Dipartimento di Informatica e
Scienze dell'Informazione

Central Clustering in Kernel-Induced Spaces

by

Maurizio Filippone

Theses Series **DISI-TH-2008-03**

DISI, Università di Genova
v. Dodecaneso 35, 16146 Genova, Italy
http://www.disi.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica e
Scienze dell'Informazione

Dottorato di Ricerca in Informatica

Ph.D. Thesis in Computer Science

Central Clustering in Kernel-Induced Spaces

by

Maurizio Filippone

June, 2008

Dottorato di Ricerca in Informatica
Dipartimento di Informatica e Scienze dell'Informazione
Università degli Studi di Genova

DISI, Univ. di Genova
via Dodecaneso 35
I-16146 Genova, Italy
http://www.disi.unige.it/

Ph.D. Thesis in Computer Science (S.S.D. INF/01)

Submitted by Maurizio Filippone
DISI, Univ. di Genova
filippone@disi.unige.it

Date of submission: February 2008

Date of revision: June 2008

Title: Central Clustering in Kernel-Induced Spaces

Advisor: Francesco Masulli
DISI, Univ. di Genova
masulli@disi.unige.it

Ext. Reviewers: Mark Girolami
Department of Computing Science, University of Glasgow
girolami@dcs.gla.ac.uk

Alessandro Sperduti
Dipartimento di Matematica Pura ed Applicata, Univ. di Padova
sperduti@math.unipd.it

Abstract

Clustering is the problem of grouping objects on the basis of a similarity measure. Clustering algorithms are a class of useful tools to explore structures in data. Nowadays, the size of data collections is steadily increasing, due to high throughput measurement systems and mass production of information. This makes human intervention and analysis unmanageable without the aid of automatic and unsupervised tools. The use of kernels is definitely having an important place in the application of supervised machine learning tools to real problems. In recent years, the interest in the use of kernels in unsupervised applications has grown. The focus of this thesis is on new advances in the central clustering paradigm, with a special emphasis on kernel methods for clustering. In this thesis, we propose some advances in both the theoretical foundation and the application of such algorithms. The main contributions of the thesis can be summarized as follows:

1. A survey of kernel and spectral methods for clustering. We propose a classification of kernel methods for clustering, reporting the strong connections with spectral clustering. An explicit proof of the fact that these two paradigms optimize the same objective function is reported from literature.

2. A comparison of kernel and spectral methods for clustering. Many kernel and spectral clustering algorithms have not been experimentally validated on a wide range of real applications. We propose a comparative study of several clustering methods, based on kernels and spectral theory, on many synthetic and real data sets. The tested clustering algorithms, implemented during the work of thesis, have been collected in a software package written in R language.

3. Advances in fuzzy relational clustering. In this thesis, we study in detail the relational fuzzy clustering problem, proposing theoretical studies on the applicability of relational duals of central clustering algorithms in situations when patterns are described in terms on non-metric pairwise dissimilarities. As a byproduct, the equivalence between clustering of patterns represented by pairwise dissimilarities and central clustering in kernel-induced spaces is shown.

4. A novel algorithm, named the Possibilistic c-Means in Feature Space, that is a clustering method in feature space based on the possibilistic approach to clustering. The proposed algorithm can be used also for non-parametric density estimation and as an

outlier detection algorithm. We show the strong connections between this learning model and One Class SVM. The regularized properties and the simple optimization procedure suggest the potentialities of this novel algorithm in applications. The experimental validation consists in a comparison with One Class SVM in the context of outlier detection on synthetic and real data sets.

To my family

In theory there is no difference between theory and practice.
In practice there is. (L. P. Berra / J. L. A. van de Snepscheut)

Acknowledgements

There are a lot of people I have to thank for their support during these three years. A special thanks goes to my family and Erica, for supporting my choices when I needed to take some hard decisions. I want to thank my friends at DISI (ViCoLab and 210) and all my friends (in particular, my training buddies and my band), for sharing with me several moments. Among them, a special thanks goes to Gabriele, Federico, and Dario... they know what they did for me. In the last year, I had the chance to live a unique experience in USA, thanks to extraordinary people like Carlotta, Daniel, and Zoran, whose influence has been very important. Among all the people I met in USA, I want to thank also Marko, Sean, Jessica, Rick, Ed, Aline, Cristiane, Andrew, and Ayanna; without them, it would have been hard to live there. Finally, I want to thank all the people I exchanged ideas with, in particular: Federico, Dario, Franco, Stefano, Francesco, Carlotta, Daniel, Zoran, Matte, Giuse, and Giorgio. [1]

[1] Ci sono molte persone che vorrei ringraziare per il loro sostegno durante questi tre anni. Un ringraziamento speciale va alla mia famiglia e ad Erica per aver sostenuto le mie scelte quando ho dovuto prendere decisioni difficili. Voglio ringraziare i miei amici del DISI (ViCoLab e 210) e tutti i miei amici (in particolare i miei compagni di allenamento e la mia band), per aver condiviso con me molti momenti. Fra loro, un ringraziamento speciale va a Gabriele, Federico e Dario... loro sanno quello che hanno fatto per me. Lo scorso anno, ho avuto la possibilità di vivere un'esperienza unica negli USA, grazie a persone straordinarie come Carlotta, Daniel e Zoran, la cui influenza è stata molto importante. Fra tutte le persone che ho conosciuto negli USA, vorrei ringraziare anche Marko, Sean, Jessica, Rick, Ed, Aline, Cristiane, Andrew e Ayanna; senza di loro sarebbe stato difficile vivere lì. Infine, vorrei ringraziare tutte le persone con cui ho scambiato idee, in particolare: Federico, Dario, Franco, Stefano, Francesco, Carlotta, Daniel, Zoran, Matte, Giuse e Giorgio.

Contents

Chapter 1

Introduction

Clustering is the problem of grouping objects on the basis of a similarity measure. Clustering algorithms represent a class of useful tools to explore structures in data. Nowadays, the size of data collections is steadily increasing, due to high throughput measurement systems and mass production of information. This makes human intervention and analysis unmanageable, without the aid of automatic and unsupervised tools. The use of kernels is definitely having an important place in the application of supervised machine learning tools to real problems. In recent years, the interest in the use of kernels in unsupervised applications is growing.

The focus of this thesis is on new developments in the central clustering paradigm, with a special emphasis on kernel methods for clustering. In this thesis, we propose some advances in the use of kernel in clustering applications. Such advances, are both in the theoretical foundation of these algorithms and in their extensive application to synthetic and real data sets. In particular, the main contributions of the thesis can be summarized as follows:

A survey of kernel and spectral methods for clustering. We propose a classification of kernel methods for clustering, reporting the strong connections with spectral clustering. We can broadly classify kernel approaches to clustering in three categories, based respectively on the kernelization of the metric, clustering in feature space, and description via support vectors. Methods based on kernelization of the metric search for centroids in input space, and compute the distances between patterns and centroids by means of kernels. Clustering in feature space is made by mapping each pattern in the kernel-induced space; centroids are then computed in this new space. The description via support vectors makes use of the quadratic programming approach to find a minimum enclosing sphere in the kernel-induced space. This sphere encloses almost all data excluding the outliers, and creates a separation among clusters in the input space. Spectral clustering arises from spectral graph theory. The

clustering problem is configured as a graph cut problem, where an appropriate objective function has to be optimized. The connection between kernel and spectral methods for clustering lies in the equivalence between the objective function of the K-means in the kernel-induced space and the spectral clustering algorithm minimizing the ratio association objective function. An explicit proof of the fact that these two algorithms optimize the same objective function is reported from literature.

A comparison of kernel and spectral methods for clustering. Many kernel and spectral clustering algorithms have not been experimentally validated on a wide range of real applications. We propose a comparative study of several clustering methods based on kernels and spectral theory. This comparison is conducted on many synthetic and real data sets, showing the performances of the studied clustering algorithms in comparison with those of standard methods. The tested clustering algorithms, implemented during these years have been collected in a software package written in R language.

Advances in fuzzy relational clustering. When patterns are represented by means of non-metric pairwise dissimilarities, central clustering algorithms cannot be directly applied. Moreover, their relational duals are not guaranteed to converge. Symmetrization and shift operations have been proposed to transform the dissimilarities between patterns from non-metric to metric. It has been shown that they modify the K-Means objective function by a constant, that does not influence the optimization procedure. In literature, some fuzzy clustering algorithms have been extended, in order to handle patterns described by means of pairwise dissimilarities. The literature, however, lacks of an explicit analysis on what happens to central fuzzy clustering algorithms, when the dissimilarities are transformed to become metric. In this thesis, we study in detail the relational fuzzy clustering problem, proposing theoretical studies on the applicability of relational duals of central clustering algorithms in situations when patterns are described in terms on non-metric pairwise dissimilarities. As a byproduct, the equivalence between clustering of patterns represented by pairwise dissimilarities and clustering in kernel-induced spaces is shown.

A novel algorithm, named the Possibilistic c-Means in Feature Space, that is a kernel method for clustering in feature space based on the possibilistic approach to clustering. The proposed algorithm can be used also for non-parametric density estimation, as it retains the properties of the possibilistic clustering, and as an outlier detection algorithm. In this thesis, we study the strong connection of the proposed model with One Class Support Vector Machines. In particular, we show the duality between the Lagrange multipliers in One Class Support Vector Machines and the memberships in One Cluster Possibilistic c-Means in feature space and the regularized properties of the proposed algorithm. These facts, along with the simple optimization procedure, suggest the potentialities of this novel algorithm in applications. The

experimental validation shows a comparison with One Class Support Vector Machines in the context of outlier detection. Such comparison is performed by means of a statistics-based procedure.

The thesis is organized as follows: Chapter 2 introduces the state of the art on clustering. Chapter 3 shows a comparison of kernel and spectral clustering methods on some benchmarks and real data sets. Chapter 4 is devoted to the advances in relational clustering; Chapter 5 introduces the possibilistic clustering in feature space. The last Chapter draws the conclusions.

Chapter 2

State of the art

In this Chapter, we review the state of the art on kernel, spectral, and relational clustering algorithms. This survey contains the general background that represented the starting point of the original contributions proposed in this thesis. The Chapter is divided in four Sections. The first Section introduces some basic concepts about the central clustering paradigm. Here, the K-means and the fuzzy and possibilistic variants are discussed. The second and third Sections are devoted to the kernel and spectral methods for clustering, along with the discussions on the theoretical connections between them. Section 2 contains a categorization of kernel methods for clustering in three main families: clustering in feature space, clustering with the kernelization of the metric, and support vector clustering. In particular, the kernel versions of central clustering algorithms are presented. Section 3 introduces the clustering algorithms based on the spectral graph theory, showing two among the most popular algorithms. Moreover, we report from literature the formal equivalence between the objective functions of the spectral clustering with the ratio association as objective function and K-means in feature space. Section 4 extends the survey to the class of relational clustering algorithms, with a special emphasis on the relational duals of the central clustering algorithms. Several parts of Sections 1, 2, and 3 have been published as a survey paper [FCMR08].

The overview of the state of the art is not complete. Some approaches involving kernels, that are worth to be mentioned, are not present in this thesis. In particular, there is a vast literature comprising the probabilistic and Bayesian approaches to clustering [BR93, Bis06, RD06, LJGP07, OMSRA07, Mac02]. Also, Gaussian processes [RW05, Bis06], that have been widely studied for regression and classification, have been recently applied to the clustering problem [KL07]. In Gaussian processes, and more in general in stochastic processes, kernels arise in a natural way. Therefore, when such methods are applied to the clustering problem, they belong to the framework of kernel methods for clustering.

2.1 Central Clustering

In this section we briefly recall some basic facts about partitioning clustering methods. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a data set composed by n *patterns*, for which every $\mathbf{x}_i \in \mathbb{R}^d$. The *codebook* (or *set of centroids*) V is defined as the set $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$, typically with $c \ll n$. Each element $\mathbf{v}_i \in \mathbb{R}^d$ is called *codevector* (or *centroid* or *prototype*)[1].

The *Voronoi region* R_i of the codevector $\mathbf{v}_i$ is the set of vectors in $\mathbb{R}^d$ for which $\mathbf{v}_i$ is the nearest vector:

$$R_i = \left\{ \mathbf{z} \in \mathbb{R}^d \;\middle|\; i = \arg\min_j \|\mathbf{z} - \mathbf{v}_j\|^2 \right\} . \tag{2.1}$$

It is possible to prove that each Voronoi region is convex [LBG80], and the boundaries of the regions are linear segments.

The definition of the *Voronoi set* π_i of the codevector $\mathbf{v}_i$ is straightforward. It is the subset of X for which the codevector $\mathbf{v}_i$ is the nearest vector:

$$\pi_i = \left\{ \mathbf{x} \in X \;\middle|\; i = \arg\min_j \|\mathbf{x} - \mathbf{v}_j\|^2 \right\} , \tag{2.2}$$

that is, the set of patterns belonging to R_i. A partition on $\mathbb{R}^d$ induced by all Voronoi regions is called *Voronoi tessellation* or *Dirichlet tessellation*.

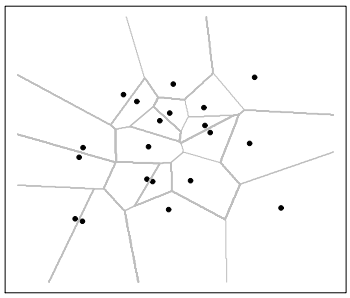


Figure 2.1: An example of Voronoi tessellation where each black point is a codevector.

[1] We will use indifferently these terms to denote the elements of V.

2.1.1 K-Means

A simple algorithm able to construct a Voronoi tessellation of the input space was proposed in 1957 by Lloyd [Llo82], and it is known as *batch K-Means*. Starting from the finite data set X, this algorithm moves iteratively the k codevectors to the arithmetic mean of their Voronoi sets $\{\pi_i\}_{i=1,\dots,k}$. Theoretically speaking, a necessary condition for a codebook V to minimize the *Empirical Quantization Error*:

$$E(X) = \frac{1}{2n} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \pi_i} \|\mathbf{x} - \mathbf{v}_i\|^2 \tag{2.3}$$

is that each codevector $\mathbf{v}_i$ fulfills the *centroid condition* [GG92]. In the case of a finite data set X and with Euclidean distance, the centroid condition reduces to:

$$\mathbf{v}_i = \frac{1}{|\pi_i|} \sum_{\mathbf{x} \in \pi_i} \mathbf{x} \; . \tag{2.4}$$

Batch K-Means is formed by the following steps:

1. choose the number k of clusters;
2. initialize the codebook V with vectors randomly picked from X;
3. compute the Voronoi set π_i associated to the codevector $\mathbf{v}_i$;
4. move each codevector to the mean of its Voronoi set using Eq. 2.4;
5. return to step 3 if any codevector has changed, otherwise return the codebook.

At the end of the algorithm a codebook is found and a Voronoi tessellation of the input space is provided. It is guaranteed that after each iteration the quantization error does not increase. Batch K-Means can be viewed as an *Expectation-Maximization* [Bis96] algorithm, ensuring the convergence after a finite number of steps.

This approach presents many disadvantages [DH73b]. Local minima of $E(X)$ make the method dependent on initialization, and the codevectors are sensitive to outliers. Moreover, the number of clusters to find must be provided, and this can be done only using some a priori information or additional validity criterion. Finally, K-Means can deal only with clusters with spherically symmetrical point distribution, since Euclidean distances of patterns from centroids are computed, leading to a spherical invariance. Different distances lead to different invariance properties as in the case of Mahalanobis distance which produces invariance on ellipsoids [DH73b].

Batch K-Means takes into account the whole data set to update the codevectors. When the cardinality n of the data set X is very high (e.g., several hundreds of thousands), the batch procedure is computationally expensive. For this reason, an on-line update has been introduced, leading to the *on-line K-Means* algorithm [LBG80, Mac67]. At each step, this method simply randomly picks an input pattern and updates its nearest codevector, ensuring that the scheduling of the updating coefficient is adequate to allow convergence and consistency.

2.1.2 Fuzzy Central Clustering Algorithms

In many applications, it is desirable that the membership of patterns is shared among clusters. This would allow to better describe situations where some patterns can belong to overlapped clusters, or some patterns do not belong to any clusters, since they are outliers. These are few examples of scenarios where the generalization of the concept of membership from crisp to fuzzy can be useful in applications.

Popular fuzzy central clustering algorithms are the fuzzy versions of the K-means with the probabilistic and possibilistic description of the memberships: Fuzzy c-means [Bez81] and Possibilistic c-means [KK93]. These algorithms, belonging to the K-means family, are based on the concept of centroids and memberships. Given a set X of n patterns, the set of centroids $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_c\}$ and the membership matrix U are defined. The set V contains the prototypes/representatives of the c clusters. U is a $c \times n$ matrix where each element u_{ih} represents the membership of the pattern h to the cluster i. Both Fuzzy and Possibilistic c-means are fuzzy, since $u_{ih} \in [0, 1]$, while $u_{ih} \in \{0, 1\}$ for K-means. In K-means and FCM algorithms the memberships of a pattern to all the c clusters are constraint to sum up to one:

$$\sum_{i=1}^{c} u_{ih} = 1 \quad \forall k = 1, \ldots, n \tag{2.5}$$

This is the so called *Probabilistic Constraint*. In the possibilistic paradigm, this constraint is relaxed, leading to an interpretation of the membership as a degree of typicality.

Among the presented fuzzy central clustering algorithms, we may recognize an algorithm, the FCM II, resembling the Expectation Maximization algorithm for fitting a mixture of isotropic Gaussians [Bis06]. It can be found also with the name of Soft K-means [Mac02]. For the sake of presentation, however, we prefer to present all the fuzzy algorithms under the same formalism, viewing them as the optimization of a general objective function.

The formal definition of fuzzy c-partition is the following [Bez81]:

Definition 2.1.1. *Let A_{cn} denote the vector space of $c \times n$ real matrices over $\mathbb{R}$. Considering X, A_{cn} and $c \in \mathbb{N}$ such that $2 \leq c < n$, the* Fuzzy c-partition space for X *is the*

set:

$$M_{\mathrm{fc}} = \left\{ U \in A_{cn} \;\middle|\; u_{ih} \in [0,1]\, \forall i,h;\; \sum_{i=1}^{c} u_{ih} = 1\, \forall h\;;\; 0 < \sum_{h=1}^{n} u_{ih} < n\, \forall i \right\} . \tag{2.6}$$

This definition generalizes the notion of hard c-partitions in Ref. [Bez81].

In general, all the K-means family algorithms are based on the minimization of a functional composed of two terms:

$$J(U,V) = G(U,V) + H(U) \tag{2.7}$$

The first term is a measure of the distortion (or intra-cluster distance) and the second is an entropic score on the memberships. The distortion can be written as the following sum:

$$G(U,V) = 2 \sum_{i=1}^{c} \sum_{h=1}^{n} u_{ih}^{\theta} \|\mathbf{x}_h - \mathbf{v}_i\|^2 \tag{2.8}$$

with $\theta \geq 1$. The aim of the entropy term $H(U)$ is to avoid trivial solutions where all the memberships are zero or equally shared among the clusters. For the algorithms having a constraint on U, the Lagrange multipliers technique has to be followed in order to perform the optimization. This means that a further term, depending only from U, must be added to $J(U,V)$. The Lagrangian associated to the optimization problem can be introduced:

$$L(U,V) = G(U,V) + H(U) + W(U) \tag{2.9}$$

The technique used by these methods to perform the minimization is the so called Picard iteration technique [Bez81]. The Lagrangian $L(U,V)$ depends on two groups of variables U and V related to each other, namely $U = U(V)$ and $V = V(U)$. In each iteration one of the two groups of variables is kept fixed, and the minimization is performed with respect to the other group. In other words:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = 0 \tag{2.10}$$

with U fixed, gives a formula for the update of the centroids $\mathbf{v}_i$, and:

$$\frac{\partial L(U,V)}{\partial u_{ih}} = 0 \tag{2.11}$$

with V fixed, gives a formula for the update of the memberships u_{ih}. The algorithms start by randomly initializing U or V, and iteratively update U and V by means of the previous two equations. It can be proved that the value of L does not increase after each

iteration [HK03]. The algorithms stop when a convergence criterion is satisfied on U, V or G. Usually the following is considered:

$$\|U - U'\|_p < \varepsilon \tag{2.12}$$

where U' is the updated version of the memberships and $\|\|_p$ is a p-norm.

We now explicitly derive the update equations of four clustering algorithms based on fuzzy memberships, in particular: Fuzzy c-means I (FCM I) [Bez81], Fuzzy c-means II (FCM II) [BL94], Possibilistic c-means I (PCM I) [KK93], and Possibilistic c-means II (PCM II) [KK96]. In the following derivations, we will use the Euclidean distance function:

$$d_{\mathrm{E}}^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) = \sum_{i=1}^{d} (x_i - y_i)^2$$

with $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^d$. It is worth noting that it is possible to use other distance functions, e.g., Minkowski:

$$d_{\mathrm{M}}^p(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} (x_i - y_i)^p$$

with $p \geq 1$, that extends the Euclidean distance. For all values of p, such distance lead to hyperspherical clusters. Other popular choices of metrics extending the Euclidean one are those belonging to this class:

$$d_A = (\mathbf{x} - \mathbf{y})^T A^{-1} (\mathbf{x} - \mathbf{y})$$

with A invertible. When A is the identity matrix, d_A corresponds to the Euclidean distance. When A is diagonal, the features are scaled, leading to ellipsoid-shaped clusters oriented along the directions of the features. When A is the covariance matrix, d_A is the so called Mahalanobis distance, that leads to ellipsoid-shaped clusters oriented along the principal components of data. Other modifications are represented by the incorporation of prior information on the shape of clusters. These algorithms, called Fuzzy c-Varieties, can be found in Ref. [Bez81].

