. Martinetz and K. J. Schulten. A "neural gas" network learns topologies. In Teuvo Kohonen, Kai Mäkisara, Olli Simula, and Jari Kangas, editors, *Proceedings of the International Conference on Artificial Neural Networks 1991* (Espoo, Finland), pages 397–402. Amsterdam; New York: North-Holland, 1991. 28, 113

[MS01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001. 203, 204

[MS04] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. 205

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, Oct. 2005. 196, 197, 199

[MS06] M. Marszaek and C. Schmid. Spatial weighting for bag-of-features. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2118–2125, Washington, DC, USA, 2006. IEEE Computer Society. 212

[MTMG03] S. Monti, P. Tamayo, J. Mesirov, and T. R. Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.*, 52(1-2):91–118, 2003. 144, 161, 162, 164, 166, 170, 183

[MTO99] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedigns of the International Workshop on Multimedia Intelligent Storage and Retrieval Management*, Oct. 1999. 226

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005. 191, 206

[MV00] R. Mollineda and E. Vidal. A relative approach to hierarchical clustering. In IOS Press, editor, *Frontiers in Artificial Intelligence and Applications*, volume 56 of *Pattern Recognition and Applications*, pages 19–28. M. Torres and A. Sanfeliu, Amsterdam, The Netherlands, 2000. 63

[MW06] J. Ma and T. Wang. A cost-function approach to rival penalized competitive learning (RPCL). *IEEE Transactions on System Man Cybernetics B, Cybernetics*, 36(4):722 – 737, August 2006. 71, 81, 86, 95, 96, 107, 108, 110, 293

[MZ99] W.-Y. Ma and H. Zhang. *Handbook of Multimedia Computing*, chapter Content-Based Image Indexing and Retrieval. CRC Press, 1999. 193

[NJT06] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision*. Springer, 2006. 208, 211

[NJW01] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. of Advances in Neural Information Processing Systems*, 2001. 64, 65, 93

[NO03] K.A. Norman and R.C. O'Reilly. Modeling hippocampal and neocortical contributions to recognition memory: A complementary learning systems approach. *Psychological Review*, (110):611–646, 2003. 20

[NS04] S. Nakkrase and P. Sophatsathit. An RPCL-based indexing approach for software component classification. *Int J. Soft Eng Know*, 14:497–518, 2004. 26

[NZF+03] T.M. Nair, C.L. Zheng, J.L. Fink, R.O. Stuart, and M. Gribskov. Rival penalized competitive learning (RPCL): a topology-determining algorithm for analyzing gene expression data. *Computational Biology and Chemistry*, 27:565–574, 2003. 26

[OB05] B. Ommer and J. M. Buhmann. Object categorization by compositional graphical models. In Anand Rangarajan, Baba C. Vemuri, and Alan L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition, 5th International Workshop, EMMCVPR 2005, St. Augustine, FL, USA, November 9-11, 2005, Proceedings*, volume 3757 of *Lecture Notes in Computer Science*, pages 235–250. Springer, 2005. 211

[OBW96] J. J. Oliver, R. A. Baxter, and C. S. Wallace. Unsupervised Learning using MML. In *Machine Learning: Proceedings of the Thirteenth International Conference (ICML 96)*, pages 364–372. Morgan Kaufmann Publishers, 1996. 32

[OSB06] B. Ommer, M. Sauter, and J .M. Buhmann. Learning top-down grouping of compositional hierarchies for recognition. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 194, Washington, DC, USA, 2006. IEEE Computer Society. 222

[OT01] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. Journal of Computer Vision (IJCV)*, 42(3):145–175, 2001. 194

[PBEK93] N.R. Pal, J.C. Bezdek, and E.C.-K.Tsao. Generalized clustering networks and kohonen's self-organizing scheme. *IEEE Transactions on Neural Networks*, 4(4):549–557, Jul 1993. 63

[PCM07] A. Perina, M. Cristani, and V. Murino. Natural scenes categorization by hierarchical extraction of typicality patterns. In *ICIAP '07: Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 801–806, Washington, DC, USA, 2007. IEEE Computer Society. 222

[Per08] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1243–1256, July 2008. 213