2.1.2.1 Fuzzy c-means I - FCM I

The Lagrangian $L(U)$ is introduced [Bez81]:

$$L(U, V) = \sum_{i=1}^{c} \sum_{h=1}^{n} u_{ih}^m \|\mathbf{x}_h - \mathbf{v}_i\|^2 + \sum_{h=1}^{n} \beta_h (1 - \sum_{i=1}^{c} u_{ih}) \tag{2.13}$$

The first term is the distortion $G(U, V)$ and the second is $W(U)$ which is not zero, since the memberships are subject to the probabilistic constraint in Eq. 2.5. The parameter $m > 1$

works as a fuzzifier parameter; for high values of m, the memberships tend to be equally distributed among clusters. Setting to zero the derivatives of $L(U,V)$ with respect to u_{ih}:

$$\frac{\partial L(U,V)}{\partial u_{ih}} = m u_{ih}^{m-1} \|\mathbf{x}_h - \mathbf{v}_i\|^2 - \beta_h = 0 \tag{2.14}$$

we obtain:

$$u_{ih} = \left(\frac{\beta_h}{m\|\mathbf{x}_h - \mathbf{v}_i\|^2} \right)^{\frac{1}{m-1}} \tag{2.15}$$

Substituting the expression of u_{ih} into the constraint equation:

$$\sum_{i=1}^{c} \left(\frac{\beta_h}{m\|\mathbf{x}_h - \mathbf{v}_i\|^2} \right)^{\frac{1}{m-1}} = 1 \tag{2.16}$$

we can obtain the Lagrange multipliers:

$$\beta_h = \left[\sum_{i=1}^{c} \left(\frac{1}{m\|\mathbf{x}_h - \mathbf{v}_i\|^2} \right)^{\frac{1}{m-1}} \right]^{1-m} \tag{2.17}$$

Substituting Eq. 2.17 into Eq. 2.15, the equation for the update of the memberships u_{ih} can be obtained:

$$u_{ih}^{-1} = \sum_{j=1}^{c} \left(\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\|\mathbf{x}_h - \mathbf{v}_j\|^2} \right)^{\frac{1}{m-1}} \tag{2.18}$$

To compute the equation for the update of the $\mathbf{v}_i$, we set to zero the derivatives of $L(U,V)$ with respect to $\mathbf{v}_i$:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = -\sum_{h=1}^{n} u_{ih}^m \left(\mathbf{x}_h - \mathbf{v}_i \right) = 0 \tag{2.19}$$

obtaining:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih}^m \mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}^m} \tag{2.20}$$

2.1.2.2 Fuzzy c-means II - FCM II

This algorithm, also known as soft K-means, fits a mixture of isotropic Gaussians. In the formalism we are using in this thesis, the FCM II Lagrangian $L(U,V)$ is [BL94]:

$$L(U,V) = \sum_{h=1}^{n} \sum_{i=1}^{c} u_{ih} \|\mathbf{x}_h - \mathbf{v}_i\|^2 + \lambda \sum_{h=1}^{n} \sum_{i=1}^{c} u_{ih} \ln(u_{ih}) + \sum_{h=1}^{n} \beta_h (1 - \sum_{i=1}^{c} u_{ih}) \tag{2.21}$$

The entropic term favors values of the memberships near zero or one (Fig. 2.2). Let's compute the derivative of $L(U,V)$ with respect to u_{ih}:

$$\frac{\partial L(U,V)}{\partial u_{ih}} = \|\mathbf{x}_h - \mathbf{v}_i\|^2 + \lambda(\ln(u_{ih}) + 1) - \beta_h = 0 \tag{2.22}$$

This leads to:

$$u_{ih} = \frac{1}{e} \exp\left(\frac{\beta_h}{\lambda}\right) \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\lambda}\right) \tag{2.23}$$

Substituting the last equation into the probabilistic constraint, we obtain:

$$\sum_{i=1}^{c} \frac{1}{e} \exp\left(\frac{\beta_h}{\lambda}\right) \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\lambda}\right) = 1 \tag{2.24}$$

This allows to compute the Lagrange multipliers:

$$\beta_h = \lambda - \lambda \ln\left(\sum_{j=1}^{c} \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_j\|^2}{\lambda}\right)\right) \tag{2.25}$$

Substituting Eq. 2.25 into Eq. 2.23, we obtain the equation for the update of the u_{ih}:

$$u_{ih} = \frac{\exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\lambda}\right)}{\sum_{j=1}^{c} \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_j\|^2}{\lambda}\right)} \tag{2.26}$$

Setting to zero the derivatives of $L(U,V)$ with respect to $\mathbf{v}_i$:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = -\sum_{h=1}^{n} u_{ih} (\mathbf{x}_h - \mathbf{v}_i) = 0 \tag{2.27}$$

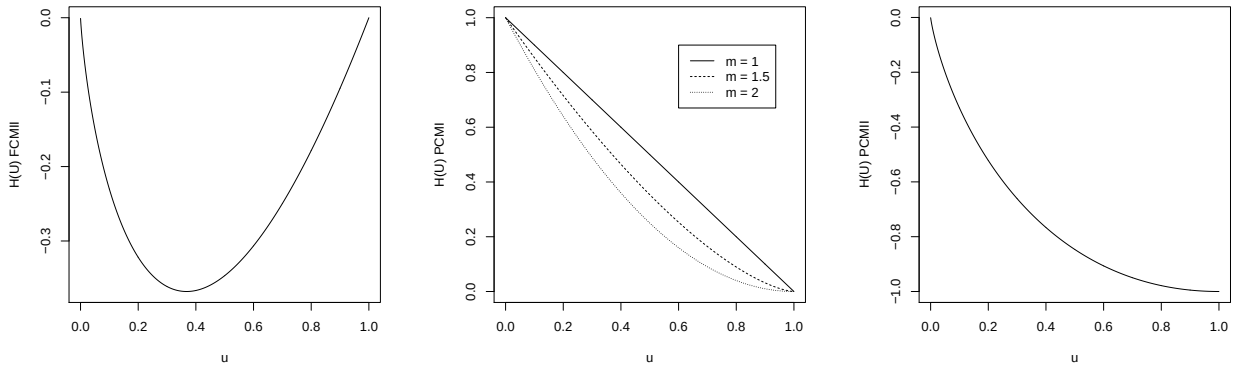
the following update formula for the centroids $\mathbf{v}_i$ is obtained:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih} \mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}} \tag{2.28}$$

2.1.2.3 Possibilistic c-means I - PCM I

The PCM I Lagrangian $L(U,V)$ does not have the $W(U)$ term coming from the probabilistic constraint on the memberships [KK93]:

$$L(U,V) = \sum_{h=1}^{n} \sum_{i=1}^{c} u_{ih}^m \|\mathbf{x}_h - \mathbf{v}_i\|^2 + \sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (1 - u_{ih})^m \tag{2.29}$$



(a) (b) (c)

Figure 2.2: (a) Plot of the FCM II entropy $H(u_{ih}) = u_{ih}\ln(u_{ih})$. (b) Plot of the PCM I entropy $H(u_{ih}) = (1-u_{ih})^m$ for increasing values of m. (c) Plot of the PCM II entropy $H(u_{ih}) = u_{ih}\ln(u_{ih}) - u_{ih}$.

The entropic term penalizes small values of the memberships.

Setting to zero the derivatives of $L(U,V)$ with respect to the memberships u_{ih}:

$$\frac{\partial L(U,V)}{\partial u_{ik}} = m u_{ih}^{m-1}(\|\mathbf{x}_h - \mathbf{v}_i\|^2) - \eta_i m(1-u_{ih})^{m-1} = 0 \tag{2.30}$$

We obtain directly the update equation:

$$u_{ih}^{-1} = \left(\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\eta_i}\right)^{\frac{1}{m-1}} + 1 \tag{2.31}$$

The following derivative of $L(U,V)$:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = -\sum_{h=1}^{n} u_{ih}^m (\mathbf{x}_h - \mathbf{v}_i) = 0 \tag{2.32}$$

gives the update equation for the centroids $\mathbf{v}_i$:

$$\mathbf{v}_i = \frac{\sum_{h=1}^n u_{ih}^m \mathbf{x}_h}{\sum_{h=1}^n u_{ih}^m} \tag{2.33}$$

The following criterion is suggested to estimate the value of η_i:

$$\eta_i = \gamma \frac{\sum_{h=1}^n (u_{ih})^m \|\mathbf{x}_h - \mathbf{v}_i\|^2}{\sum_{h=1}^n (u_{ih})^m} \tag{2.34}$$

where γ is usually set to one. In order to have a reliable estimation of η_i, it is suggested to use the results of a fuzzy clustering algorithm (e.g., FCM I or FCM II).

2.1.2.4 Possibilistic c-means II - PCM II

The PCM II Lagrangian $L(U, V)$ is [KK96]:

$$L(U,V) = \sum_{h=1}^{n}\sum_{i=1}^{c} u_{ih}\|\mathbf{x}_h - \mathbf{v}_i\|^2 + \sum_{i=1}^{c}\eta_i \sum_{h=1}^{n}\left(u_{ih}\ln(u_{ih}) - u_{ih}\right) \tag{2.35}$$

The entropic term penalizes small values of the memberships.

Setting to zero the derivatives of $L(U, V)$ with respect to the memberships u_{ih}:

$$\frac{\partial L(U,V)}{\partial u_{ik}} = \|\mathbf{x}_h - \mathbf{v}_i\|^2 + \eta_i \ln(u_{ih}) = 0 \tag{2.36}$$

we obtain:

$$u_{ik} = \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\eta_i}\right) \tag{2.37}$$

Setting to zero the derivatives of $L(U, V)$ with respect to $\mathbf{v}_i$:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = -\sum_{h=1}^{n} u_{ih}\left(\mathbf{x}_h - \mathbf{v}_i\right) = 0 \tag{2.38}$$

we obtain the update formula for the centroids $\mathbf{v}_i$:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih}\mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}} \tag{2.39}$$

As in the PCM I, the value of η_i can be estimated as:

$$\eta_i = \gamma\frac{\sum_{h=1}^{n} u_{ih}\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\sum_{h=1}^{n} u_{ih}} \tag{2.40}$$

2.2 Kernel Methods for Clustering

In machine learning, the use of the kernel functions [Mer09] has been introduced by Aizerman et al. [ABR64] in 1964. In 1995 Cortes and Vapnik introduced *Support Vector Machines* (SVMs) [CV95], which perform better than other classification algorithms in several problems. The success of SVM has brought to extend the use of kernels to other learning algorithms (e.g., *Kernel PCA* [SSM98]). The choice of the kernel is crucial to incorporate a priori knowledge on the application, for which it is possible to design *ad hoc* kernels.

2.2.1 Mercer kernels

We recall the definition of Mercer kernels [Aro50, Sai88], considering, for the sake of simplicity, vectors in $\mathbb{R}^d$ instead of $\mathbb{C}^d$.

Definition 2.2.1. *Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a nonempty set where $\mathbf{x}_i \in \mathbb{R}^d$. A function $K : X \times X \to \mathbb{R}$ is called a* positive definite kernel *(or* Mercer kernel*) if and only if K is symmetric (i.e. $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$) and the following equation holds:*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \quad \forall n \geq 2 \,, \tag{2.41}$$

where $c_r \in \mathbb{R}\ \forall r = 1, \ldots, n$

Each Mercer kernel can be expressed as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \,, \tag{2.42}$$

where $\Phi : X \to \mathcal{F}$ performs a mapping from the input space X to a high dimensional *feature space* $\mathcal{F}$. One of the most relevant aspects in applications is that it is possible to compute Euclidean distances in $\mathcal{F}$ without knowing explicitly Φ. This can be done using the so called *distance kernel trick* [MMR+01, SSM98]:

$$\begin{aligned} \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 &= (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)) \cdot (\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)) \\ &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) + \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_j) - 2\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \\ &= K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{2.43}$$

in which the computation of distances of vectors in feature space is just a function of the input vectors. In fact, every algorithm where input vectors appear only in dot products with other input vectors can be kernelized [SS01]. In order to simplify the notation, we introduce the so called *Gram matrix* K, having its entries k_{ij} representing the scalar product $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Thus, Eq. 2.43 can be rewritten as:

$$\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 = k_{ii} + k_{jj} - 2k_{ij} \,. \tag{2.44}$$

Examples of Mercer kernels are the following [Vap95]:

- linear:
$$K^{(\mathrm{l})}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.45}$$
- polynomial of degree p:
$$K^{(\mathrm{p})}(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^p \qquad p \in \mathbb{N} \tag{2.46}$$

- Gaussian:

$$K^{(\mathrm{g})}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \qquad \sigma \in \mathbb{R} \tag{2.47}$$

It is important to stress that the use of the linear kernel in Eq. 2.43 simply leads to the computation of the Euclidean norm in the input space. Indeed:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= \mathbf{x}_i \cdot \mathbf{x}_i + \mathbf{x}_j \cdot \mathbf{x}_j - 2\mathbf{x}_i \cdot \mathbf{x}_j \\ &= K^{(\mathrm{l})}(\mathbf{x}_i, \mathbf{x}_i) + K^{(\mathrm{l})}(\mathbf{x}_j, \mathbf{x}_j) - 2K^{(\mathrm{l})}(\mathbf{x}_i, \mathbf{x}_j) \\ &= \|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 \ , \end{aligned} \tag{2.48}$$

shows that choosing the kernel $K^{(\mathrm{l})}$ implies $\Phi = I$ (where I is the identity function). Following this consideration, we can think that kernels can offer a more general way to represent the elements of a set X and possibly, for some of these representations, the clusters can be easily identified.

In literature there are some applications of kernels in clustering. These methods can be broadly divided in three categories, based respectively on:

- *kernelization of the metric* [WXY03, ZC03, ZC04];
- *clustering in feature space* [GO98, IM04, MF00, QS04, ZC02];
- *description via support vectors* [CV05, HHSV01].

Methods based on kernelization of the metric look for centroids in input space and the distances between patterns and centroids is computed by means of kernels:

$$\|\Phi(\mathbf{x}_h) - \Phi(\mathbf{v}_i)\|^2 = K(\mathbf{x}_h, \mathbf{x}_h) + K(\mathbf{v}_i, \mathbf{v}_i) - 2K(\mathbf{x}_h, \mathbf{v}_i) \ . \tag{2.49}$$

Clustering in feature space is made by mapping each pattern using the function Φ and then computing centroids in feature space. Calling $\mathbf{v}_i^{\Phi}$ the centroids in feature space, we will see in the next sections that it is possible to compute the distances $\left\|\Phi(\mathbf{x}_h) - \mathbf{v}_i^{\Phi}\right\|^2$ by means of the kernel trick.

The description via support vectors makes use of One Class SVM to find a minimum enclosing sphere in feature space able to enclose almost all data in feature space excluding outliers. The computed hypersphere corresponds to nonlinear surfaces in input space enclosing groups of patterns. The Support Vector Clustering algorithm allows to assign labels to patterns in input space enclosed by the same surface. In the next sub-sections we will outline these three approaches. We decided to include the clustering with the kernelization of the metric in the subsection about fuzzy clustering using kernels.

2.2.2 K-Means in Feature Space

Given the data set X, we map our data in some feature space $\mathcal{F}$, by means of a nonlinear map Φ and we consider k centers in feature space ($\mathbf{v}_i^\Phi \in \mathcal{F}$ with $i = 1, \ldots, k$) [Gir02, SSM98]. We call the set $V^\Phi = (\mathbf{v}_1^\Phi, \ldots, \mathbf{v}_k^\Phi)$ *Feature Space Codebook*, since in our representation the centers in the feature space play the same role of the codevectors in the input space. In analogy with the codevectors in the input space, we define for each center $\mathbf{v}_i^\Phi$ its *Voronoi Region* and *Voronoi Set* in feature space. The *Voronoi Region in feature space* (R_i^Φ) of the center $\mathbf{v}_i^\Phi$ is the set of all vectors in $\mathcal{F}$ for which $\mathbf{v}_i^\Phi$ is the closest vector

$$R_i^\Phi = \left\{ \mathbf{x}^\Phi \in \mathcal{F} \;\middle|\; i = \arg\min_j \left\| \mathbf{x}^\Phi - \mathbf{v}_j^\Phi \right\| \right\} . \tag{2.50}$$

The *Voronoi Set in Feature Space* π_i^Φ of the center $\mathbf{v}_i^\Phi$ is the set of all vectors $\mathbf{x}$ in X such that $\mathbf{v}_i^\Phi$ is the *closest vector* to their images $\Phi(\mathbf{x})$ in the feature space:

$$\pi_i^\Phi = \left\{ \mathbf{x} \in X \;\middle|\; i = \arg\min_j \left\| \Phi(\mathbf{x}) - \mathbf{v}_j^\Phi \right\| \right\} . \tag{2.51}$$

The set of the Voronoi Regions in feature space define a *Voronoi Tessellation of the Feature Space*. The Kernel K-Means algorithm has the following steps:

1. Project the data set X into a feature space $\mathcal{F}$, by means of a nonlinear mapping Φ.
2. Initialize the codebook $V^\Phi = (\mathbf{v}_1^\Phi, \ldots, \mathbf{v}_k^\Phi)$ with $\mathbf{v}_i^\Phi \in \mathcal{F}$
3. Compute for each center $\mathbf{v}_i^\Phi$ the set π_i^Φ
4. Update the codevectors $\mathbf{v}_i^\Phi$ in $\mathcal{F}$

$$\mathbf{v}_i^\Phi = \frac{1}{|\pi_i^\Phi|} \sum_{\mathbf{x} \in \pi_i^\Phi} \Phi(\mathbf{x}) \tag{2.52}$$

5. Go to step 3 until any $\mathbf{v}_i^\Phi$ changes
6. Return the feature space codebook.

This algorithm minimizes the quantization error in feature space.

Since we do not know explicitly Φ, it is not possible to compute directly Eq. 2.52. Nevertheless, it is always possible to compute distances between patterns and codevectors by using the kernel trick, allowing to obtain the Voronoi sets in feature space π_i^Φ. Indeed,

writing each centroid in feature space as a combination of data vectors in feature space, we have:

$$\mathbf{v}_j^{\Phi} = \sum_{h=1}^{n} \gamma_{jh} \Phi(\mathbf{x}_h) \ , \tag{2.53}$$

where γ_{jh} is one if $\mathbf{x}_h \in \pi_j^{\Phi}$ and zero otherwise. Now the quantity:

$$\left\| \Phi(\mathbf{x}_i) - \mathbf{v}_j^{\Phi} \right\|^2 = \left\| \Phi(\mathbf{x}_i) - \sum_{h=1}^{n} \gamma_{jh} \Phi(\mathbf{x}_h) \right\|^2 \tag{2.54}$$

can be expanded by using the scalar product and the kernel trick in Eq. 2.43:

$$\left\| \Phi(\mathbf{x}_i) - \sum_{h=1}^{n} \gamma_{jh} \Phi(\mathbf{x}_h) \right\|^2 = k_{ii} - 2 \sum_{h} \gamma_{jh} k_{ih} + \sum_{r} \sum_{s} \gamma_{jr} \gamma_{js} k_{rs} \ . \tag{2.55}$$

This allows to compute the closest feature space codevector for each pattern and to update the coefficients γ_{jh}. It is possible to repeat these two operations until any γ_{jh} changes to obtain a Voronoi tessellation of the feature space.

An on-line version of the kernel K-Means algorithm can be found in [SSM98]. A further version of K-Means in feature space has been proposed in Ref. [Gir02]. In his formulation the number of clusters is denoted by c and a fuzzy membership matrix U is introduced. Each element u_{ih} denotes the fuzzy membership of the point $\mathbf{x}_h$ to the Voronoi set π_i^{Φ}. This algorithm tries to minimize the following functional with respect to U:

$$J^{\Phi}(U, V^{\Phi}) = \sum_{h=1}^{n} \sum_{i=1}^{c} u_{ih} \left\| \Phi(\mathbf{x}_h) - \mathbf{v}_i^{\Phi} \right\|^2 \ . \tag{2.56}$$

The minimization technique used in Ref. [Gir02] is *Deterministic Annealing* [Ros98] which is a stochastic method for optimization. A parameter controls the fuzziness of the membership during the optimization and can be thought proportional to the temperature of a physical system. This parameter is gradually lowered during the annealing and at the end of the procedure the memberships have become crisp; therefore a tessellation of the feature space is found. This linear partitioning in $\mathcal{F}$, back to the input space, forms a nonlinear partitioning of the input space.

There are other popular clustering algorithms that have a formulation in kernel-induced spaces. For example, the the SOMs [Koh90] and Neural Gas [MBS93] versions in feature space can be find respectively in Refs. [IM04, MF00] and [QS04].

2.2.3 One-Class SVM

One among the approaches using kernels in clustering applications, is based on the support vector description of data [TD99, SS01]. The aim of this approach is to look for an hyper-

sphere centered in $\mathbf{v}$ containing almost all data, namely allowing some outliers. Support Vector Clustering (SVC) [HHSV01, HHSV00] takes advantage of this description to perform clustering. In particular, the support vector description of data in the kernel-induced space, leads to possibly non-linear surfaces separating the clusters in the original space. A labeling algorithm is necessary to assign the same label to the patterns belonging to the same region.

We show the derivation of this method, starting from the problem of finding an hypersphere in input space containing almost all data, extending it by means of kernels. The hypersphere has to enclose almost all patterns. This means that the distance between patterns and the center of the sphere $\mathbf{v}$ is less than the sum of the radius R and a positive amount:

$$\|\mathbf{x}_h - \mathbf{v}\|^2 \leq R^2 + \xi_h \qquad \xi_h \geq 0$$

The positive slack variables ξ_h take into account the fact that a pattern can be outside the sphere. The problem of finding such sphere can be formalized in the following way:

$$\min_{\mathbf{v},\xi_1,\ldots,\xi_n} R^2 \qquad \text{subject to :}$$

$$\|\mathbf{x}_h - \mathbf{v}\|^2 \leq R^2 + \xi_h \qquad \text{and} \qquad -\xi_h \leq 0$$

This problem can be reformulated by means of the Lagrange multiplier technique, where the functional has to be minimized under two inequality constraints. The Lagrangian L can be introduced:

$$L = R^2 - \sum_h \alpha_h \left(R^2 + \xi_h - \|\mathbf{x}_h - \mathbf{v}\|^2\right) - \sum_h \beta_h \xi_h + C \sum_h \xi_h \tag{2.57}$$

with the Lagrange multipliers satisfying the following:

$$\alpha_h \geq 0 \qquad \beta_h \geq 0$$

The parameter C regulates the number of patterns allowed to be outside the sphere.

To obtain the dual formulation of the problem, we set to zero the derivatives of L with respect to its variables:

$$\frac{\partial L}{\partial R} = 2R - 2R \sum_h \alpha_h = 0 \tag{2.58}$$

$$\frac{\partial L}{\partial \xi_h} = C - \alpha_h - \beta_h = 0 \tag{2.59}$$

$$\frac{\partial L}{\partial \mathbf{v}} = -2 \sum_h \alpha_h (\mathbf{x}_h - \mathbf{v}) \tag{2.60}$$

obtaining:

$$\sum_h \alpha_h = 1 \tag{2.61}$$

$$\beta_h = C - \alpha_h \tag{2.62}$$

$$\mathbf{v} = \sum_h \alpha_h \mathbf{x}_h \tag{2.63}$$

Note that Eq. 2.62 implies that $0 \leq \alpha_h \leq C$.

The Karush-Kuhn-Tucker (KKT) complementary conditions [Bur98] result in:

$$\beta_h \xi_h = 0, \quad \alpha_h \left(R^2 + \xi_h - \|\mathbf{x}_h - \mathbf{v}\|^2 \right) = 0. \tag{2.64}$$

Following simple considerations regarding all these conditions, it is possible to see that:

- when $\xi_h > 0$, the image of $\mathbf{x}_h$ lies outside the hypersphere. These points are called *bounded support vectors*. For them, $\alpha_h = C$ holds;
- when $\xi_h = 0$ and $0 < \alpha_h < C$, the image of $\mathbf{x}_h$ lies on the surface of the hypersphere. These points are called *support vectors*.
- when $\xi_h = 0$ and $\alpha_h = 0$, the image of $\mathbf{x}_h$ is inside the hypersphere.

The parameter C gives a bound on the number of bounded support vectors with respect to the cardinality of the data set. In order to have a direct interpretation on the number of bounded support vectors to select, C is sometimes substituted by ν:

$$\nu = \frac{1}{nC} \tag{2.65}$$

In the following we will use both these notations to select the number of outliers.

Substituting Eqs. 2.61-2.63 in L, we get the dual formulation of the Lagrangian that has to be maximized with respect to the Lagrange multipliers:

$$\begin{aligned}
L &= R^2 - \sum_h \alpha_h \left(R^2 + \xi_h - \|\mathbf{x}_h - \sum_r \alpha_r \mathbf{x}_r\|^2 \right) - \sum_h (C - \alpha_h)\xi_h + C \sum_h \xi_h \\
&= \sum_h \alpha_h \|\mathbf{x}_h - \sum_r \alpha_r \mathbf{x}_r\|^2 \\
&= \sum_h \alpha_h \mathbf{x}_h \mathbf{x}_h + \sum_h \alpha_h \sum_r \sum_s \alpha_r \alpha_s \mathbf{x}_r \mathbf{x}_s - 2 \sum_h \sum_r \alpha_h \alpha_r \mathbf{x}_h \mathbf{x}_r \\
&= \sum_h \alpha_h \mathbf{x}_h \mathbf{x}_h - \sum_r \sum_s \alpha_r \alpha_s \mathbf{x}_r \mathbf{x}_s
\end{aligned}$$

Since the last equation contains only scalar products between patterns, it is possible to substitute them by any positive semidefinite kernels, leading to the following optimization problem:

$$\min_{\alpha_1,\ldots,\alpha_n} \left(\sum_r \sum_s \alpha_r \alpha_s k_{rs} - \sum_h \alpha_h k_{hh} \right) \qquad \text{subject to :}$$

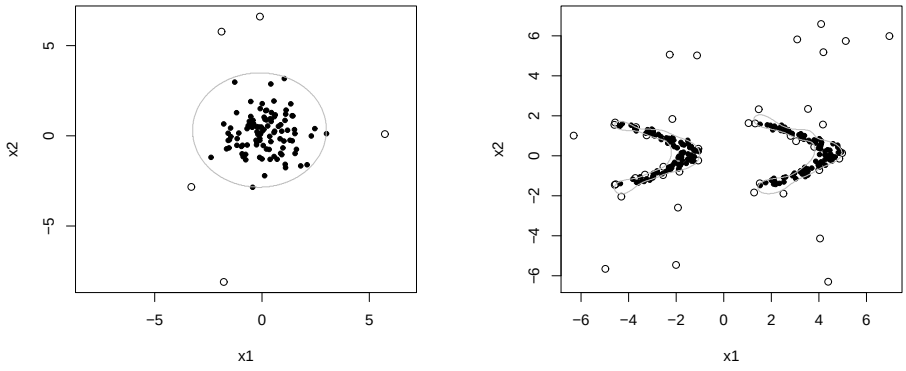
$$\sum_h \alpha_h = 1 \qquad \text{and} \qquad 0 \leq \alpha_h \leq C$$

The solution of this problem can be computed using SMO algorithm [Pla99]. In the case of Gaussian kernel, the problem is equivalent to find an hyperplane separating most of data points from outliers [SS01].

Once a solution is obtained, the distance from the image of a point $\mathbf{x}_h$ and the center $\mathbf{v}$ of the enclosing sphere can be computed as follows:

$$d_h = \|\Phi(\mathbf{x}_h) - \mathbf{v}\|^2 = k_{hh} - 2\sum_r \alpha_r k_{hr} + \sum_r \sum_s \alpha_r \alpha_s k_{rs} \tag{2.66}$$

In Fig. 2.3 it is possible to see the ability of this algorithm to find the smallest enclosing sphere without outliers.



(a) (b)

Figure 2.3: One class SVM applied to two data sets with outliers. The gray line shows the projection in input space of the smallest enclosing sphere in feature space. In (a) a linear kernel and in (b) a Gaussian kernel have been used.

2.2.3.1 Support Vector Clustering

Once boundaries in input space are found, a labeling procedure is necessary in order to complete clustering. In [HHSV01] the cluster assignment procedure follows a simple geometric idea. Any path connecting a pair of points belonging to different clusters must exit from the enclosing sphere in feature space. Denoting with Y the image in feature space of one of such paths and with $\mathbf{y}$ the elements of Y, it will result that $R(\mathbf{y}) > R$ for some $\mathbf{y}$. Thus it is possible to define an adjacency structure in this form:

$$\begin{cases} 1 & \text{if } R(\mathbf{y}) < R \quad \forall \mathbf{y} \in Y \\ 0 & \text{otherwise.} \end{cases} \tag{2.67}$$

Clusters are simply the connected components of the graph with the adjacency matrix just defined. In the implementation in Ref. [HHSV00], the matrix is constructed sampling the line segment Y in 20 equidistant points. There are some modifications on this labeling algorithm (e.g., [Lee05, YEC02]) that improve performances. An improved version of SVC algorithm with application in handwritten digits recognition can be found in Ref. [CH03]. A technique combining K-Means and One Class SVM can be found in Ref. [CV05].

2.2.4 Kernel fuzzy clustering methods

2.2.4.1 FCM I with the kernelization of the metric

The basic idea is to minimize the functional [WXY03, ZC03, ZC04]:

$$J^{\Phi}(U,V) = \sum_{h=1}^{n}\sum_{i=1}^{c} (u_{ih})^m \left\|\Phi(\mathbf{x}_h) - \Phi(\mathbf{v}_i)\right\|^2 \ , \tag{2.68}$$

with the probabilistic constraint over the memberships (Eq. 2.5). The procedure for the optimization of $J^{\Phi}(U,V)$ is again the Picard iteration technique. The minimization of the functional in Eq. 2.68 has been proposed only in the case of a Gaussian kernel $K^{(\mathrm{g})}$. The reason is that the derivative of $J^{\Phi}(U,V)$ with respect to the $\mathbf{v}_i$ using a Gaussian kernel is particularly simple, since it allows to use the kernel trick:

$$\frac{\partial K(\mathbf{x}_h, \mathbf{v}_i)}{\partial \mathbf{v}_i} = \frac{(\mathbf{x}_h - \mathbf{v}_i)}{\sigma^2} K(\mathbf{x}_h, \mathbf{v}_i) \ . \tag{2.69}$$

We obtain for the memberships:

$$u_{ih}^{-1} = \sum_{j=1}^{c} \left(\frac{1 - K(\mathbf{x}_h, \mathbf{v}_i)}{1 - K(\mathbf{x}_h, \mathbf{v}_j)} \right)^{\frac{1}{m-1}} \ , \tag{2.70}$$

and for the codevectors:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} (u_{ih})^m K(x_h, v_i)\mathbf{x}_h}{\sum_{h=1}^{n} (u_{ih})^m K(x_h, v_i)} \ . \tag{2.71}$$

2.2.4.2 FCM I in feature space

Here we derive the Fuzzy c-Means in feature space, which is a clustering method that allows to find a soft linear partitioning of the feature space. This partitioning, back to the input space, results in a soft nonlinear partitioning of data. The functional to optimize [GO98, ZC02] with the probabilistic constraint in Eq. 2.5 is:

$$J^{\Phi}(U, V^{\Phi}) = \sum_{h=1}^{n} \sum_{i=1}^{c} (u_{ih})^m \left\| \Phi(\mathbf{x}_h) - \mathbf{v}_i^{\Phi} \right\|^2 \ . \tag{2.72}$$

It is possible to rewrite explicitly the norm in Eq. 2.72 by using:

$$\mathbf{v}_i^{\Phi} = \frac{\sum_{h=1}^{n} (u_{ih})^m \Phi(\mathbf{x}_h)}{\sum_{h=1}^{n} (u_{ih})^m} = a_i \sum_{h=1}^{n} (u_{ih})^m \Phi(\mathbf{x}_h) \ , \tag{2.73}$$

which is the kernel version of Eq. 2.20. For simplicity of notation we used:

$$a_i^{-1} = \sum_{r=1}^{n} (u_{ir})^m \ . \tag{2.74}$$

Now it is possible to write the kernel version of Eq. 2.18:

$$u_{ih}^{-1} = \sum_{j=1}^{c} \left[\frac{k_{hh} - 2a_i \sum_{r=1}^{n} (u_{ir})^m k_{hr} + a_i^2 \sum_{r=1}^{n} \sum_{s=1}^{n} (u_{ir})^m (u_{is})^m k_{rs}}{k_{hh} - 2a_j \sum_{r=1}^{n} (u_{jr})^m k_{hr} + a_j^2 \sum_{r=1}^{n} \sum_{s=1}^{n} (u_{jr})^m (u_{js})^m k_{rs}} \right]^{\frac{1}{m-1}} \ . \tag{2.75}$$

Eq. 2.75 gives the rule for the update of the memberships.

2.2.4.3 PCM I with the kernelization of the metric

The formulation of the Possibilistic c-Means PCM-I with the kernelization of the metric used in [ZC03] involves the minimization of the following functional:

$$J^{\Phi}(U, V) = \sum_{h=1}^{n} \sum_{i=1}^{c} (u_{ih})^m \left\| \Phi(\mathbf{x}_h) - \Phi(\mathbf{v}_i) \right\|^2 + \sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (1 - u_{ih})^m \tag{2.76}$$

The minimization leads to:

$$u_{ih}^{-1} = 1 + \left(\frac{\|\Phi(\mathbf{x}_h) - \Phi(\mathbf{v}_i)\|^2}{\eta_i} \right)^{\frac{1}{m-1}} , \tag{2.77}$$

that can be rewritten, considering a Gaussian kernel, as:

$$u_{ih}^{-1} = 1 + 2 \left(\frac{1 - K(\mathbf{x}_h, \mathbf{v}_i)}{\eta_i} \right)^{\frac{1}{m-1}} . \tag{2.78}$$

The update of the codevectors follows:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} (u_{ih})^m K(x_h, v_i) \mathbf{x}_h}{\sum_{h=1}^{n} (u_{ih})^m K(x_h, v_i)} . \tag{2.79}$$

The computation of the η_i is straightforward.

2.3 Spectral Clustering and Connections with Kernel Clustering Methods

Spectral clustering methods [CTK01] have a strong connection with graph theory [Chu97, DH73a]. A comparison of some spectral clustering methods has been recently proposed in Ref. [VM05]. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the set of patterns to cluster. Starting from X, we can build a *complete, weighted undirected graph* $G(V, A)$ having a set of nodes $V = \{v_1, \ldots, v_n\}$ corresponding to the n patterns, and edges defined through the $n \times n$ adjacency (also *affinity*) matrix A. The adjacency matrix of a weighted graph is given by the matrix whose element a_{ij} represents the weight of the edge connecting nodes i and j. Being an undirected graph, the property $a_{ij} = a_{ji}$ holds. The adjacency between two patterns can be defined as follows:

$$a_{ij} = \begin{cases} h(\mathbf{x}_i, \mathbf{x}_j) & \text{if } i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{2.80}$$

The function h measures the similarity between patterns and typically a Gaussian function is used:

$$h(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2} \right) , \tag{2.81}$$

where d measures the dissimilarity between patterns and σ controls the rapidity of decay of h. This particular choice has the property that A has only some terms significantly different from 0, i.e., it is sparse.

The degree matrix D is the diagonal matrix whose elements are the degrees of the nodes of G.

$$d_{ii} = \sum_{j=1}^{n} a_{ij} \ . \tag{2.82}$$

In this framework the clustering problem can be seen as a graph cut problem [Chu97] where one wants to separate a set of nodes $S \subset V$ from the complementary set $\bar{S} = V \setminus S$. The graph cut problem can be formulated in several ways, depending on the choice of the function to optimize. One of the most popular functions to optimize is the cut [Chu97]:

$$cut(S, \bar{S}) = \sum_{v_i \in S, v_j \in \bar{S}} a_{ij} \ . \tag{2.83}$$

It is easy to verify that the minimization of this objective function favors partitions containing isolated nodes. To achieve a better balance in the cardinality of S and $\bar{S}$, it is suggested to optimize the normalized cut [SM00]:

$$Ncut(S, \bar{S}) = cut(S, \bar{S}) \left(\frac{1}{assoc(S, V)} + \frac{1}{assoc(\bar{S}, V)} \right) \ , \tag{2.84}$$

where the association $assoc(S, V)$ is also known as the volume of S:

$$assoc(S, V) = \sum_{v_i \in S, v_j \in V} a_{ij} \equiv vol(S) = \sum_{v_i \in S} d_{ii} \ . \tag{2.85}$$

There are other definitions of functions to optimize (e.g., the conductance [KVV00], the normalized association [SM00], ratio cut [DGK05]).

The complexity in optimizing these objective functions is very high (e.g., the optimization of the normalized cut is a NP-hard problem [SM00, WW93]); for this reason it has been proposed to relax it by using spectral concepts of graph analysis. This relaxation can be formulated by introducing the *Laplacian* matrix [Chu97]:

$$L = D - A \ , \tag{2.86}$$

which can be seen as a linear operator on G. In addition to this definition of Laplacian there are alternative definitions:

- Normalized Laplacian $L_{\mathrm{N}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$
- Generalized Laplacian $L_{\mathrm{G}} = D^{-1} L$
- Relaxed Laplacian $L_\rho = L - \rho D$

Each definition is justified by special properties desirable in a given context. The spectral decomposition of the Laplacian matrix can give useful information about the properties of the graph. In particular it can be seen that the second smallest eigenvalue of L is related to the graph cut [Fie73] and the corresponding eigenvector can cluster together similar patterns [BH03, Chu97, SM00].

Spectral approach to clustering has a strong connection with *Laplacian Eigenmaps* [BN03]. The dimensionality reduction problem aims to find a proper low dimensional representation of a data set in a high dimensional space. In [BN03], each node in the graph, which represents a pattern, is connected just with nodes corresponding to neighboring patterns and the spectral decomposition of the Laplacian of the obtained graph permits to find a low dimensional representation of X. The authors point out the close connection with spectral clustering and *Local Linear Embedding* [RS00], providing theoretical and experimental validations.

2.3.1 The graph cut

We now motivate the use of the spectral decomposition of the Laplacian for the clustering problem. We describe the connection between the $cut(S, \bar{S})$ of a graph and the second eigenvalue of L [Chu97].

The Laplacian can be seen as a discrete version of the Laplacian operator $\triangle$, which plays a key role in mathematical physics. The Laplacian L can be seen as an operator on the space of the functions $g : V \to \mathbb{R}$ satisfying[2] [Chu97]:

$$L(g(v_i)) = \sum_{v_j \sim v_i} (g(v_i) - g(v_j))a_{ij} \tag{2.87}$$

Since L is a linear operator, it is possible to use linear algebra to infer important characteristics of G.

There are some basic properties of the eigenvalues and eigenvectors of L [Moh92]. The eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ of L are real since L is symmetric. Recalling that the Rayleight quotient [GVL96] is defined as:

$$\mathcal{R} = \frac{(g, L(g))}{(g, g)} \tag{2.88}$$

it is easy to verify that L is positive semidefinite, since the Rayleight quotient is greater than zero.

$$\frac{(g, L(g))}{(g, g)} = \frac{\sum_{v_j \sim v_i} (g(v_i) - g(v_j))^2 a_{ij}}{\sum_x g^2(x)} \geq 0 \tag{2.89}$$

[2]The sum where $v_j \sim v_i$ means that we have to sum when v_j and v_i are connected.

The smallest eigenvalue of L is $\lambda_1 = 0$ with an associate eigenvector proportional to $(1, 1, \ldots, 1)$ (it follows from the definition of L). Another interesting property associated to the Laplacian is that the multiplicity of the eigenvalues equal to zero is exactly the number of connected components of the graph. This can be verified by applying algebraic concepts [Chu97, Apo67], noting that L assumes a block diagonal structure.

We recall that the cut of G in S and $\bar{S}$ is defined as:

$$cut(S, \bar{S}) = \sum_{v_i \in S, v_j \in \bar{S}} a_{ij} \tag{2.90}$$

The cut measures the loss in removing a set of edges when disconnecting a set of nodes S from its complement. A small value of cut indicates that S and its complement are weakly connected, revealing a strong biclustered structure.

Based on the cut, it is possible to define the Cheeger constant as:

$$h_g = \min_S \frac{cut(S, \bar{S})}{\min(vol(S), vol(\bar{S}))} \tag{2.91}$$

The second smallest eigenvalue λ_2 of L is related to the Cheeger constant by the following relation:

$$2h_g \geq \lambda_2 \geq h_g^2/2 \tag{2.92}$$

Thus, λ_2 of L [Fie73] gives useful information about the connectivity of the graph and on the bounds of the cut. Other theoretical studies on spectral clustering, motivate the use of the decomposition of the Laplacian to cut the graph [BH03, SM00].

Applying simple concepts of algebra, it is possible to show that the second smallest eigenvalue λ_2 and its corresponding eigenvector $\mathbf{e}_2$ are related to the Rayleight quotient [GVL96] by the following:

$$\lambda_2 = \inf_{g \perp (1,1,\ldots,1)} \frac{(g, L(g))}{(g, g)} \tag{2.93}$$

$$\mathbf{e}_2 = \arg \min_{g \perp (1,1,\ldots,1)} \frac{(g, L(g))}{(g, g)} \tag{2.94}$$

This approach can be useful when only λ_2 and $\mathbf{e}_2$ are needed. Indeed, solving the complete eigenproblem can be very expensive, especially for large databases (as in image segmentation problems). In cases where the eigenvectors and eigenvalues of a sparse matrix have to be computed, the Lanczos method [PSL90, GVL96] can be used, speeding up the computing process.

2.3.2 Shi and Malik algorithm

The algorithm proposed by Shi and Malik [SM00] applies the concepts of spectral clustering to image segmentation problems. In this framework, each node is a pixel and the definition of adjacency between them is suitable for image segmentation purposes. In particular, if $\mathbf{x}_i$ is the position of the i-th pixel and $\mathbf{f}_i$ a feature vector which takes into account several of its attributes (e.g., intensity, color and texture information), they define the adjacency as:

$$a_{ij} = \exp\left(-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{2\sigma_1^2}\right) \cdot \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_2^2}\right) & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < R \\ 0 & \text{otherwise.} \end{cases} \tag{2.95}$$

Here R has an influence on how many neighboring pixels can be connected with a pixel, controlling the sparsity of the adjacency and Laplacian matrices. They provide a proof that the minimization of $Ncut(S, \bar{S})$ can be done solving the eigenvalue problem for the normalized Laplacian L_{N}. In summary, the algorithm is composed of these steps:

1. Construct the graph G starting from the data set X calculating the adjacency between patterns using Eq. 2.95
2. Compute the degree matrix D
3. Construct the matrix $L_{\mathrm{N}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$
4. Compute the eigenvector $\mathbf{e}_2$ associated to the second smallest eigenvalue λ_2
5. Use $D^{-\frac{1}{2}}\mathbf{e}_2$ to segment G

In the ideal case of two non connected subgraphs, $D^{-\frac{1}{2}}\mathbf{e}_2$ assumes just two values; this allows to cluster together the components of $D^{-\frac{1}{2}}\mathbf{e}_2$ with the same value. In a real case, the splitting point must be chosen to cluster the components of $D^{-\frac{1}{2}}\mathbf{e}_2$. The authors suggest to use the median value, zero, or the value for which the clustering gives the minimum $Ncut$. The successive partitioning can be made recursively on the obtained sub-graphs, or it is possible to use more than one eigenvector. An interesting approach for clustering simultaneously the data set in more than two clusters can be found in [YS03].

2.3.3 Ng, Jordan, and Weiss algorithm

The algorithm that has been proposed by Ng et al. [NJW02] uses the adjacency matrix A as Laplacian. This definition allows to consider the eigenvector associated with the largest eigenvalues as the "good" one for clustering. This has a computational advantage since the principal eigenvectors can be computed for sparse matrices efficiently using the power

iterations technique. The idea is the same as in other spectral clustering methods, i.e., one finds a new representation of patterns on the first k eigenvectors of the Laplacian of the graph.

The algorithm is composed of these steps:

1. Compute the affinity matrix $A \in \mathbb{R}^{n \times n}$:

$$a_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{2.96}$$

2. Construct the matrix D

3. Compute a normalized version of A, defining this Laplacian:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{2.97}$$

4. Find the k eigenvectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ of L associated to the largest eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$.

5. Form the matrix Z by stacking the k eigenvectors in columns.

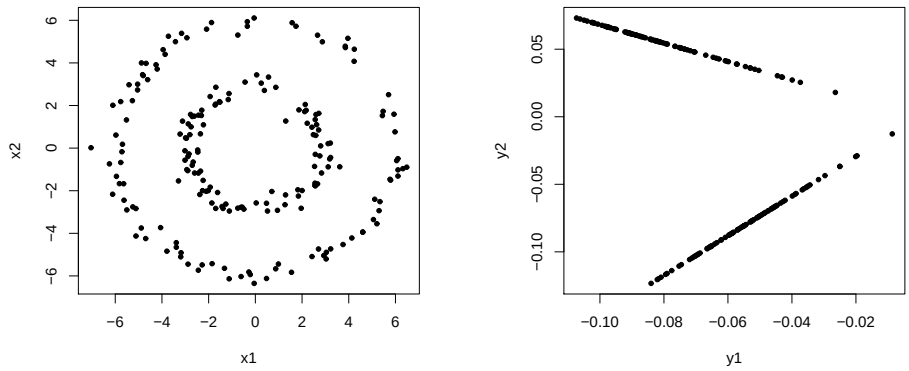
6. Compute the matrix Y by normalizing each of the Z's rows to have unit length:

$$y_{ij} = \frac{z_{ij}}{\sum_{r=1}^{k} z_{ir}^2} \tag{2.98}$$

   In this way all the original points are mapped into a unit hypersphere.

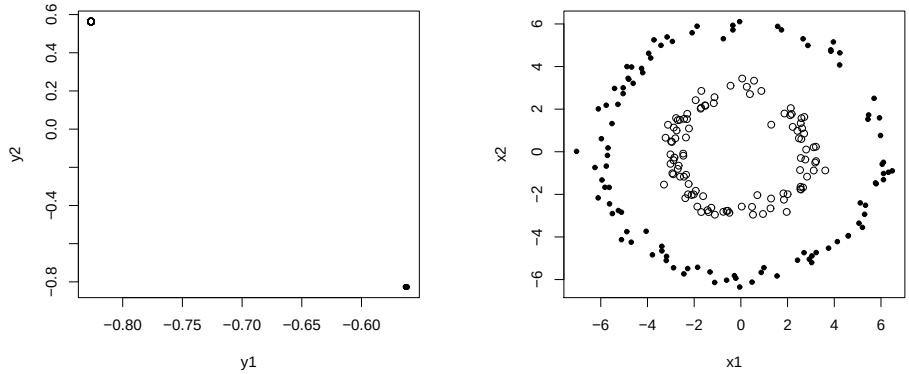
7. In this new representation of the original n patterns, apply a clustering algorithm that attempts to minimize distortion such as K-means.

As a criterion to choose σ, it is suggested to use the value that guarantees the minimum distortion when the clustering stage is performed on Y. This algorithm has been tested on artificial data sets, showing its to separate nonlinear structures. Here we show the steps of the algorithm when applied to the data set in Fig. 2.4a. Once the singular value decomposition of L is computed, we can see the new pattern representations given by the matrices Z and Y in Figs. 2.4b and 2.5a (here obtained with $\sigma = 0.4$). Once Y is computed, it is easy to cluster the two groups of points obtaining the result shown in Fig. 2.5b.



(a) (b)

Figure 2.4: (a) A ring data set. (b) The matrix Z obtained with the first two eigenvectors of the matrix L.



(a) (b)

Figure 2.5: (a) The matrix Y obtained by normalizing the rows of Z clustered by K-means algorithm with two centroids. (b) The result of the Ng-Jordan-Weiss algorithm on the ring data set.

2.3.4 Other Methods

An interesting view of spectral clustering is provided by Meilă et al. [MS00] who describe it in the framework of Markov random walks, leading to a different interpretation of the graph cut problem. It is known, from the theory of Markov random walks, that if we construct the stochastic matrix $P = D^{-1}A$, each element p_{ij} represents the probability of moving from node i to node j. In their work, they provide an explicit connection between the spectral decomposition of L and P, showing that both have the same solution with eigenvalues of P equal to $1 - \lambda_i$, where λ_i are the eigenvalues of L. Moreover, they propose a method to learn a function of the features able to produce a correct segmentation starting from a segmented image.

An interesting study on spectral clustering has been conducted by Kannan et al. [KVV00]. They study spectral clustering objective functions, showing that there is no objective function able to cluster properly every data set. In other words, there always exists some data set for which the optimization of a particular objective function has some drawback. For this reason, they propose a bi-criteria objective function. These two objectives are respectively based on the conductance and the ratio between the auto-association of a subset of nodes S and its volume. Again, the relaxation of this problem is achieved by the decomposition of the Laplacian of the graph associated to the data set.

2.3.5 A Unified View of Spectral and Kernel Clustering Methods

A possible connection between unsupervised kernel algorithms and spectral methods has been recently studied, to find whether these two seemingly different approaches can be described under a more general framework. The hint for this unifying theory lies in the adjacency structure constructed by both these approaches. In the spectral approach there is an adjacency between patterns which is the analogous of the kernel functions in kernel methods.

A direct connection between Kernel PCA and spectral methods has been shown in Refs. [BDLR$^+$04, BVP03]. A unifying view of kernel K-means and spectral clustering methods has been pointed out in Refs. [DGK07, DGK05, DGK04]. In this Section, we report the equivalence between them highlighting that these two approaches have the same foundation; in particular, both can be viewed as a matrix trace maximization problem.

2.3.5.1 Kernel clustering methods objective function

To show the direct equivalence between kernel and spectral clustering methods, we introduce the weighted version of the K-means in kernel induced space [DGK04]. We introduce

a weight matrix W having weights w_k on the diagonal. Recalling that we denote with π_i the i-th cluster, we have that the functional to minimize is the following:

$$J^{\Phi}(W, V^{\Phi}) = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in \pi_i} w_k \left\| \Phi(\mathbf{x}_k) - \mathbf{v}_i^{\Phi} \right\|^2 , \tag{2.99}$$

where:

$$\mathbf{v}_i^{\Phi} = \frac{\sum_{\mathbf{x}_k \in \pi_i} w_k \Phi(\mathbf{x}_k)}{\sum_{\mathbf{x}_k \in \pi_i} w_k} = \frac{\sum_{\mathbf{x}_k \in \pi_i} w_k \Phi(\mathbf{x}_k)}{s_i} , \tag{2.100}$$

with:

$$s_i = \sum_{\mathbf{x}_k \in \pi_i} w_k . \tag{2.101}$$

Let's define the matrix Z having:

$$z_{ki} = \begin{cases} s_i^{-1/2} & \text{if } \mathbf{x}_k \in \pi_i \\ 0 & \text{otherwise.} \end{cases} \tag{2.102}$$

Since the columns of Z are mutually orthogonal, it is easy to verify that:

$$s_i^{-1} = (Z^T Z)_{ii} , \tag{2.103}$$

and that only the diagonal elements are not null.

Now, we denote with F the matrix whose columns are the $\Phi(\mathbf{x}_k)$. It is easy to verify that the matrix FW yields a matrix whose columns are the $w_k \Phi(\mathbf{x}_k)$. Moreover, the expression $FWZZ^T$ gives a matrix having n columns which are the nearest centroids in feature space of the $\Phi(\mathbf{x}_k)$.