[PHK+00] J. Puzicha, M. Held, J. Ketterer, J. M. Buhmann, and D. W. Fellner. On spatial quantization of color images. *IEEE Transactions on Image Processing*, 9(4):666–682, Apr 2000. 86

[PPF+03] G. Pelosi, F. Pasini, F. Fraggetta, U. Pastorino, A. Iannucci, P. Maisonneuve, G. Arrigoni, G. De Manzoni, E. Bresaola, and G. Viale. Independent value of fascin immunoreactivity for predicting lymph node metastases in typical and atypical pulmonary carcinoids. *Lung Cancer*, 42(2):203–213, November 2003. 174

[PR01] G. Patané and M. Russo. The enhanced LBG algorithm. *Neural Netw.*, 14(9):1219–1237, 2001. 113, 116, 137

[QH07] Xiaojun Qi and Yutao Han. Incorporating multiple svms for automatic image annotation. *Pattern Recogn.*, 40(2):728–741, 2007. 225

[QMO+05] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 1:883–890, Oct. 2005. 218

[QMO+07] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, and T. Tuytelaars. A thousand words in a scene. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(9):1575–1589, Sept. 2007. 213, 218

[RB01] R.J.Hathaway and J.C. Bezdek. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on System, Man and Cybernetics B, Cybernetics*, 31(5):735–744, Oct 2001. 63

[RD05] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision*, volume 2, pages 1508–1511, October 2005. 204

[RFE+06] B.C. Russell, W.T. Freeman, A.A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2:1605–1614, 2006. 201, 219, 239, 256, 258, 259

[RGF92] K. Rose, E. Gurewitz, and G.C. Fox. Vector quantization by deterministic annealing. *Information Theory, IEEE Transactions on*, 38(4):1249–1257, Jul 1992. 113

[Ris96] J.J. Rissanen. Fisher information and stochastic complexity. *Information Theory, IEEE Transactions on*, 42(1):40–47, Jan 1996. 32

[RLL+06] M. Reich, T. Liefeld, G.J. Lerner, P. Tamayo, and J.P. Mesirov. Genepattern 2.0. *Nature Genetics*, 38(5):500–501, 2006. 171

[RTMF08] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, 2008. 253, 255, 272

[Rub98] Y. Rubner. The earth-movers distance as a metric for image retrieval. Technical Report STAN-CS-TN-98-86, Stanford University, 1998. 229

[RZ85] D.E. Rumelhart and D. Zipser. Competitive learning. *Cognitive Science*, 9:75 – 112, 1985. 11

[RZ86] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. pages 151–193, 1986. 24

[RZ06] D. A. Ross and R. S. Zemel. Learning parts-based representations of data. *Journal of Machine Learning Research*, 7:2369–2397, Jun 2006. 221

[SB91] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, Nov. 1991. 193

[SB97] S. M. Smith and J. M. Brady. SUSAN: A new approach to low level image processing. *Int. J. Comput. Vision*, 23(1):45–78, 1997. 204

[SB04] S. Still and W. Bialek. How many clusters? an information-theoretic perspective. *Neural Comput.*, 16(12):2483–2506, 2004. 32

[Sch78] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. 32

[SH06a] F. Shen and O. Hasegawa. An adaptive incremental LBG for vector quantization. *Neural Netw.*, 19(5):694–704, 2006. 113, 117, 137

[SH06b] A. I. Su and G. M. Hampton. Molecular signatures of commonly fatal carcinomas, September 2006. 174

[SM00] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, Aug 2000. 201, 227

[SMB+99] B. Scholkopf, S. Mika, C. J. C. Burges, P. Knirsch, K. R. Muller, G. Ratsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE Trans. on Neur. Net.*, 10(5):1000–1017, sep 1999. 23, 104, 105, 123, 125, 126

[SO03] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003. 11

[Spa80] H. Spath. *Cluster analysis algorithms*. Ellis Horwood, 1980. 101

[Squ92] L. R. Squire. Declarative and nondeclarative memory: Multiple brain systems supporting learning and memory. *J. Cognitive Neuroscience*, 4(3):232–243, 1992. 16