Thus, substituting Eq. 2.100 in Eq. 2.99 we obtain the following matrix expression for $J^{\Phi}(W, V^{\Phi})$:

$$J^{\Phi}(W, V^{\Phi}) = \sum_{k=1}^{n} w_k \left\| F_{\cdot k} - (FWZZ^T)_{\cdot k} \right\|^2 \tag{2.104}$$

Here the dot has to be considered as a selection of the k-th column of the matrices. Introducing the matrix $Y = W^{1/2}Z$, which is orthonormal ($Y^TY = I$), the objective function can be rewritten as:

$$\begin{aligned} J^{\Phi}(W, V^{\Phi}) &= \sum_{k=1}^{n} w_k \left\| F_{\cdot k} - (FW^{1/2}YY^TW^{-1/2})_{\cdot k} \right\|^2 \\ &= \left\| FW^{1/2} - FW^{1/2}YY^T \right\|_{\mathrm{F}}^2 \end{aligned} \tag{2.105}$$

where the norm $\|\|_{\mathrm{F}}$ is the Frobenius norm [GVL96]. Using the fact that $\|A\|_{\mathrm{F}} = \mathrm{tr}(AA^T)$ and the properties of the trace, it is possible to see that the minimization of the last equation is equivalent to the maximization of the following [DGK07, DGK05]:

$$J^{\Phi}(W, V^{\Phi}) = \mathrm{tr}(Y^TW^{1/2}F^TFW^{1/2}Y) \tag{2.106}$$

2.3.5.2 Spectral clustering methods objective function

Recalling that the definition of association between two sets of edges S and T of a weighted graph is the following:

$$assoc(S,T) = \sum_{i \in S, j \in T} a_{ij} \tag{2.107}$$

it is possible to define many objective functions to optimize in order to perform clustering. Here, for the sake of simplicity, we consider just the ratio association problem, where one has to maximize:

$$J(S_1, \ldots, S_c) = \sum_{i=1}^{c} \frac{assoc(S_i, S_i)}{|S_i|} \tag{2.108}$$

where $|S_i|$ is the size of the i-th partition. Now we introduce the indicator vector $\mathbf{z}_i$, whose k-th value is zero if $\mathbf{x}_k \notin \pi_i$ and one otherwise. Rewriting the last equation in a matrix form, we obtain the following:

$$J(S_1, \ldots, S_c) = \sum_{i=1}^{c} \frac{\mathbf{z}_i^T A \mathbf{z}_i}{\mathbf{z}_i^T \mathbf{z}_i} \tag{2.109}$$

Normalizing $\mathbf{z}_i$ letting:

$$\mathbf{y}_i = \frac{\mathbf{z}_i}{(\mathbf{z}_i^T \mathbf{z}_i)^{1/2}} \tag{2.110}$$

we obtain:

$$J(S_1, \ldots, S_c) = \sum_{i=1}^{c} \mathbf{y}_i^T A \mathbf{y}_i = \mathrm{tr}(Y^T A Y) \tag{2.111}$$

2.3.5.3 A unified view of the two approaches

Comparing Eq. 2.111 and Eq. 2.106 it is possible to see the perfect equivalence between kernel K-means and the spectral approach to clustering when one wants to maximize the ratio association. To this end, indeed, it is enough to set the weights in the weighted K-means in kernel induced space equal to one, obtaining the classical kernel K-means. It is possible to obtain more general results when one wants to optimize other objective functions in the spectral approach, such as the ratio cut [CSZ93], the normalized cut and the Kernighan-Lin [KL70] objective. For instance, in the case of the minimization of the normalized cut which is one of the most used objective functions, the functional to optimize is:

$$J(S_1, \ldots, S_c) = \mathrm{tr}(Y^T D^{-1/2} A D^{-1/2} Y) \tag{2.112}$$

Thus the correspondence with the objective in the kernel K-means imposes to choose $Y = D^{1/2}Z$, $W = D$ and $K = D^{-1}AD^{-1}$. It is worth noting that for an arbitrary A it

is not guaranteed that $D^{-1}AD^{-1}$ is definite positive. In this case the kernel K-means will not necessarily converge. To cope with this problem in [DGK07] the authors propose to enforce positive definiteness by means of a diagonal shift [RLKB03]:

$$K = \sigma D^{-1} + D^{-1}AD^{-1} \tag{2.113}$$

where σ is a positive coefficient large enough to guarantee the positive definiteness of K.

2.4 Relational Clustering

In this Section, we briefly introduce some basic concepts about crisp and fuzzy relational clustering. In some clustering applications, it is not possible to have a feature-based representation for the patterns, and the description is given in terms of pairwise (dis)similarity relationships among them. Some approaches have been proposed to cluster objects represented in this way. Popular crisp relational clustering algorithms form hierarchical structures agglomerating patterns on the basis of the given dissimilarities; they are the so called Sequential Agglomerative Hierarchical Non-Overlapping SAHN approaches [SS73, JD88, War63]. Agglomerative procedures are bottom-up, since they start by placing each object in its own cluster and gradually merge smaller clusters in larger clusters, until all objects are agglomerated. The basic idea is to define a criterion to compute the distance between clusters, and iteratively merge pairs of clusters having the smallest distance. This scheme is called Johnson's algorithm [JD88]; we can find seven different distance computations [SS73] [JD88] [War63]: Single-link (nearest neighbor), Complete-link (furthest neighbor), Unweighted Pair Group Method using Arithmetic averages (UPGMA), Weighted Pair Group Method using Arithmetic averages (WPGMA), Unweighted Pair Group Method using Centroids (UPGMC), Weighted Pair Group Method using Centroids (WPGMC), and Ward's method (minimum variance). In single-link, the distance between clusters is defined as the distance between the closest pair of objects belonging to them, in complete-link is defined as the distance between the most distant pair of objects. UPGMA and WPGMA assess the dissimilarity between clusters by the average distance between the pairs of objects belonging to the different clusters. The weighted case, gives a different weight to the patterns depending on the size of the cluster. UPGMC and WPGMC compute the dissimilarity between representatives of the clusters to merge. In Ward's method, the union of every possible cluster pair is considered at each step. The two clusters whose fusion results in minimum increase in variance of the new cluster are combined.

Other crisp approaches to the relational clustering are the Partitions Around Medoids (PAM) method [KR90], Clustering LARge Applications (CLARA) [KR90], and Clustering Large Applications based upon RANdomized Search (CLARANS) [NJW02]. PAM method starts with choosing k objects as the initial medoids and assigning objects to the cluster

represented by its medoid. It is possible to compute a score of this configuration, based on a squared error criterion. Randomly selecting a non-medoid object, the total cost of swapping the medoid with it can be computed. If this cost is less than zero, the medoid is swapped with the randomly selected pattern. This procedure can be repeated until no changes are found. PAM has been found to be more robust than k-means in the presence of noise and outliers, since medoids are less influenced by outliers. PAM is computationally inefficient for large data sets; for this reason the modifications CLARA [KR90] and CLARANS [NJW02] have been proposed.

Some fuzzy relational clustering algorithms can be found in literature, for instance those proposed by Ruspini [Rus93], Diday [Did71], Roubens [Rou78], the Relational Fuzzy c-means (RFCM) [HDB89], the Relational Possibilistic c-means (RPCM) [dCOF06], Fuzzy Analysis (FANNY) [KR90], and the Windham association prototypes [Win85]. RFCM is based on the optimization of a proper objective function similar to that of FCM. Also the optimization procedure resembles the one used in FCM. In fact, it turns out to be the relational dual of the FCM; in other words, the RFCM with the Euclidean distances as dissimilarities, gives the FCM. This duality can be found also between the RPCM [dCOF06] and the Possibilistic c-means [KK93]. In general, all the K-means style clustering algorithms are based on the concept of memberships and centroids, and are asked to find the clusters in the input space that is usually Euclidean. In the dual versions, the concept of centroids looses its meaning, since the patterns are not described in terms of features. Moreover, if the dissimilarities are not metric, the convergence of the algorithms is not guaranteed. For this reason, some solutions have been proposed. In Ref. [KJNY01], the authors propose a fuzzy relational algorithm that selects the centroids among the objects composing the data set. FANNY optimizes the same objective function as RFCM with $m = 2$, but employing the Lagrange multiplier technique; this gives an elegant way to handle non-metric dissimilarities. Another approach proposes to transform the dissimilarities between patterns from non-metric to metric [RLBM02, HB94]; this is the basis of the modification allowing NERF c-means to deal with non-metric dissimilarities. In Chapter 4 we will study in more detail the implications of the non metric dissimilarities in relational clustering algorithms based on fuzzy memberships. In particular, we will discuss how to transform the dissimilarities from non-metric to metric, and how these transformations influence the behavior of four relational fuzzy clustering algorithms.

2.4.1 Relational Dual of the FCM I

Let $Y = \{y_1, \ldots, y_n\}$ be a set of n objects and R the $n \times n$ matrix with entries r_{ij} representing the dissimilarity between patterns y_i and y_j. As in the FCM, the membership

matrix U is defined. The objective function to optimize is the following [HDB89]:

$$L(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^m u_{ik}^m r_{hk}}{2\sum_{h=1}^{n} u_{ih}^m} \tag{2.114}$$

subject to the probabilistic constraint in Eq. 2.5. Bezdek proved that minimization of the FCM I and RFCM objective functions are equivalent if R is squared Euclidean. In this case, RFCM can be considered as the relational dual of FCM. In order to derive the necessary update equations for the RFCM, Hathaway and Bezdek proved that the squared Euclidean distance, $d_{ik}^2 = \|\mathbf{x}_j - \mathbf{v}_i\|^2$, from feature vector $\mathbf{x}_j$ to the center of the i^{th} cluster, $\mathbf{v}_i$, can be written in terms of R as follows:

$$d_{ik}^2 = (R\mathbf{v}_i)_k - \mathbf{v}_i^t R\mathbf{v}_i/2 \tag{2.115}$$

where $\mathbf{v}_i$ is the membership vector defined by:

$$\mathbf{v}_i = \frac{(u_{i1}^m, \ldots, u_{in}^m)}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.116}$$

This allows to compute the distances between the objects and the prototypes when only R is given. This means that even when only relational data is available in the form of an $n \times n$ relation matrix, the relational dual of FCM is expected to perform in an equivalent way to FCM, provided that R is Euclidean. In this case, there exists a set of n vectors, called embedding vectors, satisfying:

$$r_{hk} = \|\mathbf{x}_h - \mathbf{x}_k\|^2 \tag{2.117}$$

Resuming, the iteration of the following equations is performed:

$$\mathbf{v}_i = \frac{(u_{i1}^m, \ldots, u_{in}^m)}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.118}$$

$$d_{ik}^2 = (R\mathbf{v}_i)_k - \mathbf{v}_i^t R\mathbf{v}_i/2 \tag{2.119}$$

$$u_{ih}^{-1} = \sum_{j=1}^{c} \left(\frac{d_{ih}^2}{d_{jh}^2} \right)^{\frac{1}{m-1}} \tag{2.120}$$

When R is not metric, it is possible that come d_{ik}^2 are negative. To cope with this this problem, NERF c-means [HB94] transforms R, to turn it to Euclidean. Let $e = \{1, 1, \ldots, 1\}^T$ and I the $n \times n$ identity matrix. The proposed transformation is:

$$R_\beta = R + \beta(ee^T - I) \tag{2.121}$$

where β is a suitable scalar. In NERF c-means, the distances d_{ik}^2 are checked in every iteration for negativity. If they are negative, the transformation in Eq. 2.121 has to be applied with a suitable value of β, to make d_{ik}^2 positive. We will study this transformation in more detail in Chapter 4.

2.4.2 Relational Dual of the PCM I

The objective function of the Relational Dual of the PCM I is called RPCM [dCOF06]. Its objective function is:

$$L(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^m u_{ik}^m r_{hk}}{2\sum_{h=1}^{n} u_{ih}^m} + \sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (1 - u_{ih})^m \tag{2.122}$$

Following the same considerations as in RFCM, we obtain the iteration of the following equations:

$$\mathbf{v}_i = \frac{(u_{i1}^m, \ldots, u_{in}^m)}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.123}$$

$$d_{ik}^2 = (R\mathbf{v}_i)_k - \mathbf{v}_i^t R \mathbf{v}_i / 2 \tag{2.124}$$

$$u_{ih}^{-1} = \left(\frac{d_{ih}^2}{\eta_i}\right)^{\frac{1}{m-1}} + 1 \tag{2.125}$$

In Ref. [dCOF06] the authors suggest a criterion to estimate the values of η_i. Again, it is possible to deal with non-Euclidean matrices R by applying a β-transformation.

Chapter 3

An Experimental Comparison of Kernel and Spectral Clustering Methods

In this Chapter, we compare the performances of some among the algorithms presented in Chapter 2 on several data sets. In particular, we compare the clustering algorithms in feature space, clustering with the kernelization of the metric, Support Vector Clustering, two spectral clustering algorithms, and four standard methods: K-means, FCM I, FCM II, and hierarchical clustering. We decided to include such comparative study in the thesis, since these recent clustering models have not been sufficiently validated in applications by the authors. The data sets considered in the present study are well known in Machine Learning community. All the data sets are labeled. Some of them can be found in the UCI repository [AN07], while two of them are Bioinformatics data sets. We decided to include a variety of data sets differing from cardinality, dimensionality, and number of classes (Tab. 3.1). The comparison is done on the basis of three performance indexes, in particular: misclassifications, normalized mutual information, and conditional entropy.

In the next Sections we briefly describe the data sets, the compared methods, and the performance indexes. Section 3.4 shows the results, and the last Section is devoted to a discussion about them.

3.1 Data Sets

Breast The Breast Cancer Wisconsin (Original) Data Set was obtained by the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [WM90]. The samples

Name	n	d	b
Breast	683	9	2
Colon	62	2000	2
Ecoli	336	7	8
Glass	214	9	6
Haberman	306	3	2
Ionosphere	351	34	2
Iris	150	4	3
Leuk	38	7129	2
Lung	32	54	3
Pima	768	8	2
Sonar	208	60	2
Spambase	4601	57	2
Spectf	80	44	2
Wine	178	13	3

Table 3.1: Resuming table of the characteristics of the studied data sets. n is the number of patterns, d the dimensionality, and b the number of classes.

were analyzed in different moments, since they were received periodically. The data set is composed by 699 nine-dimensional patterns, labeled as benign or malignant. Since there are some missing values, we decided to remove the corresponding patterns, obtaining 683 patterns. The class distribution is 65% for the benign class and 35% for the Malignant class.

Colon The Colon data set by Alon et al. [ABN$^+$99] is an oligonucleotide microarray analysis of gene expression in 40 tumor and 22 normal colon tissue samples. It is used to characterize the role and behavior of more than 6500 human genes in colon adenocarcinoma. The normal samples were obtained from a subset of the tumor samples, so that they are well paired to the corresponding positive samples. The actual data used in the experiments [1], contain only the 2000 most clearly expressed in the experiments, i.e., those with the highest minimal intensity across the 62 tissue samples.

Ecoli Contains the protein localization sites of a E. coli [HN96]. The 336 patterns are described by seven features, and are classified in eight classes. Three of these classes contain less than five patterns.

[1] http://microarray.princeton.edu/oncology/affydata/index.html

Glass This data set contains 214 patterns related to the analysis of types of glasses. The nine features describing each pattern are the refractive index and the concentration of eight chemical elements (Na, Mg, Al, Si, K, Ca, Ba, and Fe). The type of glass can be one among these seven: building windows float processed, building windows non float processed, vehicle windows float processed, vehicle windows non float processed (none in this database), containers, tableware, and headlamps.

Haberman The data set contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. It contains 306 patterns described by the following three features: age of patient at time of operation, patient's year of operation, and number of positive axillary nodes detected. Each pattern is labeled according to the survival of the patient for more than 5 years.

Ionosphere This radar data was collected by a phased array of 16 high-frequency antennas in Goose Bay, Labrador having the free electrons in the ionosphere as target [SWHB89]. The class labels are two: "Good" radar returns are those showing evidence of some type of structure in the ionosphere, while "Bad" returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an appropriate autocorrelation function. The system used 17 pulse numbers and the patterns in the data set are described by 2 features per pulse number.

Iris This is one of the most popular data sets studied by the Machine Learning community [Fis36, DH73b]. The data set contains three classes of 50 patterns each; each class refers to a type of iris plant. One class is linearly separable from the other two that are overlapped. The features are four: sepal length, sepal width, petal length, and petal width.

Leuk The Leukemia data has been provided by Golub et al. [GST+99] [2]. The Leukemia problem consists in characterizing two forms of acute leukemia, Acute Lymphoblastic Leukemia (ALL) and Acute Mieloid Leukemia (AML). The data set contains 38 patterns for which the expression level of 7129 genes has been measured with the DNA microarray technique (the interesting human genes are 6817, and the other are controls required by the technique). Of these samples, 27 are cases of ALL and 11 are cases of AML. Moreover, it is known that the ALL class is composed of two different diseases, since they are originated from different cell lineages (either T-lineage or B-lineage).

[2] http://www.broad.mit.edu/cancer/software/genepattern/datasets/

Lung The data set was published in Ref. [HY91]. It contains 32 54-dimensional patterns that can belong to one out of three types of pathological lung cancers. The Authors give no information about the individual variables.

Pima This data set contains a study on the onset of signs of diabetes in a population living in Phoenix, Arizona, USA. Several constraints were placed on the selection of these patterns from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The data set is composed of 768 patterns described by eight features: Number of times pregnant, Plasma glucose concentration, Diastolic blood pressure, Triceps skin fold thickness, 2-Hour serum insulin, Body mass index, Diabetes pedigree function, and Age.

Sonar Sonar data set is the data set collected and used in Ref. [GS88] on the classification of sonar signals using neural networks. The task is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The data set is contains 198 60-dimensional patterns obtained by measuring the signal at various angles and under various conditions.

Spambase This data set contains 4601 patterns described by 57 features. Each pattern is labeled as spam or non-spam. The features are the frequencies of 48 particular words and 6 particular characters; 3 features are related to the length of uninterrupted sequences of capital letters.

Spectf The data set contains the description of diagnosing of cardiac SPECT images [KCT+01]. The features are extracted from the original SPECT images. Each 44-dimensional pattern is classified into two categories: normal and abnormal. The data set is divided in training and test sets. We decided to use only the 80 patterns of the training set, since they are balanced between the two classes.

Wine This data set contains the results of a chemical analysis of wines grown in the same region in Italy. Such wines are derived from three different cultivars. The analysis determined the quantities of 13 among the constituents found in the three types of wines: Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Non-flavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline. The distribution of the classes is the following: class 1 33.1%, class 2 39.9%, and class 3 27.0%.

3.2 Methods

K-means This is the standard K-means clustering algorithms. The initialization is random, and it is made by selecting the position of the centroids among the patterns to cluster. The only input is the number k of clusters to find[3].

agnes This algorithm is the Sequential Agglomerative Hierarchical Non-Overlapping approach [KR90]. A number of different aggregating procedures are available: single linkage, complete linkage, average linkage, weighted linkage, and Ward (minimum variance).

FCM I and FCM II These two algorithms are the Fuzzy c-means I and Fuzzy c-means II. In FCM I we have to set the number of clusters c and the fuzziness m. In FCM II we have to set the number of clusters c and the fuzziness λ.

FCM I fs and FCM II fs These two methods are respectively the Fuzzy c-means I in feature space and the Fuzzy c-means II in feature space. In both the algorithms, we have to select the number of clusters c and the kernel function along with its parameters. In the following, we will use the Gaussian kernel with standard deviation σ. In FCM I fs we have to set the fuzziness m, while in FCM II fs we have to set the fuzziness λ.

FCM I km and FCM II km These two methods are respectively the Fuzzy c-means I with the kernelization of the metric and the Fuzzy c-means II with the kernelization of the metric. In both the algorithms, we have to select the number of clusters c and the kernel function along with its parameters. In the following, we will use the Gaussian kernel with standard deviation σ. In FCM I fs we have to set the fuzziness m, while in FCM II fs we have to set the fuzziness λ.

SVC This is the Support Vector Clustering algorithm. We have to select the parameter C (or ν) and the kernel function along with its parameters. In the following, we will use the Gaussian kernel with standard deviation σ. We set $C = 1$, in order to avoid outlier rejection that is not handled by the other comparing algorithms. The algorithm will automatically find the number of clusters.

Ng-JW This is the spectral clustering algorithm proposed by Ng. et al. The algorithm requires the selection of the adjacency function along with its parameters. In the following,

[3]We will use c instead of k to use the same notation as in fuzzy clustering algorithms.

we will use the Gaussian function with standard deviation σ. Then we need to select the number of clusters c and the dimension s of the new space, namely the number of eigenvectors used for the new representation.

Shi-Malik This is the spectral clustering algorithm proposed by Shi and Malik. In this algorithm, we have to select the proposed function along with its parameters. In the following we will use the Gaussian function with standard deviation σ. The algorithm splits the data set in two parts according to the eigenvector associated to the second smallest eigenvalue of the Laplacian. The same procedure is applied to the obtained subset and so on, until a value of the normalized cut reaches a fixed threshold that we set to one.

3.3 Performance Indexes

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a labeled data set composed by n patterns. Let's denote by t_i the class labels, belonging to the set of the possible realizations $\mathcal{T} = \{t_1, \ldots, t_b\}$. The class labels can be considered as the realization of a random variable T. Applying a clustering algorithm to the elements of X, we obtain the cluster labels z_i that can be seen as the realization of a random variable Z. Here z_i belongs to the set of possible realizations $\mathcal{Z} = \{z_1, \ldots, z_c\}$. In this context, it is possible to apply some statistical tools to analyze the dependence between these two random variables.

Simple Match A simple choice could be the match between the two realizations. In order to do that, we have to take into account two things: in general c and b are not the same and the sets of labels $\mathcal{T}$ and $\mathcal{Z}$ might be different. For these reasons we need to rearrange the cluster labels in order to match as much as possible the class labels. In other words, we need to transform the cluster label with a function $\pi_k : \mathcal{Z} \to \mathcal{T}$ such that $\pi_k(z_i) = t_j$. In this way we obtain the new cluster labels vector $\{t'_1, \ldots, t'_n\}$. Now it is possible to compute the match between the two label vectors. We will use the misclassification [LRBB04]:

$$\mu = \#\{t'_i \neq t_i\} \tag{3.1}$$

and the accuracy:

$$\psi = \#\{t'_i = t_i\}/n \tag{3.2}$$

Among all the permutations π_k, we select the one leading to the minimum value of μ.

Preliminary definitions for entropy based scores Let's define the confusion matrix:

		Cluster Labels			
		z_1	z_2	$\cdots$	z_c
Class Labels	t_1	a_{11}	a_{12}	$\cdots$	a_{1c}
	t_2	a_{21}	a_{22}	$\cdots$	a_{2c}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	t_b	a_{b1}	a_{b2}	$\cdots$	a_{bc}

Each entry a_{ij} of the confusion matrix contains the number of times that the clustering algorithm assigned the cluster label z_j to the pattern $\mathbf{x}_i$ having class labels t_i. On the basis of the confusion matrix, the following probabilities can be defined:

- $p(t_i) = \frac{|t_i|}{n} = \frac{\sum_r a_{ir}}{n}$
- $p(z_j) = \frac{|z_j|}{n} = \frac{\sum_r a_{rj}}{n}$
- $p(t_i, z_j) = \frac{a_{ij}}{n}$

We recall the definition of the entropy; for the random variables T and Z it reads:

$$H(T) = \sum_i p(t_i) \log\left(p(t_i)\right) \tag{3.3}$$

$$H(Z) = \sum_j p(z_j) \log\left(p(z_j)\right) \tag{3.4}$$

The joint entropy of T and Z is:

$$H(T, Z) = -\sum_{ij} p(t_i, z_j) \log\left(p(t_i, z_j)\right) \tag{3.5}$$

The two entropy based scores that we will use to assess the goodness of the clustering results are the Conditional Entropy $H(T|Z)$ and the Normalized Mutual Information $I_{\mathrm{N}}(T, Z)$.

Conditional Entropy The Conditional Entropy $H(T|Z)$ is a measure of the uncertainty of a random variable T given the value of the random variable Z [FB03]. It can be formalized in the following way:

$$H(T|Z) = \sum_j p(z_j) H(T|Z = z_j) = -\sum_j p(z_j) \sum_i p(t_i|z_j) \log\left(p(t_i|z_j)\right) \tag{3.6}$$

Applying some transformations, it is possible to rewrite the Conditional Entropy:

$$H(T|Z) = H(T, Z) - H(Z) = -\sum_{ij} p(t_i, z_j) \log\left(p(t_i, z_j)\right) + \sum_j p(z_j) \log\left(p(z_j)\right) \tag{3.7}$$

Intuitively, if the two random variables are identical, knowing the realization of Z gives no uncertainty about T, leading to a null conditional entropy. On the contrary, if the two random variables are independent, there is still uncertainty in the value of T given Z. Formally, in the dependent case, $p(t_i|z_j) = 1$ leading to $H(T|Z) = 0$. In the independent case, $p(t_i|z_j) = p(t_i)$ leading to $H(T|Z) = H(T)$. The Conditional Entropy is zero when each cluster found contains pattern from a single class. This can be useful to check the purity of the cluster labels Z with respect to the class labels T. On the other hand, the method is biased when the number of clusters c is very large. In the extreme case when we assign one pattern per cluster, the Conditional Entropy results $H(T|Z) = 0$.

Normalized Mutual Information The mutual information between two discrete random variables T and Z is [FB03]:

$$I(T, Z) = \sum_{ij} p(t_i, z_j) \log \left(\frac{p(t_i, z_j)}{p(t_i)p(z_j)} \right) \tag{3.8}$$

The mutual information measures the information shared by two discrete random variables: it measures how much knowing one of these variables reduces our uncertainty about the other. Intuitively, if the two random variables are independent, knowing the realization of one of them does not give any information about the other and viceversa; their mutual information is zero. If the two random variables are identical, the realization of one of them determines the value of the other and viceversa. As a result, the mutual information is the same as the uncertainty contained in either one of the random variables, that is their entropy. Formally, if they are uncorrelated, it is possible to factorize the joint probability $p(t_i, z_j) = p(t_i)p(z_j)$ leading to $I(T, Z) = 0$. If they are identical, $I(T, Z)$ reduces to the entropy $H(T) = H(Z)$, since $p(x, y) = p(x) = p(y)$. These considerations show that the mutual information is dependent on the data set; in other words, the upper bound is not independent from the considered problem. It is possible to normalize $I(T, Z)$ in the interval $[0, 1]$ using the following [FB03]:

$$I_{\mathrm{N}}(T, Z) = \frac{I(T, Z)}{\sqrt{H(T)H(Z)}} \tag{3.9}$$

In this way, a value of $I_{\mathrm{N}}(T, Z)$ near one means high correlation between cluster and class labels, a value near zero means independence.