[SR96] S. Sobotka and J.L. Ringo. Mnemonic responses of single units recorded from monkey inferotemporal cortex, accessed via transcommissural versus direct pathways: a dissociation between unit activity and behavior. *J. Neuroscience*, (16):4222–4230, 1996. 20

[SRE+05] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 1:370–377, Oct. 2005. 4, 208, 211, 212, 214, 217, 218, 220, 232, 247, 249

[SRZ+08] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 239, 258, 260, 294, 296

[SSM98] B. Scholkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998. 32, 92, 93, 121

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. 71, 90, 128

[SWB+07] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):411–426, 2007. 49, 68

[SWY75] G Salton, A Wong, and C Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. 224

[SZ02] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431, London, UK, 2002. Springer-Verlag. 196

[SZ03] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477 vol.2, Oct. 2003. 210

[TG04a] M. Tsodyks and C. Gilbert. Neural networks and perceptual learning. *Nature*, 7010(431):775–781, October 2004. 17

[TG04b] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004. 205

[THL06] J. Tang, J.S. Hare, and P.H. Lewis. Image auto-annotation using a statistical model with salient regions. *Multimedia and Expo, 2006 IEEE International Conference on*, pages 525–528, July 2006. 228, 267, 269, 270, 271, 287, 294

[TL07] J Tang and P Lewis. An image based feature space and mapping for linking regions and words. In *Proceedings of 2nd International Conference on Computer Vision Theory and Applications*, page 2935, 2007. 224

[TMY78] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, June 1978. 194

[TNY+06] H. Takahashi, T. Nemoto, T. Yoshida, H. Honda, and T. Hasegawa. Cancer diagnosis marker extraction for soft tissue sarcomas based on gene expression profiling data by using projective adaptive resonance theory (part) filtering method. *BMC Bioinformatics*, 7(1):399–410, 2006. 171

[Tur86] M. R. Turner. Texture discrimination by gabor functions. *Biological Cybernetics*, 55(2):71–82, Nov. 1986. 194

[TZZR01] C. Tang, L. Zhang, A. Zhang, and M. Ramanathan. Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. *Bioinformatics and Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on*, pages 41–48, Nov 2001. 144

[Ved07] A. Vedaldi. An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD, 2007. 198, 239

[VFJHJ01] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, and Z. Hong-Jiang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, Jan 2001. 194

[VG02] A. Vinokourov and M. Girolami. A probabilistic framework for the hierarchic organisation and classification of document collections. *J. Intell. Inf. Syst.*, 18(2-3):153–172, 2002. 249, 257

[VR07] N. Villa and F. Rossi. A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. In *Proceedings of the 6th Workshop on Self-Organizing Maps (WSOM 07)*, Bielefeld, Germany, September 2007. 121

[WaTM05] J. Winn and A. Criminisi adn T. Minka. Object categorizaion by learned universal dictionary. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1800–1807, Oct. 2005. 221, 238, 240, 259

[WC92] M. P. Windham and A. Cutler. Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, 87(420):1188–1192, 1992. 32

[WC03] S. Wu and T. W. S. Chow. Self-organizing-map based clustering using a local clustering validity index. *Neural Processing Letters*, 17(3):253–271, 2003. 63

[WFCX06] Z. Wang, D. D. Feng, Z. Chi, and T. Xia. Annotating image regions using spatial context. In *Proceedigns of the Eighth IEEE International Symposium on Multimedia (ISM'06)*, Sep. 2006. 233

[WG07] Y. Wang and S. Gong. Translating topics to words for image annotation. In *Proceedings of the CIKM'07*, pages 995–998, Nov. 2007. 233

[WLL+07] L. Wu, M. Li, Z. Li, WY. Ma, and N. Yu. Visual language modeling for image classification. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 115–124, New York, NY, USA, 2007. ACM. 212

[WM98] C. L. Wiggs and A. Martin. Properties and mechanism of perceptual priming. *Current Opinion in Neurobioly*, (8):227–233, 1998. 17, 19, 20

[WS82] G. Wyszecki and W.S. Styles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley and Sons, New York, second edition edition, 1982. 238

[WWP00] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the European Conference on Computer Vision (ECCV'00)*, pages 18–32, 2000. 220