3.4 Results

The methods presented in Section 3.2 have been tested on the data sets described in Section 3.1. The number of classes can give some guidelines on the selection of the number

of clusters. It is worth noting, however, that in general the number of clusters and the number of classes might be not related to each other. A typical example is the Iris data set, where the two overlapped classes are very likely to be identified as one cluster by a clustering algorithm. In other situations, it is possible to use some prior information about the number of clusters, as we did for Leuk data set, since we know that the patients belong to three different groups. In the context of central clustering, estimating the number of clusters is a very important issue, and there is a vast literature on methods designed for it. Even though, we prefer to use the class labels as our prior knowledge about the number of clusters, thus avoiding a further model selection step.

To perform a fair comparison among the methods, we used the same number of clusters for all of them. Some algorithms find the natural number of clusters given a particular set of parameters. In this case, we set the parameters in order to have a selection of the wanted number of clusters by the algorithm. We tested the methods varying all the parameters in a wide range; we report the results for the selection giving the best performances. "Varying all the parameters in a wide range" means that we tried a set of values for all the parameters. For example, in FCM I fs we tried the values of σ in the interval $[0.8, 9.6]$ with steps of 0.8 and m in the interval $[1, 2.6]$ with steps of 0.1, checking all the combinations of such parameters. For the algorithms starting with a random initialization, the results are averaged over 20 runs; in Tabs. 3.2, 3.3, 3.4, 3.5, and 3.6 each score is reported along with its standard deviation.

3.5 Discussions

From Tabs. 3.2, 3.3, 3.4, 3.5, and 3.6 reporting the results, it is possible to see that there are no methods that perform better or worse than the others in general.

For clustering methods using kernels, in general, we can see that the methods in feature space perform better than methods with the kernelization of the metric. Clustering with the kernelization of the metric, in some situations give very poor results, especially when the number of clusters is very high. SVC has been used only with $C = 1$, i.e., without the rejection of the outliers. This fact affected the results that are not very good in general. On the other hand, this choice was necessary to compare its results with the other methods that do not handle an outlier class.

The spectral algorithm proposed by Ng et al. achieves good results in almost all the data sets. This means that a low dimensional representation, based on the eigenvectors of the Laplacian of the graph obtained on the data, is quite effective to highlight structures in data. On the other hand, it requires the tune of three parameters, namely the adjacency function parameter σ, the number of clusters c, and the dimensions of the new representation s. It is surprising that the other spectral algorithm, the one by Shi and Malik, gives

Breast				
Method	Parameters	$\psi \quad (\mu)$	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=3$, $\sigma=7.2$, $m=1.2$	0.972, 0.003 (18.9, 2.2)	0.702, 0.039	0.103, 0.014
FCM II fs	$c=2$, $\sigma=8$, $\lambda=0.35$	0.972, 0.000 (19.0, 0.0)	0.814, 0.000	0.116, 0.000
FCM I km	$c=2$, $\sigma=0.1$, $m=1.2$	0.653, 0.000 (237.0, 0.0)	0.009, 0.000	0.646, 0.000
FCM II km	$c=2$, $\sigma=0.01$, $\lambda=0.02$	0.652, 0.000 (238.0, 0.0)	0.007, 0.000	0.646, 0.000
SVC	$c=3$, $C=1$, $\sigma=3.75$	0.652, 0.000 (238.0, 0.0)	0.018, 0.000	0.646, 0.000
FCM I	$c=2$, $m=1.2$	0.960, 0.000 (27.0, 0.0)	0.748, 0.000	0.166, 0.000
FCM II	$c=2$, $\lambda=400$	0.972, 0.000 (19.0, 0.2)	0.812, 0.002	0.118, 0.001
Ng-JW	$c=2$, $\sigma=7.2$, $s=3$	0.977, 0.000 (16.0, 0.0)	0.836, 0.000	0.104, 0.000
Shi-Malik	$c=2$, $\sigma=3$	0.958, 0.000 (29.0, 0.0)	0.741, 0.000	0.174, 0.000
agnes	ward, $c=2$	0.969, 0.000 (21.0, 0.0)	0.815, 0.000	0.113, 0.000
K-means	$c=2$	0.960, 0.000 (27.0, 0.0)	0.748, 0.000	0.166, 0.000
Colon				
Method	Parameters	$\psi \quad (\mu)$	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=10$, $m=2.6$	0.659, 0.061 (21.2, 3.8)	0.041, 0.128	0.623, 0.084
FCM II fs	$c=2$, $\sigma=7$, $\lambda=0.005$	0.656, 0.032 (21.4, 2.0)	0.055, 0.056	0.614, 0.038
FCM I km	$c=2$, $\sigma=10$, $m=2$	0.661, 0.000 (21.0, 0.0)	0.026, 0.000	0.639, 0.000
FCM II km	$c=2$, $\sigma=10$, $\lambda=0.1$	0.661, 0.000 (21.0, 0.0)	0.026, 0.000	0.639, 0.000
SVC	$c=2$, $C=1$, $\sigma=10$	0.645, 0.000 (22.0, 0.0)	0.031, 0.000	0.643, 0.000
FCM I	$c=2$, $m=1.4$	0.645, 0.000 (22.0, 0.0)	0.046, 0.000	0.622, 0.000
FCM II	$c=2$, $\lambda=10$	0.652, 0.032 (21.6, 2.0)	0.052, 0.041	0.619, 0.026
Ng-JW	$c=2$, $\sigma=80$, $s=5$	0.770, 0.097 (14.3, 6.0)	0.241, 0.162	0.493, 0.106
Shi-Malik	$c=2$, $\sigma=10$	0.645, 0.000 (22.0, 0.0)	0.087, 0.000	0.613, 0.000
agnes	average, $c=2$	0.645, 0.000 (22.0, 0.0)	0.087, 0.000	0.613, 0.000
K-means	$c=2$	0.645, 0.000 (22.0, 0.0)	0.041, 0.014	0.625, 0.009
Ecoli				
Method	Parameters	$\psi \quad (\mu)$	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=7$, $\sigma=0.6$, $m=1.6$	0.732, 0.001 (90.0, 0.2)	0.459, 0.001	0.731, 0.002
FCM II fs	$c=7$, $\sigma=0.8$, $\lambda=0.09$	0.727, 0.009 (91.8, 2.9)	0.455, 0.012	0.739, 0.022
FCM I km	$c=7$, $\sigma=0.1$, $m=1.2$	0.446, 0.000 (186.0, 0.0)	0.046, 0.000	1.446, 0.000
FCM II km	$c=7$, $\sigma=0.1$, $\lambda=0.002$	0.443, 0.000 (187.0, 0.0)	0.045, 0.000	1.448, 0.000
SVC	$c=7$, $C=1$, $\sigma=0.22$	0.446, 0.000 (186.0, 0.0)	0.148, 0.000	1.450, 0.000
FCM I	$c=7$, $m=1.6$	0.724, 0.001 (92.8, 0.4)	0.458, 0.004	0.738, 0.007
FCM II	$c=7$, $\lambda=0.06$	0.720, 0.009 (94.1, 3.1)	0.453, 0.015	0.746, 0.025
Ng-JW	$c=7$, $\sigma=60$, $s=5$	0.798, 0.010 (68.0, 3.3)	0.532, 0.012	0.609, 0.019
Shi-Malik	$c=7$, $\sigma=0.19$	0.738, 0.000 (88.0, 0.0)	0.581, 0.000	0.694, 0.000
agnes	ward, $c=7$	0.682, 0.000 (107.0, 0.0)	0.419, 0.000	0.806, 0.000
K-means	$c=7$	0.705, 0.016 (99.0, 5.4)	0.429, 0.024	0.790, 0.047

Table 3.2: Clustering results on Breast, Colon, and Ecoli data sets

Glass				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=6$, $\sigma=1$, $m=1.4$	0.623, 0.019 (80.8, 4.1)	0.408, 0.006	0.856, 0.013
FCM II fs	$c=6$, $\sigma=0.8$, $\lambda=0.2$	0.624, 0.010 (80.5, 2.2)	0.381, 0.012	0.898, 0.018
FCM I km	$c=6$, $\sigma=2$, $m=1.2$	0.463, 0.000 (115.0, 0.0)	0.074, 0.000	1.391, 0.000
FCM II km	$c=6$, $\sigma=10$, $\lambda=0.001$	0.393, 0.000 (130.0, 0.0)	0.039, 0.000	1.451, 0.000
SVC	$c=6$, $C=1$, $\sigma=1.3$	0.379, 0.000 (133.0, 0.0)	0.129, 0.000	1.443, 0.000
FCM I	$c=6$, $m=1.8$	0.610, 0.002 (83.4, 0.5)	0.363, 0.001	0.946, 0.0009
FCM II	$c=6$, $\lambda=1.2$	0.614, 0.038 (82.5, 8.2)	0.343, 0.027	0.976, 0.0349
Ng-JW	$c=6$, $\sigma=4$, $s=3$	0.614, 0.034 (82.7, 7.4)	0.398, 0.015	0.904, 0.0224
Shi-Malik	$c=6$, $\sigma=0.09$	0.542, 0.000 (98.0, 0.0)	0.340, 0.000	1.132, 0.000
agnes	ward, $c=6$	0.542, 0.000 (98.0, 0.0)	0.397, 0.000	0.970, 0.000
K-means	$c=6$	0.571, 0.015 (91.7, 3.2)	0.404, 0.022	0.948, 0.026
Haberman				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=0.9$, $m=1.2$	0.735, 0.000 (81.0, 0.0)	0.009, 0.009	0.572, 0.005
FCM II fs	$c=2$, $\sigma=2$, $\lambda=0.036$	0.735, 0.000 (81.0, 0.0)	0.019, 0.010	0.567, 0.006
FCM I km	$c=2$, $\sigma=3$, $m=1.2$	0.735, 0.000 (81.0, 0.0)	0.020, 0.000	0.574, 0.000
FCM II km	$c=2$, $\sigma=0.5$, $\lambda=0.1$	0.735, 0.000 (81.0, 0.0)	0.009, 0.000	0.577, 0.000
SVC	$c=3$, $C=1$, $\sigma=5.6$	0.739, 0.000 (80.0, 0.0)	0.034, 0.000	0.573, 0.000
FCM I	$c=2$, $m=1.8$	0.735, 0.000 (81.0, 0.0)	0.000, 0.000	0.578, 0.000
FCM II	$c=2$, $\lambda=10$	0.735, 0.000 (81.0, 0.0)	0.006, 0.000	0.578, 0.000
Ng-JW	$c=2$, $\sigma=1.6$, $s=3$	0.771, 0.000 (70.0, 0.0)	0.100, 0.000	0.530, 0.000
Shi-Malik	$c=2$, $\sigma=0.7$	0.745, 0.000 (78.0, 0.0)	0.074, 0.000	0.565, 0.000
agnes	complete, $c=2$	0.745, 0.000 (78.0, 0.0)	0.042, 0.000	0.561, 0.000
K-means	$c=2$	0.735, 0.000 (81.0, 0.0)	0.001, 0.000	0.578, 0.000
Ionosphere				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=4$, $\sigma=3$, $m=1.2$	0.846, 0.001 (54.2, 0.5)	0.275, 0.022	0.406, 0.021
FCM II fs	$c=4$, $\sigma=4$, $\lambda=0.03$	0.874, 0.026 (44.2, 9.1)	0.343, 0.048	0.354, 0.041
FCM I km	$c=4$, $\sigma=5$, $m=1.2$	0.846, 0.000 (54.0, 0.0)	0.260, 0.000	0.424, 0.000
FCM II km	$c=4$, $\sigma=4$, $\lambda=0.06$	0.837, 0.021 (57.3, 7.5)	0.247, 0.040	0.434, 0.032
SVC	$c=4$, $C=1$, $\sigma=1.75$	0.650, 0.000 (123.0, 0.0)	0.045, 0.000	0.644, 0.000
FCM I	$c=4$, $m=1.2$	0.838, 0.000 (57.0, 0.0)	0.244, 0.000	0.438, 0.000
FCM II	$c=4$, $\lambda=1$	0.836, 0.016 (57.7, 5.5)	0.244, 0.027	0.437, 0.023
Ng-JW	$c=4$, $\sigma=0.7$, $s=9$	0.873, 0.002 (44.7, 0.7)	0.320, 0.004	0.366, 0.005
Shi-Malik	$c=4$, $\sigma=1.3$	0.838, 0.000 (57.0, 0.0)	0.265, 0.000	0.431, 0.000
agnes	ward, $c=4$	0.875, 0.000 (44.0, 0.0)	0.354, 0.000	0.354, 0.000
K-means	$c=4$	0.822, 0.033 (62.6, 11.4)	0.227, 0.047	0.452, 0.040

Table 3.3: Clustering results on Glass, Haberman, and Ionosphere data sets.

Iris				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=3$, $\sigma=0.6$, $m=1.2$	0.947, 0.000 (8.0, 0.0)	0.845, 0.000	0.172, 0.000
FCM II fs	$c=3$, $\sigma=0.6$, $\lambda=0.1$	0.923, 0.017 (11.5, 2.6)	0.810, 0.024	0.214, 0.029
FCM I km	$c=3$, $\sigma=3$, $m=2.4$	0.907, 0.000 (14.0, 0.0)	0.766, 0.000	0.260, 0.000
FCM II km	$c=3$, $\sigma=5$, $\lambda=0.2$	0.913, 0.000 (13.0, 0.0)	0.745, 0.000	0.283, 0.000
SVC	$c=3$, $C=1$, $\sigma=0.35$	0.680, 0.000 (48.0, 0.0)	0.736, 0.000	0.453, 0.000
FCM I	$c=3$, $m=2.4$	0.900, 0.000 (15.0, 0.0)	0.758, 0.000	0.270, 0.000
FCM II	$c=3$, $\lambda=5.4$	0.913, 0.000 (13.0, 0.0)	0.745, 0.000	0.283, 0.000
Ng-JW	$c=3$, $\sigma=2.4$, $s=5$	0.942, 0.044 (8.7, 6.6)	0.833, 0.091	0.184, 0.101
Shi-Malik	$c=3$, $\sigma=0.7$	0.900, 0.000 (15.0, 0.0)	0.798, 0.000	0.234, 0.000
agnes	average, $c=3$	0.907, 0.000 (14.0, 0.0)	0.806, 0.000	0.224, 0.000
K-means	$c=3$	0.860, 0.083 (21.1, 12.5)	0.733, 0.061	0.309, 0.087
Leuk				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=3$, $\sigma=1000$, $m=1.2$	0.921, 0.000 (3.0, 0.0)	0.459, 0.000	0.247, 0.000
FCM II fs	$c=3$, $\sigma=1000$, $\lambda=0.07$	0.963, 0.013 (1.4, 0.5)	0.590, 0.053	0.127, 0.042
FCM I km	$c=3$, $\sigma=800$, $m=1.2$	0.942, 0.024 (2.2, 0.9)	0.522, 0.039	0.202, 0.037
FCM II km	$c=3$, $\sigma=900$, $\lambda=0.1$	0.946, 0.061 (2.1, 2.3)	0.561, 0.120	0.158, 0.102
SVC	$c=3$, $C=1$, $\sigma=360$	0.711, 0.000 (11.0, 0.0)	0.049, 0.000	0.583, 0.000
FCM I	$c=3$, $m=1.2$	0.895, 0.000 (4.0, 0.0)	0.383, 0.018	0.310, 0.004
FCM II	$c=3$, $\lambda=60000$	0.897, 0.067 (3.9, 2.5)	0.442, 0.134	0.259, 0.114
Ng-JW	$c=3$, $\sigma=1100$, $s=3$	0.947, 0.000 (2.0, 0.0)	0.516, 0.000	0.183, 0.000
Shi-Malik	$c=2$, $\sigma=420$	0.816, 0.000 (7.0, 0.0)	0.326, 0.000	0.455, 0.000
agnes	ward, $c=3$	0.816, 0.000 (7.0, 0.0)	0.278, 0.000	0.414, 0.000
K-means	$c=3$	0.883, 0.088 (4.5, 3.3)	0.424, 0.201	0.282, 0.158
Lung				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=3$, $\sigma=4$, $m=1.2$	0.563, 0.000 (14.0, 0.0)	0.300, 0.000	0.760, 0.000
FCM II fs	$c=3$, $\sigma=6$, $\lambda=0.1$	0.581, 0.029 (13.4, 0.9)	0.290, 0.028	0.777, 0.024
FCM I km	$c=3$, $\sigma=70$, $m=2$	0.553, 0.035 (14.3, 1.1)	0.293, 0.048	0.788, 0.054
FCM II km	$c=3$, $\sigma=10$, $\lambda=0.06$	0.603, 0.015 (12.7, 0.5)	0.328, 0.005	0.754, 0.009
SVC	$c=4$, $C=1$, $\sigma=1.9$	0.500, 0.000 (16.0, 0.0)	0.173, 0.000	0.970, 0.000
FCM I	$c=3$, $m=2.2$	0.548, 0.030 (14.5, 0.9)	0.285, 0.061	0.790, 0.065
FCM II	$c=3$, $\lambda=5$	0.633, 0.042 (11.8, 1.3)	0.363, 0.000	0.707, 0.011
Ng-JW	$c=3$, $\sigma=3$, $s=2$	0.681, 0.055 (10.2, 1.8)	0.369, 0.049	0.688, 0.051
Shi-Malik	$c=3$, $\sigma=1.8$	0.688, 0.000 (10.0, 0.0)	0.366, 0.000	0.731, 0.000
agnes	ward, $c=3$	0.594, 0.000 (13.0, 0.0)	0.336, 0.000	0.770, 0.000
K-means	$c=3$	0.538, 0.024 (14.8, 0.8)	0.279, 0.055	0.796, 0.055

Table 3.4: Clustering results on Iris, Leuk, and Lung data sets.

Pima				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=500$, $m=1.4$	0.660, 0.000 (261, 0)	0.035, 0.000	0.626, 0.000
FCM II fs	$c=2$, $\sigma=900$, $\lambda=0.2$	0.654, 0.012 (266, 9)	0.016, 0.023	0.636, 0.015
FCM I km	$c=2$, $\sigma=10$, $m=1.2$	0.652, 0.000 (267, 0)	0.005, 0.000	0.646, 0.000
FCM II km	$c=2$, $\sigma=10$, $\lambda=0.1$	0.652, 0.000 (267, 0)	0.006, 0.000	0.646, 0.000
SVC	$c=3$, $C=1$, $\sigma=40$	0.652, 0.000 (267, 0)	0.018, 0.000	0.644, 0.000
FCM I	$c=2$, $m=1.2$	0.660, 0.000 (261, 0)	0.030, 0.000	0.630, 0.000
FCM II	$c=2$, $\lambda=1000$	0.660, 0.000 (261, 0)	0.030, 0.000	0.630, 0.000
Ng-JW	$c=2$, $\sigma=8$, $s=2$	0.659, 0.000 (262, 0)	0.017, 0.000	0.642, 0.000
Shi-Malik	$c=2$, $\sigma=30$	0.659, 0.000 (262, 0)	0.019, 0.000	0.642, 0.000
agnes	ward, $c=2$	0.676, 0.000 (249, 0)	0.046, 0.000	0.619, 0.000
K-means	$c=2$	0.660, 0.000 (261, 0)	0.030, 0.000	0.630, 0.000
Sonar				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=40$, $m=1.2$	0.563, 0.000 (91.0, 0.0)	0.012, 0.000	0.682, 0.000
FCM II fs	$c=2$, $\sigma=100$, $\lambda=0.02$	0.567, 0.021 (90.1, 4.3)	0.014, 0.010	0.681, 0.007
FCM I km	$c=2$, $\sigma=50$, $m=1.2$	0.563, 0.000 (91.0, 0.0)	0.012, 0.000	0.682, 0.000
FCM II km	$c=2$, $\sigma=20$, $\lambda=0.07$	0.563, 0.022 (91.0, 4.5)	0.012, 0.009	0.682, 0.006
SVC	$c=3$, $C=1$, $\sigma=0.5$	0.538, 0.000 (96.0, 0.0)	0.066, 0.000	0.669, 0.000
FCM I	$c=2$, $m=1.2$	0.563, 0.000 (91.0, 0.0)	0.012, 0.000	0.682, 0.000
FCM II	$c=2$, $\lambda=1$	0.558, 0.000 (92.0, 0.0)	0.010, 0.000	0.684, 0.000
Ng-JW	$c=2$, $\sigma=4$, $s=7$	0.601, 0.037 (82.9, 7.6)	0.034, 0.023	0.667, 0.016
Shi-Malik	$c=2$, $\sigma=0.5$	0.563, 0.000 (91.0, 0.0)	0.075, 0.000	0.668, 0.000
agnes	average, $c=2$	0.553, 0.000 (93.0, 0.0)	0.009, 0.000	0.687, 0.000
K-means	$c=2$	0.553, 0.000 (93.0, 0.0)	0.009, 0.000	0.685, 0.000
Spambase				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=125$, $m=1.4$	0.697, 0.000 (1394, 0)	0.095, 0.000	0.608, 0.000
FCM II fs	$c=2$, $\sigma=75$, $\lambda=0.1$	0.697, 0.000 (1396, 0)	0.113, 0.000	0.594, 0.000
FCM I km	$c=2$, $\sigma=125$, $m=1.8$	0.700, 0.000 (1380, 0)	0.105, 0.000	0.610, 0.000
FCM II km	$c=2$, $\sigma=125$, $\lambda=0.2$	0.684, 0.000 (1453, 0)	0.088, 0.000	0.622, 0.000
FCM I	$c=2$, $m=2.6$	0.647, 0.000 (1622, 0)	0.055, 0.000	0.647, 0.000
FCM II	$c=2$, $\lambda=710^6$	0.679, 0.000 (1476, 1)	0.072, 0.000	0.627, 0.000
Ng-JW	$c=2$, $\sigma=3$, $s=10$	0.685, 0.000 (1451, 0)	0.082, 0.000	0.623, 0.000
Shi-Malik	$c=2$, $\sigma=7$	0.611, 0.000 (1789, 0)	0.033, 0.000	0.663, 0.000
agnes	ward, $c=2$	0.662, 0.000 (1557, 0)	0.064, 0.000	0.639, 0.000
K-means	$c=2$	0.636, 0.000 (1675, 0)	0.047, 0.000	0.653, 0.000

Table 3.5: Clustering results on Pima, Sonar, and Spambase data sets.

Spectf				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=2$, $\sigma=40$, $m=1.4$	0.806, 0.016 (15.6, 1.3)	0.324, 0.029	0.473, 0.020
FCM II fs	$c=2$, $\sigma=30$, $\lambda=0.1$	0.800, 0.000 (16.0, 0.0)	0.283, 0.000	0.498, 0.000
FCM I km	$c=2$, $\sigma=1000$, $m=1.8$	0.725, 0.000 (22.0, 0.0)	0.202, 0.000	0.561, 0.000
FCM II km	$c=2$, $\sigma=300$, $\lambda=0.08$	0.726, 0.012 (22.0, 0.9)	0.203, 0.017	0.561, 0.012
SVC	$c=2$, $C=1$, $\sigma=50$	0.513, 0.000 (39.0, 0.0)	0.041, 0.000	0.684, 0.000
FCM I	$c=2$, $m=2$	0.725, 0.000 (22.0, 0.0)	0.202, 0.000	0.561, 0.000
FCM II	$c=2$, $\lambda=1200$	0.725, 0.000 (22.0, 0.0)	0.202, 0.000	0.561, 0.000
Ng-JW	$c=2$, $\sigma=7$, $s=2$	0.738, 0.000 (21.0, 0.0)	0.204, 0.000	0.557, 0.000
Shi-Malik	$c=2$, $\sigma=30$	0.600, 0.000 (32.0, 0.0)	0.158, 0.000	0.618, 0.000
agnes	ward, $c=2$	0.700, 0.000 (24.0, 0.0)	0.220, 0.000	0.559, 0.000
K-means	$c=2$	0.675, 0.000 (26.0, 0.0)	0.247, 0.000	0.553, 0.000
wine				
Method	Parameters	ψ (μ)	$I_{\mathrm{N}}(T,Z)$	$H(T\|Z)$
FCM I fs	$c=3$, $\sigma=77$, $m=1.2$	0.730, 0.000 (48.0, 0.0)	0.405, 0.000	0.645, 0.000
FCM II fs	$c=3$, $\sigma=80$, $\lambda=0.15$	0.730, 0.000 (48.0, 0.0)	0.405, 0.000	0.645, 0.000
FCM I km	$c=3$, $\sigma=120$, $m=1.4$	0.730, 0.000 (48.0, 0.0)	0.423, 0.000	0.625, 0.000
FCM II km	$c=3$, $\sigma=130$, $\lambda=0.5$	0.730, 0.000 (48.0, 0.0)	0.423, 0.000	0.625, 0.000
SVC	$c=3$, $C=1$, $\sigma=50$	0.433, 0.000 (101.0, 0.0)	0.091, 0.000	1.048, 0.000
FCM I	$c=3$, $m=1.6$	0.697, 0.000 (54.0, 0.0)	0.421, 0.000	0.629, 0.000
FCM II	$c=3$, $\lambda=80000$	0.724, 0.000 (49.0, 0.0)	0.412, 0.000	0.637, 0.000
Ng-JW	$c=3$, $\sigma=4.8$, $s=3$	0.696, 0.023 (54.2, 4.0)	0.386, 0.008	0.697, 0.030
Shi-Malik	$c=4$, $\sigma=100$	0.708, 0.000 (52.0, 0.0)	0.402, 0.000	0.629, 0.000
agnes	ward, $c=3$	0.697, 0.000 (54.0, 0.0)	0.416, 0.000	0.634, 0.000
K-means	$c=3$	0.697, 0.011 (54.0, 2.1)	0.425, 0.008	0.629, 0.018

Table 3.6: Clustering results on Spectf and Wine data sets.

worse results in general. One possible explanation, in the case of more than two clusters, lies in the selection of adjacency function parameter σ, that is not tuned for the recursive splits. Once data are split in two subsets, the average distance of patterns is different, requiring the use of a different value of σ. The tune of the value of σ even for the subsets, would have required a computationally intense model selection. For two clusters problems, the analysis of the results shows that the heuristic of selecting the split on the basis of the minimum of the normalized cut is not good in general.

An important result, that is clear from the experimental validation, is that clustering in kernel induced spaces and spectral clustering outperform standard clustering algorithms. This is one of the motivations that support the interest of the Machine Learning community for these recent clustering techniques. On the other hand, the methods based on kernels and spectral clustering methods require the tune of the kernel or the adjacency function. In many applications, we found that the values of the standard deviation of such functions lead to good performances only in a narrow interval.

It is possible to make some consideration on the spatial and temporal complexity of such methods with respect to the size of the data set, its dimensionality, and the number of clusters. In particular, the complexity of a single iteration central clustering algorithms in kernel induced space is quadratic with the cardinality of the data set, while it is linear for standard central clustering algorithms. For all these iterative algorithms, we cannot take into account the number of iterations that can have a strong impact on the running time of the algorithms. Their convergence depends on the particular data set and the choice of the parameters. The computation of the eigenvectors in spectral methods is affected by the selection of the data set and parameter selection as well. The sparsity of the matrix has a big impact on the time required to solve the eigenproblem. For these reasons, it is very difficult to identify the best approach in terms of both accuracy and complexity.

Chapter 4

Advances on Relational Clustering

In this Chapter, we study in detail the relational duals of four four fuzzy clustering algorithms, when the relational matrix is not metric. In this framework, some approaches have been proposed to transform the dissimilarities between patterns from non-metric to metric. Non-metric dissimilarities are not symmetric, and do not obey to the triangular inequality. The transformations needed to let the dissimilarities become metric are symmetrization and shift operations. The symmetrization operation makes the dissimilarities symmetric. Shift means that a constant value is added to the pairwise dissimilarities, to let them satisfy the triangular inequality[1]. The point is how these transformations influence the behavior of the relational clustering algorithms. It has been shown that they do not influence the K-means objective function [RLBM02, RLKB03]. In other words, changing the dissimilarities with their transformed versions does not reflect any changes on the objective function. In fact, it changes by a constant that does not affect the optimization. Once the dissimilarities are metric, they can be considered as pairwise squared Euclidean distances between patterns. This is the link with clustering methods using positive semidefinite kernels. Such kernels can be obtained by the dissimilarity matrix, and each entry is a scalar product between vectors representing the original objects. These are called embedding vectors, and are not computed explicitly. The pairwise scalar products contain enough information to let to apply the K-means family algorithms on the embedding vectors. This corresponds to the clustering in feature space [FCMR08].

This Chapter explicitly shows how the objective functions of four clustering algorithms based on fuzzy memberships change, due to dissimilarities transformations. The considered clustering algorithms are: Fuzzy c-means I (FCM I) [Bez81], Fuzzy c-means II (FCM II) [BL94], Possibilistic c-means I (PCM I) [KK93], and Possibilistic c-means II (PCM II) [KK96]. The main contributions include the lack of invariance to shift operations, as well as the invariance to symmetrization. As a byproduct, the kernel versions of

[1] In fact, we require the stronger condition that the dissimilarities become squared Euclidean distances.

FCM I, FCM II, PCM I and PCM II are obtained, that can be viewed as relational dual of the four algorithms. FCM II and PCM I in feature space have never been proposed before, while FCM I and PCM II in feature space can be found in Refs. [ZC02] and [FMR07]. The relational duals of FCM I and PCM I have been proposed in Ref. [HDB89] and [dCOF06]; the non-Euclidean case is studied in Ref. [HB94] for FCM I. The relational dual of FCM II and PCM II have never been proposed before. The experimental analysis shows the effect of the dissimilarities transformations on the four considered clustering algorithms.

The next Section discusses how to embed in Euclidean spaces sets of patterns described by pairwise dissimilarities, along with some basic concepts on positive semidefinite kernels. Section 4.2 shows how the objective functions of four K-Means style fuzzy clustering algorithms change, due to distance transformations; Section 4.3 provides an experimental analysis on synthetic and real data sets, and then the conclusions are drawn. Many technical details concerning some theoretical aspects, can be found in Section 4.5. Part of this Chapter can be found in form of a technical report [Fil07].

4.1 Embedding Objects Described by Pairwise Dissimilarities in Euclidean Spaces

Let $Y = \{y_1, \ldots, y_n\}$ be a set of objects and $r : Y \times Y \to \mathbb{R}$ a function between pairs of its elements. The conditions that r must satisfy to be a distance are:

- $r(y_i, y_j) \geq 0 \quad \forall i, j = 1, \ldots, n$ and $r(y_i, y_i) = 0 \quad \forall i = 1, \ldots, n$ (Positivity);
- $r(y_i, y_j) = r(y_j, y_i) \quad \forall i, j = 1, \ldots, n$ (Symmetry) ;
- $r(y_i, y_j) + r(y_j, y_k) \geq r(y_i, y_k) \quad \forall i, j, k = 1, \ldots, n$ (Triangular inequality).

Let's assume that r satisfies only the first condition. In this case, r can be interpreted as a dissimilarity measure between the elements of the set Y. Clearly, it is not possible to embed the objects according to r in a Euclidean space, as long as it does not satisfy also the other two conditions. The only way to cope with this problem is to apply some transformations to let r become a distance function. Regarding the symmetry, the following, for instance, could represent a solution:

$$\hat{r}(y_i, y_j) = \max(r(y_i, y_j), r(y_j, y_i)) \qquad \forall i, j \tag{4.1}$$

or:

$$\hat{r}(y_i, y_j) = \frac{1}{2}(r(y_i, y_j) + r(y_j, y_i)) \qquad \forall i, j \tag{4.2}$$

Depending on the application, one can choose the most suitable solution to fix the symmetry.

Once the symmetry is fixed, to make r satisfy the triangular inequality, a constant shift 2α can be added to all the pairwise distances, excluding the dissimilarity between a pattern and itself:

$$\tilde{r}(y_i, y_j) = r(y_i, y_j) + 2\alpha \qquad \forall i \neq j \tag{4.3}$$

Let's introduce R as the $n \times n$ matrix with entries $r_{ij} = r(y_i, y_j)$. Let $e = \{1, 1, \ldots, 1\}^T$ and I the $n \times n$ identity matrix. Eq. 4.3 is equivalent to:

$$\tilde{R} = R + 2\alpha(ee^T - I) \tag{4.4}$$

The natural question arises: how can we choose α to guarantee that $\tilde{R}$ is a squared Euclidean distance matrix? The answer is in a theorem that can be found in Refs. [LM04, RLKB03]. In this Section the theorem is reported, while the proof can be found in Section 4.5.2.

Before showing the theorem, some preliminary definitions are needed. Let's decompose R by means of a matrix S:

$$r_{ij} = s_{ii} + s_{jj} - 2s_{ij} \tag{4.5}$$

Let $Q = I - \frac{1}{n}ee^T$. The centralized version P^c of a generic matrix P is defined:

$$P^c = QPQ \tag{4.6}$$

It's clear from Eq. 4.5 that S is not uniquely determined by R. All the matrices $S + \alpha ee^T$, for instance, lead to the same matrix R, $\forall \alpha \in \mathbb{R}$ is. It can be proved, however, that the centralized version of S is uniquely determined by R (see Section 4.5.1):

$$S^c = -\frac{R^c}{2} \tag{4.7}$$

Now we have all the elements to claim that:

Theorem 4.1.1. *R is a squared Euclidean distance matrix if and only if $S^c \succeq 0$.*

The proof can be found in Section 4.5.2. The theorem states that S^c must be positive semidefinite to ensure that R is a squared Euclidean distance matrix. It is well known that the eigenvalues λ_i of positive semidefinite matrices satisfy $\lambda_i \geq 0 \quad \forall i = 1, \ldots, n$ [Apo67]. If at least one eigenvalue of S^c is negative, R is a squared Euclidean distance matrix. Let λ_1 be the smallest eigenvalue of S^c. Simple concepts of linear algebra ensure that the following diagonal shift to S^c:

$$\tilde{S^c} = S^c - \lambda_1 I \tag{4.8}$$

makes $\tilde{S^c}$ positive semidefinite. The diagonal shift of S^c transforms R in a matrix representing squared Euclidean distances. The resulting transformation on R is the following:

$$\tilde{R} = R - 2\lambda_1(ee^T - I) \tag{4.9}$$

Since $\tilde{S^c}$ is positive semidefinite, it can be thought as representing a scalar product. Thus, it exists a matrix X for which:

$$\tilde{S^c} = XX^T \tag{4.10}$$

The rows of X are the realization of the embedding vectors $\mathbf{x}_i$. In other words, each element y_i of the set Y has been embedded in a Euclidean space and is represented by $\mathbf{x}_i$. The entries of $\tilde{S^c}$ are the scalar product between the vectors $\mathbf{x}_i$.

Resuming, if the only thing known about the data to analyze are the pairwise dissimilarities, the matrix S^c can be checked for positive semidefiniteness. If it is, S^c can be kept as is, otherwise the diagonal shift to S^c has to be applied. Either way, S^c or $\tilde{S^c}$ is the product of two unknown matrices X. This is the link between the theory of embedding a set of objects and the theory of kernel methods. $\tilde{S^c}$ can be interpreted as the Gram matrix that is used in kernel algorithms. In Ref. [LM04, LRBM06] the authors give an interpretation of the negative eigenvalues of S^c.

Before closing this Section, it is worth noting that in general there are two options when shifting R to obtain $\tilde{S^c}$. The first is to shift the dissimilarities R obtaining $\tilde{R}$, and then compute $\tilde{S^c}$ associated to $\tilde{R}$. Let's call this procedure *preshift*:

$$\tilde{S^c} = -\frac{1}{2}(Q\tilde{R}Q) \tag{4.11}$$

The second choice, the *postshift*, is to compute S^c associated to R, and then shift its diagonal elements:

$$S^c + \alpha I \tag{4.12}$$

Both the methods allow to compute a matrix S corresponding to the same shift of the distances, but:

$$S^c + \alpha I \neq -\frac{1}{2}(Q\tilde{R}Q) \tag{4.13}$$

Section 4.5.3 shows that the choice between preshift and postshift does not affect the studied clustering algorithms.

4.2 Fuzzy Central Clustering Objective Functions

We recall the general formulation of the central clustering objective functions (see Chapter 2):

$$J(U, V) = G(U, V) + H(U) + W(U) \tag{4.14}$$

The first term is a measure of the distortion and the second is an entropic score on the memberships. The distortion can be written as the following sum:

$$G(U,V) = 2\sum_{i=1}^{c}\sum_{h=1}^{n} u_{ih}^{\theta}\|\mathbf{x}_h - \mathbf{v}_i\|^2 \tag{4.15}$$

with $\theta \geq 1$. The aim of the entropy term $H(U)$ is to avoid trivial solutions where all the memberships are zero or equally shared among the clusters. For the algorithms having a constraint on U, the Lagrange multipliers technique has to be followed in order to perform the optimization. This means that a further term $W(U)$ is needed.

The technique used by these methods to perform the minimization is the Picard iteration technique, that is based on the iteration of the solutions of these two equations:

$$\frac{\partial L(U,V)}{\partial \mathbf{v}_i} = 0 \tag{4.16}$$

$$\frac{\partial L(U,V)}{\partial u_{ih}} = 0 \tag{4.17}$$

The algorithms stop when a convergence criterion is satisfied on U, V or G. Usually the following is considered:

$$\|U - U'\|_p < \varepsilon \tag{4.18}$$

where U' is the updated version of the memberships and $\|\|_p$ is a p-norm.

Since $L(U,V)$ depends on V only because of G, the update of the $\mathbf{v}_i$ is the same for all the considered algorithms. From Eq. 2.10:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih}^{\theta}\mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}^{\theta}} \tag{4.19}$$

Now it is possible to prove that the following functional is equivalent to $G(U,V)$ (see Section 4.5.4):

$$G(U) = \sum_{i=1}^{c} \frac{\sum_{r=1}^{n}\sum_{s=1}^{n} u_{ir}^{\theta}u_{is}^{\theta}d_{rs}^2}{\sum_{r=1}^{n} u_{ir}^{\theta}} \tag{4.20}$$

Here d_{rs}^2 is the squared Euclidean distance between patterns r and s. This allows to write the objective function only in terms of U, when the description of the data set is in terms of pairwise distances.

In the non-metric case, it is not possible to identify d_{rs}^2 as the squared Euclidean distance between patterns r and s. Anyway, it is still possible to think that the objective function of the clustering is:

$$G(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} r_{hk}}{\sum_{h=1}^{n} u_{ih}^{\theta}} \tag{4.21}$$

In the following, this way of writing $G(U)$ will be useful to show how the objective functions change with respect to dissimilarities transformations.

4.2.1 Invariance of $G(U)$ to Symmetrization of R

Let's analyze what happens to the Lagrangian L when R is transformed in the following way:

$$\hat{r}_{ij} = \frac{r_{ij} + r_{ji}}{2} \tag{4.22}$$

which is equivalent to:

$$\hat{R} = \frac{R + R^T}{2} \tag{4.23}$$

It's clear that the only term of the functional affected by the distance transformation is $G(U)$. Showing that:

$$\begin{aligned} \sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} \hat{r}_{hk} &= \frac{1}{2}\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} r_{hk} + \frac{1}{2}\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} r_{kh} \\ &= \sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} r_{hk} \end{aligned} \tag{4.24}$$

the invariance of the Lagrangian $L(U)$ to the symmetrization of R is proved. In other words, in presence of a non-symmetric R, the symmetrization in Eq. 4.22 does not change the clustering objective function. In force of this result, R will be considered symmetric in the rest of this Chapter.

4.2.2 Transformation of $G(U)$ to Shift Operations

This Section analyzes what happens to the Lagrangian L when transforming the distances in the following way:

$$\tilde{r}_{hk} = r_{hk} + 2\alpha \quad \forall\, h \neq k \tag{4.25}$$

which is equivalent to Eq. 4.4:

The only term in the Lagrangian $L(U)$ changing due the dissimilarities shift is $G(U)$:

$$\begin{aligned} G_\alpha(U) &= \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} \tilde{r}_{hk}}{\sum_{h=1}^{n} u_{ih}^{\theta}} \\ &= G(U) + 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^{\theta} u_{ik}^{\theta} - \sum_{h=1}^{n} u_{ik}^{2\theta}}{\sum_{h=1}^{n} u_{ih}^{\theta}} \\ &= G(U) + 2\alpha \sum_{i=1}^{c}\sum_{h=1}^{n} u_{ih}^{\theta} - 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^{2\theta}}{\sum_{h=1}^{n} u_{ih}^{\theta}} \end{aligned} \tag{4.26}$$

The Lagrangian will result in:

$$L_\alpha(U) = G(U) + H(U) + W(U) + 2\alpha\left(A(U) - B(U)\right) \tag{4.27}$$

This result shows that in general the Lagrangian for the K-means family algorithms is not invariant to such transformations. Only for K-means $A(U) - B(U) = n - c$, which means that the K-means objective function is invariant to distance shifts. Besides, for fuzzy clustering algorithms for which $\theta = 1$, $A(U)$ reduces to n.

In general, since $\theta \geq 1$ and $u_{ih} \in [0, 1]$, the following two inequalities are satisfied:

$$A(U) = \sum_{i=1}^{c}\sum_{h=1}^{n} u_{ih}^{\theta} < n \tag{4.28}$$

$$B(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^{2\theta}}{\sum_{h=1}^{n} u_{ih}^{\theta}} < c \tag{4.29}$$

The contributions of $A(U)$ and $B(U)$ to $L_\alpha(U)$ are weighted by 2α. This means that $L_\alpha(U)$ can be strongly affected by large shift values.

Given a clustering algorithm, in order to obtain the update of the memberships, the derivatives of the Lagrangian with respect to them have to be set to zero. From that, an update formula for the memberships has to be obtained (in presence of constraints, this implies to compute also the value of the Lagrange multipliers). Let's consider the term $B(U)$:

$$\frac{\partial B(U)}{\partial u_{ih}} = \frac{2\theta u_{ih}^{2\theta-1} \sum_{r=1}^{n} u_{ir}^{\theta} - \theta u_{ih}^{\theta-1} \sum_{h=1}^{n} u_{ih}^{2\theta}}{(\sum_{h=1}^{n} u_{ih}^{\theta})^2} \tag{4.30}$$

This is not easily invertible when summed to the derivative of the other terms to obtain zero. The next Section provides an experimental analysis showing the effect of the shift operation on the behavior of the memberships during the optimization.

4.2.3 Analysis of Four Clustering Algorithms

This Section shows the results just obtained to four clustering algorithms based on fuzzy memberships: Fuzzy c-means I (FCM I) [Bez81], Fuzzy c-means II (FCM II) [BL94], Possibilistic c-means I (PCM I) [KK93], and Possibilistic c-means II (PCM II) [KK96] (see Section 2.1.2 for the complete derivation of these four algorithms). In Tab. 4.1, the terms of the Lagrangian in Eq. 4.14 for the mentioned clustering algorithms are resumed. Since the sum of the memberships of a pattern to all the clusters is constrained to be one in fuzzy clustering, the term $W(U)$ is introduced. For the possibilistic algorithms $W(U) = 0$, since the memberships are not constrained. In fact, for these algorithms the minimization of $L(U)$ should be done in the hypercube $u_{ih} \in [0, 1]$. Since the form assumed by the update equations, this constrain is automatically satisfied. In FCM I and PCM I, the exponent of the memberships θ is usually called m, while $\theta = 1$ in FCM II and PCM II.

Tab. 4.2 resumes the Lagrangian $L_\alpha(U)$ of the discussed clustering algorithms, considering also the effect of the shift. K-means is invariant to distance shifts since $A(U) = n$ and $B(U) = c$. In FCM II and PCM II, $A(U) = n$; in FCM I and PCM I, both $A(U)$ and $B(U)$ are not zero.

From the analysis in Section 4.1, it is possible to choose α big enough to guarantee that $\tilde{R}$ represents a squared Euclidean distance matrix. This allows to represent each pattern in a Euclidean space $\mathcal{F}$, where the discussed clustering algorithms can be applied. In fact, the positions of the patterns in $\mathcal{F}$ is still encoded in $\tilde{R}$, and thus is unknown. Nevertheless, using the fact that $K = \tilde{S}^c$ contains the scalar products between patterns, an update formula for the memberships can be explicitly found. Each pattern is represented by a vector $\mathbf{x}_i \in \mathcal{F}$ and the set of centroids V is composed of prototypes in $\mathcal{F}$. Let's analyze, for instance, the update equations for $\mathbf{v}_i$ and u_{ih} for FCM II:

$$u_{ih} = \frac{\exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_i\|^2}{\lambda}\right)}{\sum_{j=1}^{c} \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}_j\|^2}{\lambda}\right)} \tag{4.31}$$

Table 4.1: Resuming table of the entropy functions, θ value, and constraints of the considered clustering algorithms.

Method	θ	$H(U)$	$W(U)$
FCM I	m	0	$\sum_{h=1}^{n} \beta_h \left(1 - \sum_{i=1}^{c} u_{ih}\right)$
FCM II	1	$\lambda \sum_{i=1}^{c} \sum_{h=1}^{n} u_{ih} \ln(u_{ih})$	$\sum_{h=1}^{n} \beta_h \left(1 - \sum_{i=1}^{c} u_{ih}\right)$
PCM I	m	$\sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (1 - u_{ih})^m$	0
PCM II	1	$\sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (u_{ih} \ln(u_{ih}) - u_{ih})$	0

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih} \mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}} \tag{4.32}$$

Since we don't know explicitly the vectors $\mathbf{x}_i$, it would not be possible to explicitly compute $\mathbf{v}_i$. Substituting Eq. 4.32 in Eq. 4.31, we obtain:

$$\begin{aligned}
\|\mathbf{x}_h - \mathbf{v}_i\|^2 &= \left\| \mathbf{x}_h - \frac{\sum_{r=1}^{n} u_{ir} \mathbf{x}_r}{\sum_{r=1}^{n} u_{ir}} \right\|^2 \\
&= \left(\mathbf{x}_h - \frac{\sum_{r=1}^{n} u_{ir} \mathbf{x}_r}{\sum_{r=1}^{n} u_{ir}} \right) \left(\mathbf{x}_h - \frac{\sum_{r=1}^{n} u_{ir} \mathbf{x}_r}{\sum_{r=1}^{n} u_{ir}} \right) \\
&= \mathbf{x}_h \mathbf{x}_h - 2 \frac{\sum_{r=1}^{n} u_{ir} \mathbf{x}_r \mathbf{x}_h}{\sum_{r=1}^{n} u_{ir}} + \frac{\sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir} u_{is} \mathbf{x}_r \mathbf{x}_s}{(\sum_{r=1}^{n} u_{ir})^2} \\
&= k_{hh} - 2 \frac{\sum_{r=1}^{n} u_{ir} k_{rh}}{\sum_{r=1}^{n} u_{ir}} + \frac{\sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir} u_{is} k_{rs}}{(\sum_{r=1}^{n} u_{ir})^2}
\end{aligned} \tag{4.33}$$

This allows to obtain an update equation for the memberships for the considered clustering algorithms.

To obtain a more convenient way of writing the update equations, let U_θ be the $c \times n$ matrix having u_{ih}^θ as elements, and let:

$$a_i = \left(\sum_{h=1}^{n} u_{ih}^\theta \right)^{-1} \tag{4.34}$$

$$\mathbf{z}^{(0)} = \mathrm{diag}(K) \tag{4.35}$$

Table 4.2: Resuming table of the objective functions of the considered clustering algorithms considering also the contribution given by the shift operation.

FCM I $L_\alpha(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^m u_{ik}^m r_{hk}}{\sum_{h=1}^{n} u_{ih}^m} + \sum_{h=1}^{n} \beta_h \left(1 - \sum_{i=1}^{c} u_{ih}\right) + 2\alpha \sum_{i=1}^{c}\sum_{h=1}^{n} u_{ih}^m - 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^{2m}}{\sum_{h=1}^{n} u_{ih}^m}$
FCM II $L_\alpha(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih} u_{ik} r_{hk}}{\sum_{h=1}^{n} u_{ih}} + \lambda \sum_{h=1}^{n}\sum_{i=1}^{c} u_{ih}\ln(u_{ih}) + \sum_{h=1}^{n} \beta_h \left(1 - \sum_{i=1}^{c} u_{ih}\right) + 2\alpha n - 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^2}{\sum_{h=1}^{n} u_{ih}}$
PCM I $L_\alpha(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih}^m u_{ik}^m r_{hk}}{\sum_{h=1}^{n} u_{ih}^m} + \sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (1 - u_{ih})^m + 2\alpha \sum_{h=1}^{n}\sum_{i=1}^{c} u_{ih}^m - 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^{2m}}{\sum_{h=1}^{n} u_{ih}^m}$
PCM II $L_\alpha(U) = \sum_{i=1}^{c} \frac{\sum_{h=1}^{n}\sum_{k=1}^{n} u_{ih} u_{ik} r_{hk}}{\sum_{h=1}^{n} u_{ih}} + \sum_{i=1}^{c} \eta_i \sum_{h=1}^{n} (u_{ih}\ln(u_{ih}) - u_{ih}) + 2\alpha n - 2\alpha \sum_{i=1}^{c} \frac{\sum_{h=1}^{n} u_{ih}^2}{\sum_{h=1}^{n} u_{ih}}$

Table 4.3: Resuming table of the memberships update equations, for the considered clustering algorithms.