[WY02] KL. Wu and MS. Yang. Alternative c-means clustering algorithms. *Pattern Recognition*, 35(10):2267 – 2278, October 2002. 77, 80

[WY07] KL. Wu and MS. Yang. Mean shift-based clustering. *Pattern Recogn.*, 40(11):3035–3052, 2007. 36

[WZFF06] G. Wang, Y. Zhang, and L. Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, Washington, DC, USA, 2006. IEEE Computer Society. 219, 257

[XK01] E. P. Xing and R. M. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. pages 306–315, 2001. 144

[XKO93] L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. *IEEE Transactions on Neural Networks*, 4(4):636 – 649, 1993. 12, 25, 57, 93

[Xu03] L. Xu. Data smoothing regularization, multi-sets-learning, and problem solving strategies. *Neural Networks*, 16:817–825, 2003. 12, 29

[Xu07] L. Xu. A unified prerspective and new results on rht computing, mixture based learning and multi-learner based problem solving. *Pattern Recognition*, 40:2129–2153, 2007. 12, 26, 29, 30, 50, 81, 109

[XW05] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645 – 678, May 2005. 31, 62, 140

[Yav07] A. Yavlinsky. *Image indexing and retrieval using automated annotation*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, Jun. 2007. 224

[YPK+03] SH. Yoo, Y. S. Park, HR. Kim, S. W. Sung, J. H. Kim, Y. S. Shim, S. D. Lee, YL. Choi, MK. Kim, and D. H. Chung. Expression of caveolin-1 is associated with poor prognosis of patients with squamous cell carcinoma of the lung. *Lung Cancer*, 42(2):195–202, November 2003. 173

[YRS+02] E.J. Yeoh, M.E. Ross, S.A. Shurtleff, W.K. Williams, D. Patel, R. Mahfouz, F.G. Behm, S.C. Raimondi, and et al. Classification, subtype discovery and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133–143, 2002. 169

[YSR05] A. Yavlinsky, E. J. Schofield, and S. Rüger. Automated image annotation using global features and robust nonparametric density estimation. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *Proceedings of the 4th International Conference on Image and Video Retrieval (CIVR)*, volume 3568 of *Lecture Notes in Computer Science (LNCS)*, pages 507–517, Singapore, July 2005. Springer-Verlag. 228

[YW04] MS. Yang and KL. Wu. A similarity-based robust clustering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):434–448, 2004. 35, 47, 60, 61, 65, 76, 80

[YWY07] J. Yuan, Y. Wu, and M. Yang. From frequent itemsets to semantically meaningful visual patterns. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 864–873, New York, NY, USA, 2007. ACM. 212

[YZG92] E. Yair, K. Zeger, and A. Gersho. Competitive learning and soft competition for vector quantizer design. *IEEE Trans. on Sign. Proc.*, 40(2):294–309, feb 1992. 11, 24, 29, 52, 101, 106, 115, 125

[ZB02] Z.Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, Mar 2002. 131, 132

[ZB07] M. Zhu and A. Badii. Semantic-associative visual content labelling and retrieval: A multimodal approach. *Signal Processing: Image Communication*, 22:569–582, 2007. 233

[ZC04a] DQ. Zhang and SC. Chen. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32(1):37–50, 2004. 93, 110

[ZC04b] W. Zhili and L. Chunhung. A kernel-enabled RPCL algorithm. ACM (Hong Kong Chapter) Postgraduate Research Day(PG Day 2004), 2004. 93

[ZL02] YJ. Zhang and ZQ. Liu. Self-splitting competitive learning: a new on-line clustering paradigm. *IEEE Transactions on Neural Networks*, 13(2):369–380, Mar 2002. 33

[ZW94] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV '94: Proceedings of the third European conference on Computer Vision (Vol. II)*, pages 151–158, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. 196

[ZWG06] QF. Zheng, WQ. Wang, and W. Gao. Effective and efficient object-based image retrieval using visual phrases. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 77–80, New York, NY, USA, 2006. ACM. 212

[ZWZ+07] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: Incorporating keyword correlations and region matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CIVR'07)*, pages 25–32, Jul. 2007. 233