FCM I $u_{ih}^{-1} = \sum_{j=1}^{c} \left(\frac{z_h^{(0)} - 2a_i z_{ih}^{(1)} + a_i^2 z_i^{(2)}}{z_h^{(0)} - 2a_j z_{jh}^{(1)} + a_j^2 z_j^{(2)}} \right)^{\frac{1}{m-1}}$
FCM II $u_{ih} = \frac{\exp\left(-\frac{z_h^{(0)} - 2a_i z_{ih}^{(1)} + a_i^2 z_i^{(2)}}{\lambda} \right)}{\sum_{j=1}^{c} \exp\left(-\frac{z_h^{(0)} - 2a_j z_{jh}^{(1)} + a_j^2 z_i^{(2)}}{\lambda} \right)}$
PCM I $u_{ih}^{-1} = \left(\frac{z_h^{(0)} - 2a_i z_{ih}^{(1)} + a_i^2 z_i^{(2)}}{\eta_i} \right)^{\frac{1}{m-1}} + 1$
PCM II $u_{ih} = \exp\left(-\frac{z_h^{(0)} - 2a_i z_{ih}^{(1)} + a_i^2 z_i^{(2)}}{\eta_i} \right)$

$$Z^{(1)} = U_\theta K \tag{4.36}$$

$$\mathbf{z}^{(2)} = \mathrm{diag}(U_\theta K U_\theta^T) \tag{4.37}$$

Eq. 4.33 becomes:

$$\|\mathbf{x}_h - \mathbf{v}_i\|^2 = z_h^{(0)} - 2a_i z_{ih}^{(1)} + a_i^2 z_i^{(2)} \tag{4.38}$$

Tab. 4.3 shows the update equations of the memberships for the considered clustering algorithms. Tab. 4.4 shows the steps composing the considered clustering algorithms.

4.3 Experimental Analysis

4.3.1 Synthetic Data Set

The presented clustering algorithms have been tested on a synthetic data set composed of two clusters in two dimensions (Fig. 4.1). Each cluster is composed of 200 points sampled from a Gaussian distribution. The position of their centers are respectively in

Table 4.4: Pseudocode of the considered clustering algorithms

1. **if** R is not symmetric, **then** symmetrize it using Eq. 4.22;
2. Compute S^c using Eq. 4.7;
3. **if** $S^c \succeq 0$ **then** $K = S^c$;
4. **else** $K = S^c - \lambda_1 I$;
5. Initialize parameters: c, m (FCM I, PCM I), λ (FCM II), η_i (PCM I, PCM II);
6. Initialize U;
7. Update U using the update equation in Tab. 4.3 corresponding to the chosen method;
8. **if** the convergence criteria is not satisfied **then** go to step 7;
9. **else** stop.

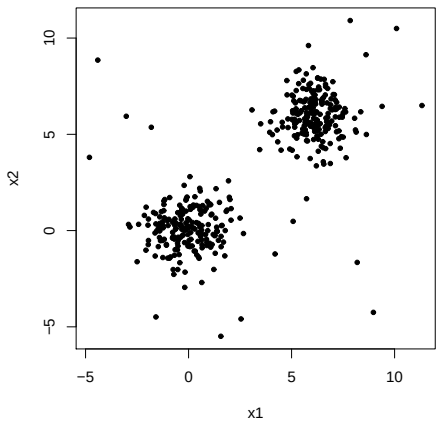


Figure 4.1: Plot of the synthetic data set composed of two clusters and some outliers.

$(0, 0)$ and $(6, 6)$, and the standard deviations are equal to one for both the features and clusters. Twenty outlier points have been added; they have been extracted with a uniform distribution in the set $[-6, 12] \times [-6, 12]$. The average of the squared distances is 43.4, the median is 34.4, and the maximum is 360.9.

For all the tested algorithms, the behavior of the memberships have been analyzed during the optimization, for different values of α. In order to do that, the r_{ij} have been set to the squared Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, and have been shifted with different values of α. This can be done in two equivalent ways, namely the preshift and the postshift (see Section 4.5.3). The proposed algorithms have been run on the modified data sets. During the optimization, the memberships have been recorded to see how the distance shifts affected their behavior. At each iteration, the difference between the matrix U when $\alpha = 0$ and U' for an $\alpha \neq 0$ has been measured. The analysis has been made on the basis of these two scores:

$$\mathrm{sd}(U - U') = \sqrt{\left(\frac{\sum_{h=1}^{n} \sum_{i=1}^{c} (u_{ih} - u'_{ih})^2}{c\, n} \right)} \tag{4.39}$$

$$\max(U - U') = \max_{i,h} \left(|u_{ih} - u'_{ih}| \right) \tag{4.40}$$

averaged over 100 runs.

FCM I has been tried with three different values of m, in particular $m = 1.1$, 1.5, 2. Fig. 4.2 shows the behavior of the memberships during the optimization for different values of α and m. The first row in Fig. 4.2 corresponds to $m = 2$, the one in the middle to $m = 1.5$, and the one on the bottom to $m = 1.1$. For small α the results are almost invariant as expected. For values of α of the order of the average of the squared distances, the memberships have a very different behavior with respect to those on the original set. Reducing the fuzziness m it can be noticed that the results are better. This is not surprising since for m tending to 1, FCM I behaves like K-means which is invariant to shift transformations. At the end of the algorithm, the memberships can be defuzzified using a threshold of 0.5 to obtain the cluster labels. The cluster labels for different values of alpha have been found to be identical for all the tested valued of α.

FCM II has been tried with three different values of λ, in particular $\lambda = 10$, 20, 30. For such values of λ, the resulting memberships range from almost crisp to moderate fuzzy. For different fuzziness levels (higher λ leads to fuzzier solutions), the memberships are almost invariant, even for values of α higher than the maximum of the original squared distances (Fig. 4.3). The Lagrangian in FCM II is not invariant to shift transformations only because of the term $B(U)$. The fact that $A(U)$ is constant gives to FCM II more robustness to distance shifts.

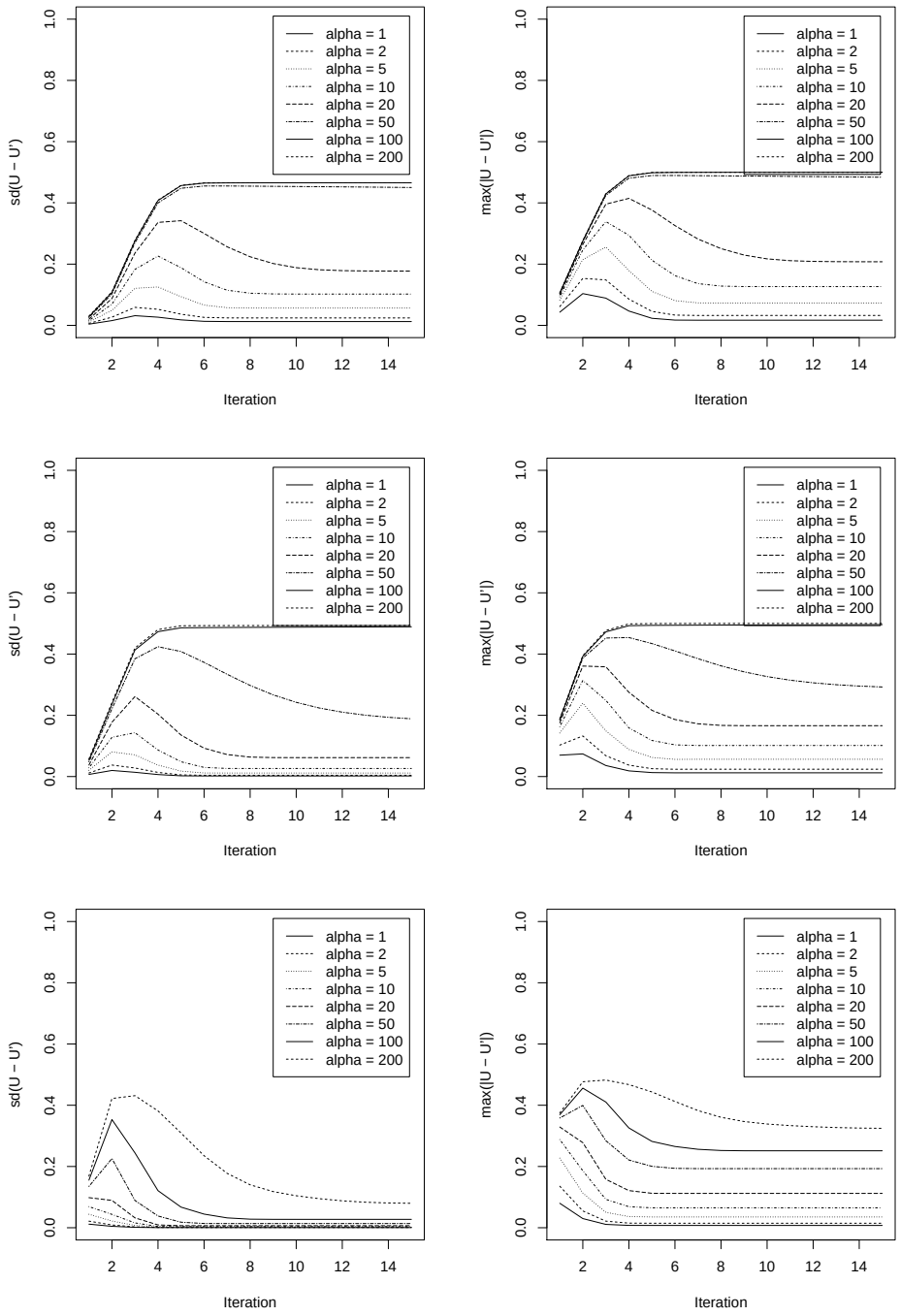


Figure 4.2: FCM I - Behavior of the memberships during the optimization for different values of α. First row $m = 2$, second row $m = 1.5$, third row $m = 1.1$. Results are averaged over 100 repetitions with different initialization of U.

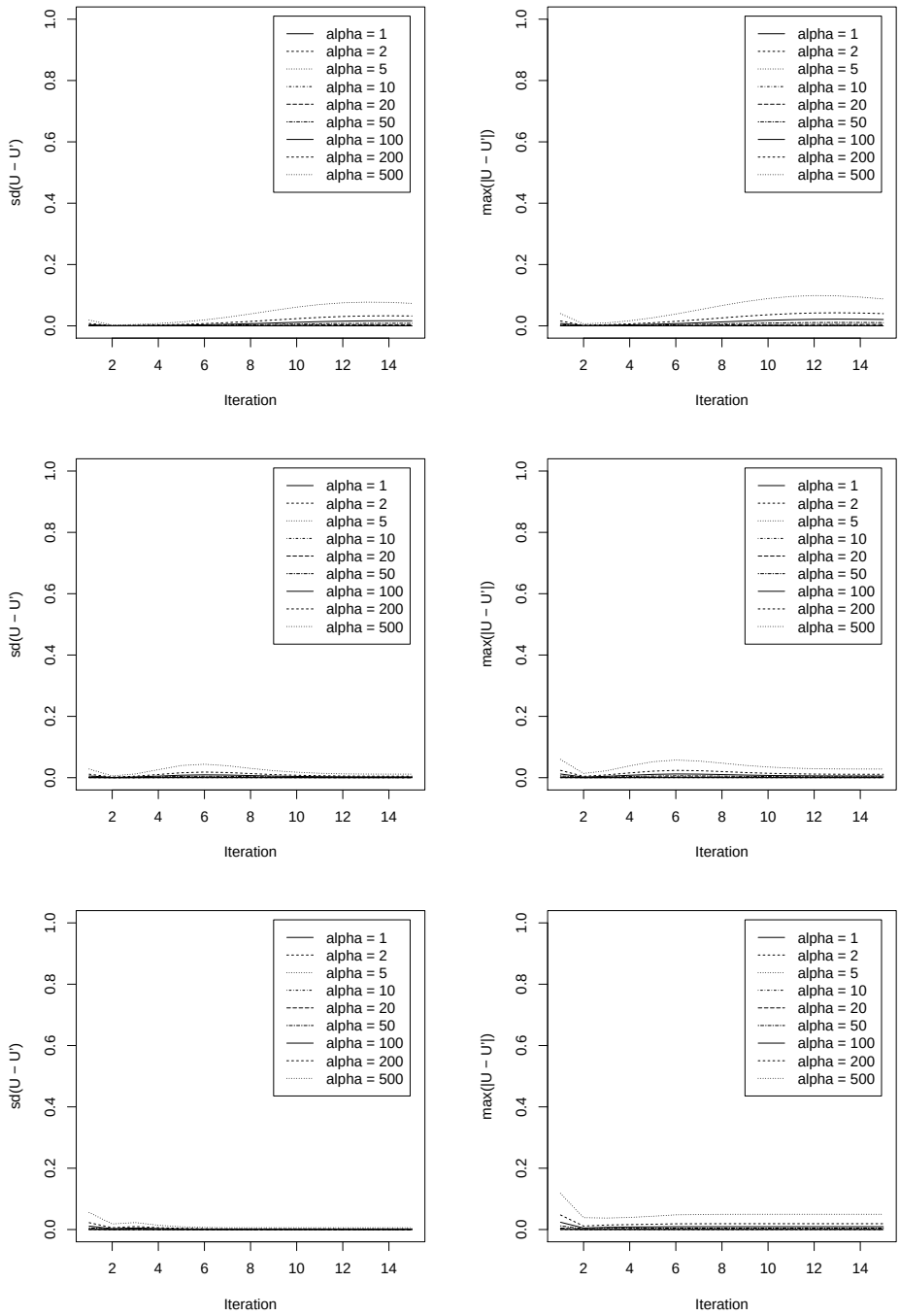


Figure 4.3: FCM II - Behavior of the memberships during the optimization for different values of α. First row $\lambda = 30$, second row $\lambda = 20$, third row $\lambda = 10$. Results are averaged over 100 repetitions with different initialization of U.

PCM I Fig. 4.4 shows the behavior of the memberships during the optimization for the PCM I with different values of m, in particular $m = 1.1, 1.5, 2$. The initialization of the memberships has been done using the result obtained by the FCM II, since it showed high robustness to distance shifts. The values of η_i have been computed on the basis of the memberships obtained by the FCM II. It can be seen that even for small values of α, the behavior of the memberships is strongly affected by the shift operation.

PCM II The initialization of the memberships and the computation of the η_i have been done on the basis of the result obtained by the FCM II. In PCM II there are no further parameters to set up. Fig. 4.5 shows that also PCM II is strongly affected by dissimilarities shifts, even for small values of α.

4.3.2 USPS Data Set

The studied algorithms have been tested on the USPS data set, which has been studied also in Refs. [SSM98, LM04]. It is composed of 9298 images acquired and processed from handwritten zip-codes appeared on real US mail. Each image is 16×16 pixels; the training set in composed by 7219 images and the test set by 2001 images. As in Ref. [LM04], only the characters in the training set labeled as "0" and "7" have been considered, obtaining a subset of 1839 images. The dissimilarity function used in Ref. [LM04] is based on the Simpson score, which is a matching function between binary images. Given two binary images, the following matrix can be constructed:

		Img 1	
		0	1
Img 2	0	d	c
	1	b	a

where: a is the number of pixels that are white in both the images; b is the number of pixels that are white in Img 2 and black in Img 1; c is the number of pixels that are white in Img 1 and black in Img 2; d is the number of pixels that are black in both the images. The Simpson score of two binary images is defined as:

$$l = \frac{a}{\min(a+b, a+c)} \tag{4.41}$$

The images in the USPS data set are not binary; this has required a normalization between 0 and 1, and a thresholding at 0.5. The dissimilarity based on the Simpson score, is:

$$r_{ij} = 2 - 2l_{ij} \tag{4.42}$$

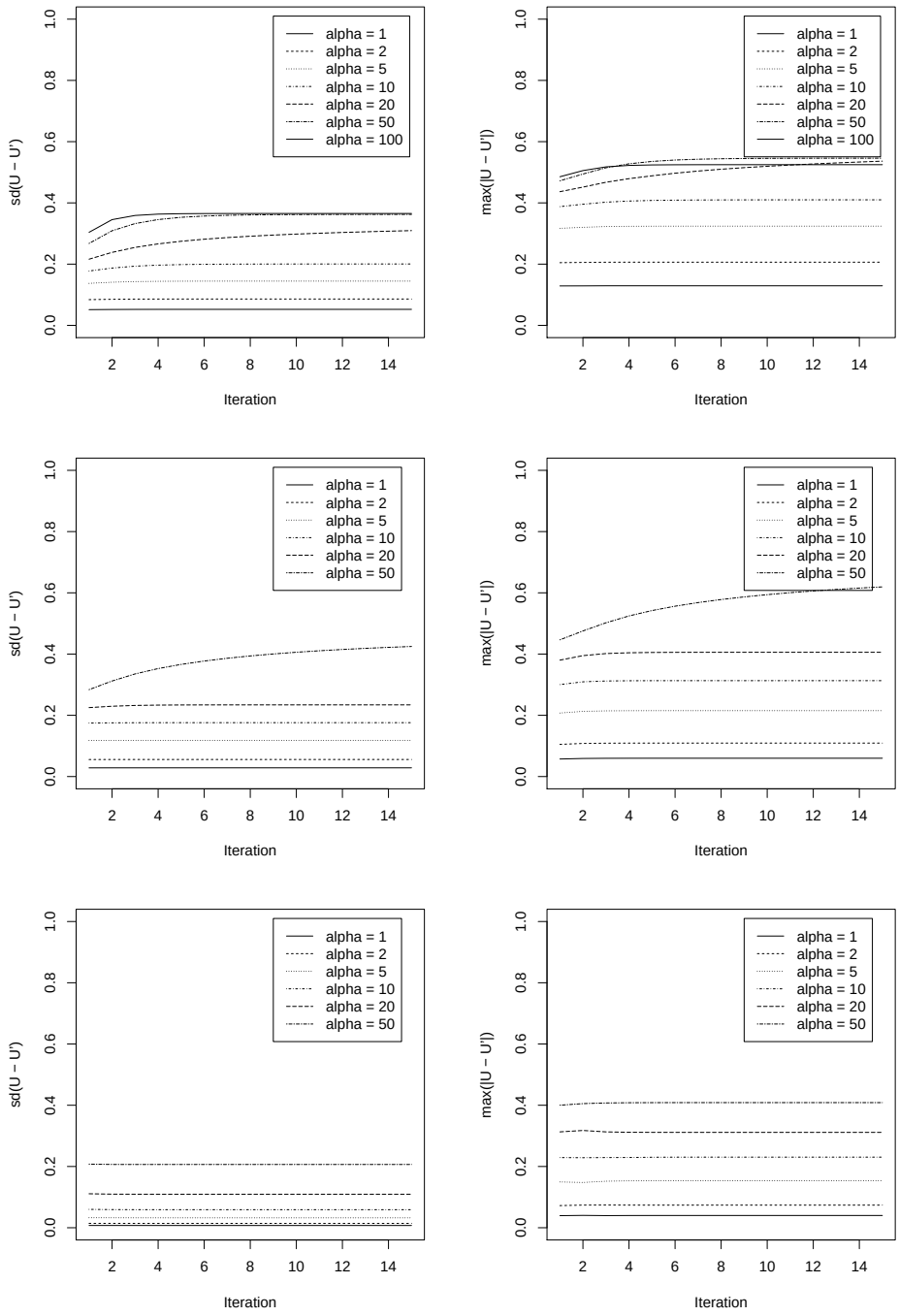


Figure 4.4: PCM I - Behavior of the memberships during the optimization for different values of α. First row $m = 2$, second row $m = 1.5$, third row $m = 1$. Results are averaged over 100 repetitions with different initialization of U.

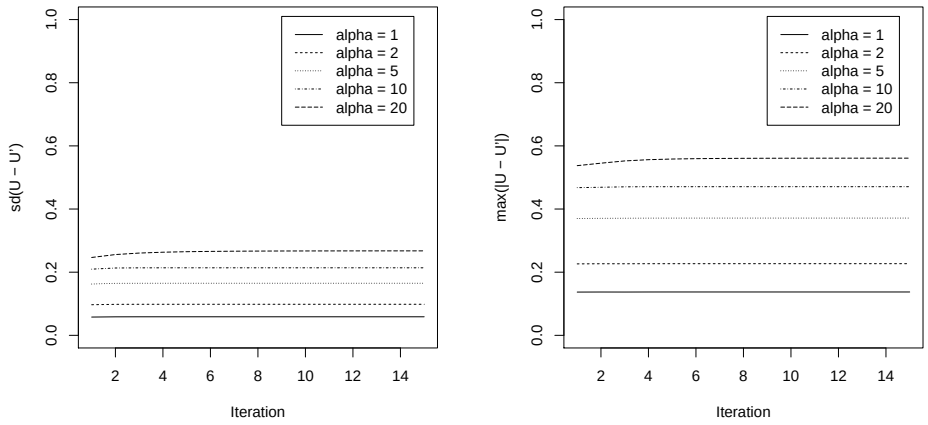


Figure 4.5: PCM II - Behavior of the memberships during the optimization for different values of α. Results are averaged over 100 repetitions with different initialization of U.

which is between 0 and 2. The mean value of R, in this data set, is 0.88, and the median is 0.92. The Simpson dissimilarity is symmetric, but does not obey to the triangular inequality. Indeed, as can be seen in Fig. 4.6, there are some negative eigenvalues of S^c. The smallest eigenvalue $\lambda_1 = -57.2$ is the value that added to the dissimilarities let $\tilde{R}$ become a squared Euclidean distance matrix. We applied the four clustering algorithms on the selected binary images, searching for two clusters.

In Fig. 4.7, we can see the plot of the entropy $H(U)$ of the memberships versus the parameters. Only FCM II, for particular values of λ, allows to obtain a meaningful distribution of the memberships. Fig. 4.8 shows the accuracy obtained of the algorithms with respect to the parameters. The accuracy is measured as the match between cluster labels and class labels. Both the entropy and the accuracy are averaged over 50 trials with different initializations. In these experiments, we noticed that FCM I resulted to be strongly affected by different initializations.

FCM II resulted the best algorithm in terms of performances. The histogram of the membership allows to refine the results, identifying the patterns that are more representative of the two clusters, and those that are on the border between them. As an illustrative example, we show (Fig. 4.9) the histogram of the highest membership of the patterns to the clusters, obtained by FCM II with $\lambda = 0.15$, that is the setup giving the best results on average (accuracy of 98.2 %). We can set a threshold on such memberships to label the patterns as objects in the border between the two clusters. By looking at the histogram, we set this threshold to 0.9. Fig. 4.9 shows the group of border objects, and the two clusters found by the algorithm. The images have been sorted with decreasing values of memberships. The image in the top-left corner has the highest membership and moving

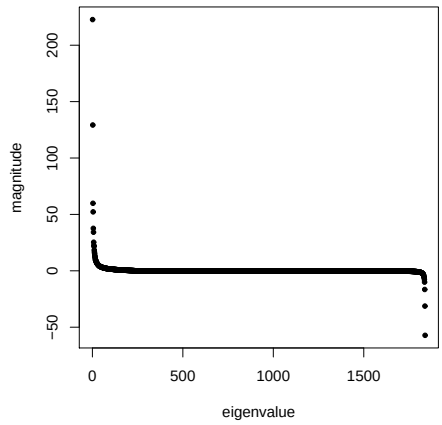


Figure 4.6: USPS data set - Eigenvalues of the matrix S^c sorted by decreasing magnitude.

to the right the memberships decrease.

4.4 Discussions

In this Chapter, four clustering algorithms based on fuzzy memberships have been studied: FCM I, FCM II, PCM I, and PCM II. In particular, it has been studied how the symmetrization and the shift operation on the dissimilarities affect their objective function. The main results include the proof of the invariance of the objective function to symmetrization and the lack of invariance to shift operations. Moreover, the four considered clustering algorithms have been presented under a more general framework, highlighting the connections between the relational clustering and the clustering in the space induced by positive semidefinite kernels.

The tests conducted on the synthetic data set show that FCM II, among the studied algorithms, is the least sensitive to shift operations. The cluster labels obtained by defuzzifying the memberships in both FCM I and FCM II are the same as the unshifted case, even for large shifts. This suggests that FCM I and FCM II could be useful to perform the optimization stage to obtain the cluster labels; anyway, the value of the memberships will be distorted by the shift. The possibilistic clustering algorithms are strongly affected by the shift operation due to the inability to deal with sparse data sets. From the results on handwritten character recognition problem, it is possible to see how FCM II performed in a real scenario. A simple analysis on the memberships can help to avoid a decision on the assignment of patterns having their membership almost equally shared among clusters.

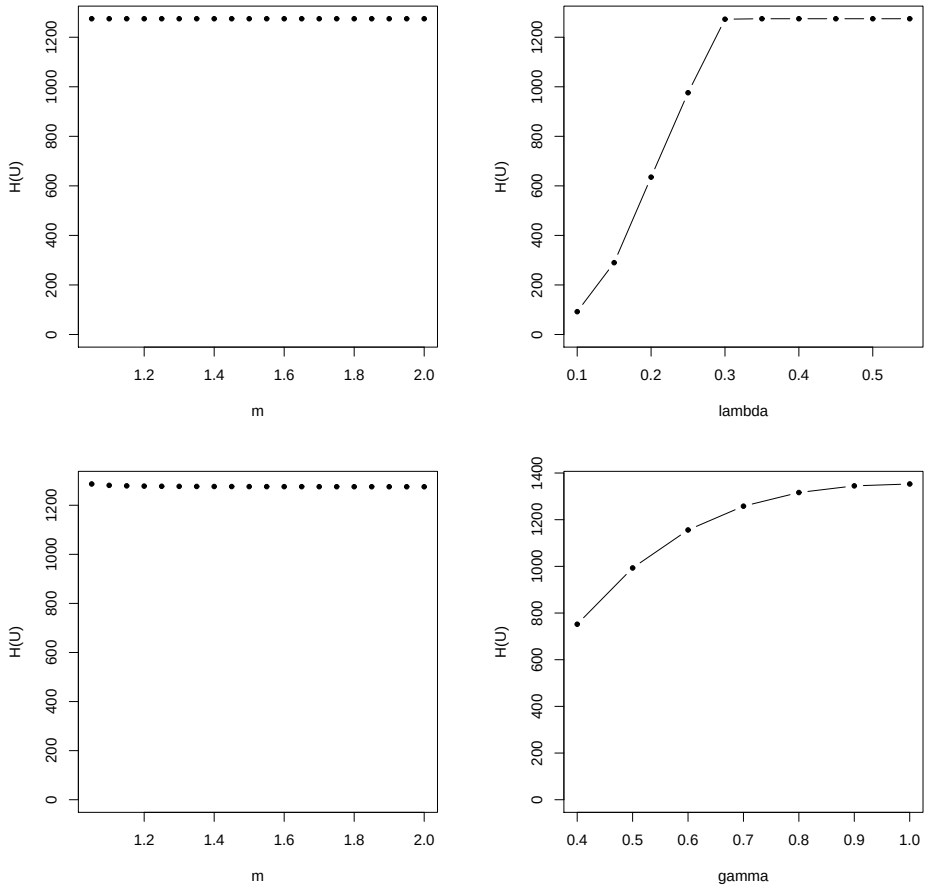


Figure 4.7: Entropy vs the clustering parameters. First row FCM I and FCM II; second row PCM I and PCM II.

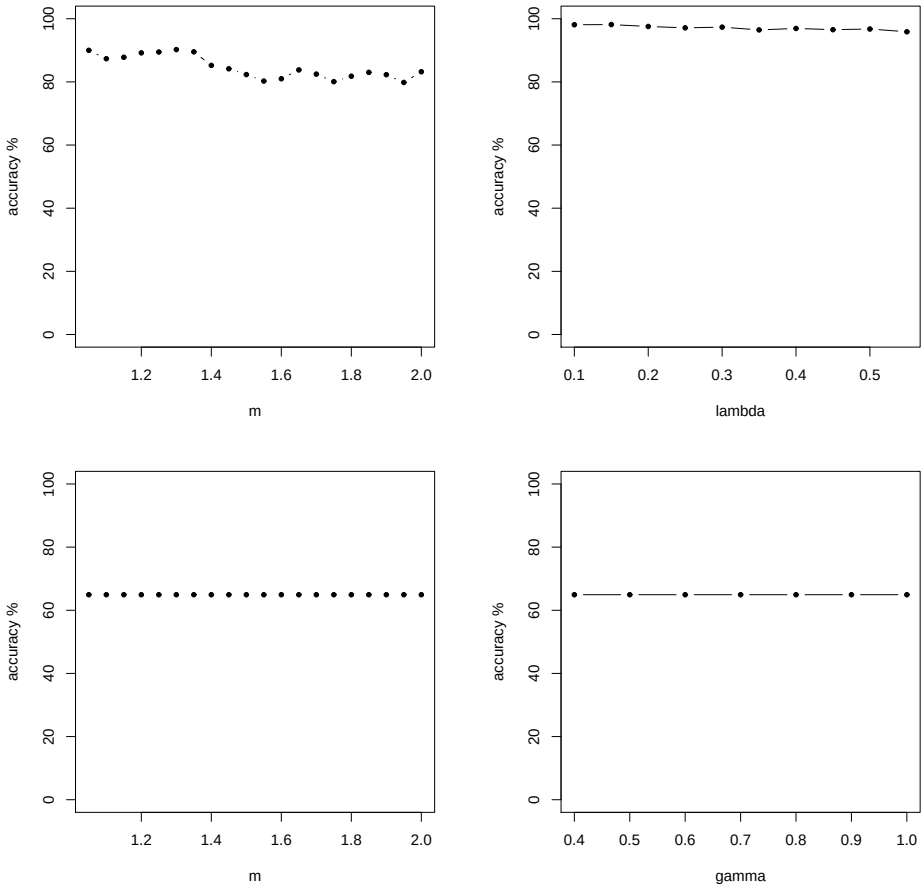


Figure 4.8: Accuracy vs the clustering parameters. First row FCM I and FCM II; second row PCM I and PCM II.

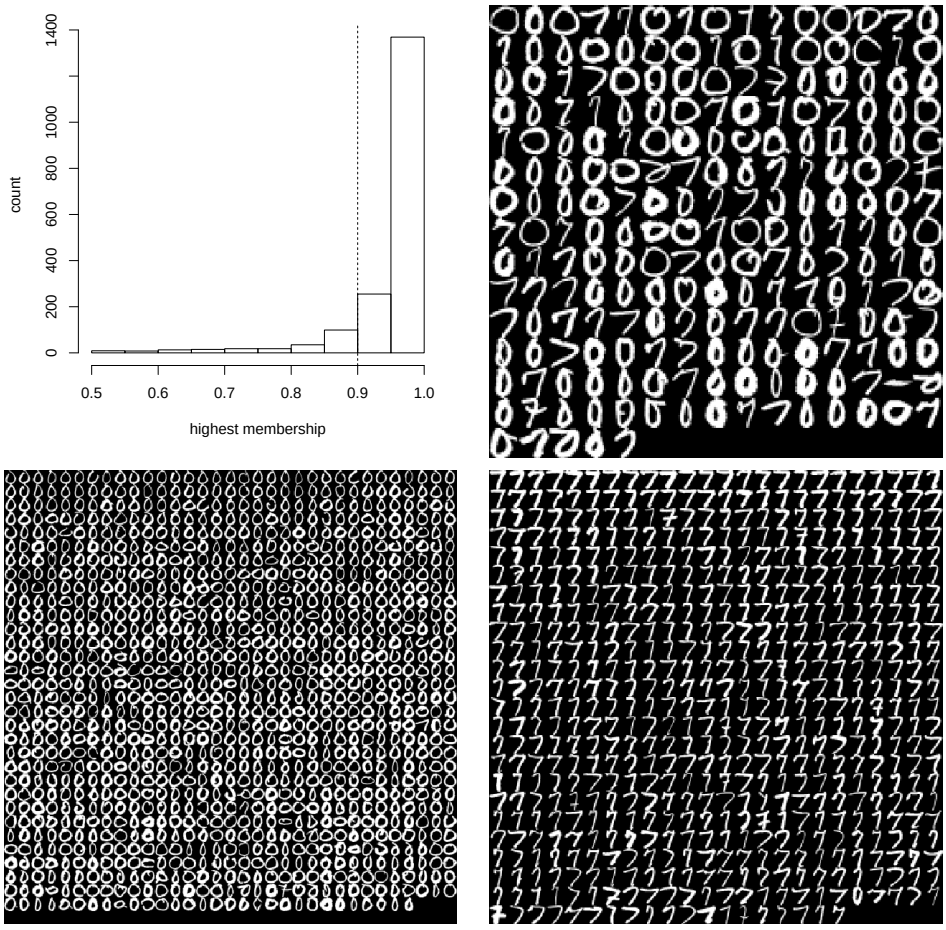


Figure 4.9: Analysis of the results obtained by FCM II with $\lambda = 0.15$. First row: histogram of the highest memberships of patterns to the two clusters and group of objects having membership below the threshold (border objects). Second row: the two clusters found by the algorithm

Other interesting studies could involve the effect of the cardinality of the data set n and the number of clusters c. It would be also interesting to try different approaches for the estimation of η_i, as suggested in Ref. [dCOF06], or see what is the difference between the behavior of memberships associated to outlier and normal patterns. All these considerations could represent the basis of new studies on the behavior of the studied clustering algorithms, for patterns described by non-metric pairwise dissimilarities.

4.5 Proofs

4.5.1 Proof that S^c is Uniquely Determined by R^c

The centralized version of a generic matrix P is the following:

$$P^c = QPQ \tag{4.43}$$

This is equivalent to:

$$p^c_{ij} = p_{ij} - \frac{1}{n}\sum_{h=1}^{n} p_{hj} - \frac{1}{n}\sum_{k=1}^{n} p_{ik} + \frac{1}{n^2}\sum_{h=1}^{n}\sum_{k=1}^{n} p_{hk} \tag{4.44}$$

Inverting Eq. 4.5, we can write:

$$s_{ij} = -\frac{1}{2}\left(r_{ij} - s_{ii} - s_{jj}\right) \tag{4.45}$$

The centralized version of S is:

$$\begin{aligned} s^c_{ij} &= -\frac{1}{2}\left[\left(r_{ij} - s_{ii} - s_{jj}\right) - \frac{1}{n}\sum_{h=1}^{n}\left(r_{hj} - s_{hh} - s_{jj}\right) - \frac{1}{n}\sum_{k=1}^{n}\left(r_{ik} - s_{ii} - s_{kk}\right)\right. \\ &\quad \left. + \frac{1}{n^2}\sum_{h=1}^{n}\sum_{k=1}^{n}\left(r_{hk} - s_{hh} - s_{kk}\right)\right] \qquad (4.46) \\ &= -\frac{1}{2}\left(r_{ij} - \frac{1}{n}\sum_{h=1}^{n} r_{hj} - \frac{1}{n}\sum_{k=1}^{n} r_{ik} + \frac{1}{n^2}\sum_{h=1}^{n}\sum_{k=1}^{n} r_{hk}\right) \qquad (4.47) \end{aligned}$$

This proves that the centralized version of S is uniquely determined by the centralized version of R:

$$S^c = -\frac{1}{2}R^c \tag{4.48}$$

4.5.2 Proof of Theorem 4.1.1

In this section we provide the proof that R is a squared Euclidean distance matrix $\Longleftrightarrow$ $S^c \succeq 0$. Let's start with $\Rightarrow$. The centralized version of R is:

$$R^c = QRQ = R - \frac{1}{n}ee^T R - \frac{1}{n}Ree^T + \frac{1}{n^2}ee^T Ree^T \tag{4.49}$$

Assuming that a set of vectors $\mathbf{x}$ exists, for which:

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{4.50}$$

the elements of R^c can be written:

$$\begin{aligned} r^c_{ij} &= \|\mathbf{x}_i - \mathbf{x}_j\|^2 - \frac{1}{n}\sum_{h=1}^{n}\|\mathbf{x}_h - \mathbf{x}_j\|^2 - \frac{1}{n}\sum_{k=1}^{n}\|\mathbf{x}_i - \mathbf{x}_k\|^2 + \frac{1}{n^2}\sum_{h=1}^{n}\sum_{k=1}^{n}\|\mathbf{x}_h - \mathbf{x}_k\|^2 \\ &= \mathbf{x}_i\mathbf{x}_i + \mathbf{x}_j\mathbf{x}_j - 2\mathbf{x}_i\mathbf{x}_j - \frac{1}{n}\left(\sum_{h=1}^{n}\mathbf{x}_h\mathbf{x}_h + \mathbf{x}_j\mathbf{x}_j - 2\mathbf{x}_h\mathbf{x}_j\right) - \frac{1}{n}\left(\sum_{k=1}^{n}\mathbf{x}_i\mathbf{x}_i + \mathbf{x}_k\mathbf{x}_k - 2\mathbf{x}_i\mathbf{x}_k\right) \\ &\quad + \frac{1}{n^2}\left(\sum_{h=1}^{n}\sum_{k=1}^{n}\mathbf{x}_h\mathbf{x}_h + \mathbf{x}_k\mathbf{x}_k - 2\mathbf{x}_h\mathbf{x}_k\right) \\ &= -2\left(\mathbf{x}_i\mathbf{x}_j - \frac{1}{n}\sum_{h=1}^{n}\mathbf{x}_h\mathbf{x}_j - \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_i\mathbf{x}_k + \frac{1}{n^2}\sum_{h=1}^{n}\sum_{k=1}^{n}\mathbf{x}_h\mathbf{x}_k\right) \end{aligned} \tag{4.51}$$

Introducing the quantity:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{h=1}^{n}\mathbf{x}_h \tag{4.52}$$

we can rewrite in a more compact way Eq. 4.51:

$$r^c_{ij} = -2(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_j - \bar{\mathbf{x}}) = -2\breve{\mathbf{x}}_i\breve{\mathbf{x}}_j \tag{4.53}$$

This is equivalent to say that:

$$S^c = \breve{X}\breve{X}^T \tag{4.54}$$

which proves $\Rightarrow$.

To prove $\Leftarrow$, since S^c is positive semidefinite, we can write:

$$S^c = XX^T \tag{4.55}$$

where the rows of X are vectors $\mathbf{x} \in \mathbb{R}^d$. From Eq. 4.5:

$$\begin{aligned} r_{ij} &= s_{ii} + s_{jj} - 2s_{ij} \\ &= \mathbf{x}_i\mathbf{x}_i + \mathbf{x}_j\mathbf{x}_j - 2\mathbf{x}_i\mathbf{x}_j \\ &= \|\mathbf{x}_i - \mathbf{x}_j\|^2 \end{aligned} \tag{4.56}$$

This proves $\Leftarrow$.

4.5.3 Preshift and Postshift

Let's analyze why:

$$S^c + \alpha I \neq -\frac{1}{2}(Q\tilde{R}Q) \tag{4.57}$$

and how this can influence the behavior of the studied clustering algorithms. First, let's see what is the difference between the resulting matrices. For the preshift we have:

$$-\frac{1}{2}(Q\tilde{R}Q) = -\frac{1}{2}(QRQ) - \alpha Q(ee^T - I)Q = S^c - \alpha Q(ee^T - I)Q \tag{4.58}$$

Now:

$$Q(ee^T - I)Q = Qee^TQ - QQ = -QQ = -Q \tag{4.59}$$

since:

$$Qe = (I - \frac{1}{n}ee^T)e = e - e = \mathbf{0} \tag{4.60}$$

and:

$$QQ = (I - \frac{1}{n}ee^T)(I - \frac{1}{n}ee^T) = I - \frac{2}{n}ee^T + \frac{1}{n^2}ee^Tee^T = I - \frac{1}{n}ee^T = Q \tag{4.61}$$

Thus:

$$-\frac{1}{2}(Q\tilde{R}Q) = S^c + \alpha Q \tag{4.62}$$

The difference between the matrices associated to postshift and preshift is:

$$\alpha(I - Q) = \frac{\alpha}{n}ee^T \tag{4.63}$$

Now we prove that $\|\mathbf{x}_h - \mathbf{v}_j\|^2$ is independent from the choice of the preshift or postshift:

$$\|\mathbf{x}_h - \mathbf{v}_j\|^2 = k'_{hh} - 2\frac{\sum_{r=1}^n u_{ir}^\theta k'_{rh}}{\sum_{r=1}^n u_{ir}^\theta} + \frac{\sum_{r=1}^n \sum_{s=1}^n u_{ir}^\theta u_{is}^\theta k'_{rs}}{(\sum_{r=1}^n u_{ir}^\theta)^2} \tag{4.64}$$

$$k' = k + \frac{\alpha}{n} \tag{4.65}$$

$$\|\mathbf{x}_h - \mathbf{v}_j\|^2 = k_{hh} + \frac{\alpha}{n} - 2\frac{\sum_{r=1}^n u_{ir}^\theta k_{rh}}{\sum_{r=1}^n u_{ir}^\theta} - 2\frac{\alpha}{n} + \frac{\sum_{r=1}^n \sum_{s=1}^n u_{ir}^\theta u_{is}^\theta k_{rs}}{(\sum_{r=1}^n u_{ir}^\theta)^2} + \frac{\alpha}{n} \tag{4.66}$$

4.5.4 Proof of Equivalence between $G(U,V)$ and $G(U)$

To prove the equivalence between the distortion functions in Eqs. 4.15 and 4.20, let's introduce the quantity:

$$b_i = \sum_{h=1}^{n} u_{ih}^{\theta} \tag{4.67}$$

Since:

$$\mathbf{v}_i = \frac{\sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{x}_h}{\sum_{h=1}^{n} u_{ih}^{\theta}} \tag{4.68}$$

part of the sum in $G(U,V)$ can be rewritten in the following way:

$$\begin{aligned}
\sum_{h=1}^{n} u_{ih}^{\theta} \|\mathbf{x}_h - \mathbf{v}_i\|^2 &= \sum_{h=1}^{n} u_{ih}^{\theta} (\mathbf{x}_h - \mathbf{v}_i)(\mathbf{x}_h - \mathbf{v}_i) \\
&= \sum_{h=1}^{n} u_{ih}^{\theta} (\mathbf{x}_h \mathbf{x}_h + \mathbf{v}_i \mathbf{v}_i - 2\mathbf{x}_h \mathbf{v}_i) \\
&= \sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{x}_h \mathbf{x}_h + \sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{v}_i \mathbf{v}_i - 2 \sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{x}_h \mathbf{v}_i \\
&= \sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{x}_h \mathbf{x}_h + b_i \mathbf{v}_i \mathbf{v}_i - 2 b_i \mathbf{v}_i \mathbf{v}_i \\
&= \sum_{h=1}^{n} u_{ih}^{\theta} \mathbf{x}_h \mathbf{x}_h - b_i \mathbf{v}_i \mathbf{v}_i
\end{aligned} \tag{4.69}$$

Rewriting part of $G(U)$, we obtain:

$$\begin{aligned}
\sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} \|\mathbf{x}_r - \mathbf{x}_s\|^2 &= \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} (\mathbf{x}_r - \mathbf{x}_s)(\mathbf{x}_r - \mathbf{x}_s) \\
&= \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} (\mathbf{x}_r \mathbf{x}_r + \mathbf{x}_s \mathbf{x}_s - 2\mathbf{x}_r \mathbf{x}_s) \\
&= \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} \mathbf{x}_r \mathbf{x}_r + \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} \mathbf{x}_s \mathbf{x}_s - 2 \sum_{r=1}^{n} \sum_{s=1}^{n} u_{ir}^{\theta} u_{is}^{\theta} \mathbf{x}_r \mathbf{x}_s \\
&= \sum_{r=1}^{n} u_{ir}^{\theta} \mathbf{x}_r \mathbf{x}_r \sum_{s=1}^{n} u_{is}^{\theta} + \sum_{s=1}^{n} u_{is}^{\theta} \mathbf{x}_s \mathbf{x}_s \sum_{r=1}^{n} u_{ir}^{\theta} - 2 \sum_{r=1}^{n} \mathbf{x}_r u_{ir}^{\theta} \sum_{s=1}^{n} u_{is}^{\theta} \mathbf{x}_s \\
&= 2 b_i \sum_{r=1}^{n} u_{ir}^{\theta} \mathbf{x}_r \mathbf{x}_r - 2 b_i^2 \mathbf{v}_i \mathbf{v}_i
\end{aligned} \tag{4.70}$$

This proves that $G(U,V) = G(U)$.

Chapter 5

One-Cluster Possibilistic c-means in Feature Space

This Chapter presents the Possibilistic c-means algorithm in feature space, that is one of the novel contributions of this thesis. It is a clustering algorithms, but it can also be used for outlier detection, and for density estimation. The algorithm has been published in Ref. [FMR07]; in this thesis, we extend the paper by reporting more theoretical and experimental validations. In particular, we study the connections between One Cluster Possibilistic c-means (OCPCM) and One-Class SVM (OCSVM). Section 5.1 introduces the algorithm, in Section 5.2 we study the connections of the proposed model with One Class SVM, and in Section 5.3 we report the experimental analysis.

5.1 Possibilistic Clustering in Feature Space

In this Section, we propose the possibilistic approach to clustering in kernel induced spaces. The main drawback for the possibilistic c-means, as well as for most central clustering methods, is its inability to model in a non-parametric way the density of clusters of generic shape (parametric approaches such as Possibilistic C-Spherical Shells [KK93], instead, have been proposed for some classes of shapes). In order to overcome this limit, we propose the Possibilistic c-Means in Feature Space algorithm, in particular the PCM II in feature space (PCM-II-fs). It consists in the application of the PCM II applied in the feature space $\mathcal{F}$ obtained by a mapping Φ from the input space S ($\Phi : S \to \mathcal{F}$). The objective function to minimize is then:

$$J^{\Phi}(U, V^{\Phi}) = \sum_{h=1}^{n}\sum_{i=1}^{c} u_{ih}\|\Phi(\mathbf{x}_h) - \mathbf{v}_i^{\Phi}\|^2 + \sum_{i=1}^{c}\eta_i \sum_{h=1}^{n}(u_{ih}\ln(u_{ih}) - u_{ih})\,. \qquad (5.1)$$

Note that the centroids $\mathbf{v}_i^\Phi$ of PCM-II-fs algorithm lie in the feature space. We can minimize $J^\Phi(U, V^\Phi)$ by setting its derivatives with respect to $\mathbf{v}_i^\Phi$ and u_{ih} equal to zero, obtaining:

$$\mathbf{v}_i^\Phi = \frac{\sum_{h=1}^{n} u_{ih}\Phi(\mathbf{x}_h)}{\sum_{h=1}^{n} u_{ih}} = b_i \sum_{h=1}^{n} u_{ih}\Phi(\mathbf{x}_h), \qquad b_i \equiv \left(\sum_{h=1}^{n} u_{ih}\right)^{-1} \tag{5.2}$$

$$u_{ih} = \exp\left(-\frac{\|\Phi(\mathbf{x}_h) - \mathbf{v}_i^\Phi\|^2}{\eta_i}\right). \tag{5.3}$$

In principle, the necessary conditions in Eq.s 5.2 and 5.3 can be used for a Picard iteration minimizing $J^\Phi(U, V^\Phi)$. Since Φ is not known explicitly, we cannot compute the centers $\mathbf{v}_i^\Phi$ directly. Despite this, if we consider Mercer Kernels [Aro50] (symmetric and semidefinite kernels) which can be expressed as a scalar product:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j), \tag{5.4}$$

this relation holds (*kernel trick* [ABR64]):

$$\|\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)\|^2 = K(\mathbf{x}_i, \mathbf{x}_i) + K(\mathbf{x}_j, \mathbf{x}_j) - 2K(\mathbf{x}_i, \mathbf{x}_j). \tag{5.5}$$

This allows us to obtain an update rule for the memberships by substituting Eq. 5.2 in Eq. 5.3:

$$u_{ih} = \exp\left[-\frac{1}{\eta_i} \cdot \left(k_{hh} - 2b_i \sum_{r=1}^{n} u_{ir}k_{hr} + b_i^2 \sum_{r=1}^{n}\sum_{s=1}^{n} u_{ir}u_{is}k_{rs}\right)\right]. \tag{5.6}$$

Note that in Eq. 5.6 we introduced the notation $k_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. The Picard iteration then reduces to the iterative update of the memberships only using Eq. 5.6, ending when an assigned stopping criterion is satisfied (e.g., when memberships change less than an assigned threshold, or when no significant improvements of $J^\Phi(U, V^\Phi)$ are noticed).

Concerning the parameters η_i, we can apply in feature space the same criterion suggested for the PCM II obtaining:

$$\eta_i = \gamma\ b_i \sum_{h=1}^{n} u_{ih} \left(k_{hh} - 2b_i \sum_{r=1}^{n} u_{ir}k_{hr} + b_i^2 \sum_{r=1}^{n}\sum_{s=1}^{n} u_{ir}u_{is}k_{rs}\right) \tag{5.7}$$

The parameters η_i can be estimated at each iteration or once at the beginning of the algorithm. In the latter case the initialization of the memberships, that allows to provide a good estimation of the η_i, can be obtained as a result of a Kernel Fuzzy c-Means [ZC02].

Note that if we chose a linear kernel $k_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$ the PCM-II-fs reduces to the standard PCM II, i.e., using a linear kernel is equivalent to set $\Phi \equiv I$, where I is the identity function. In the following, we will use a Gaussian kernel:

$$k_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{5.8}$$

for which:

$$\|\Phi(\mathbf{x}_i)\|^2 = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) = k_{ii} = 1. \tag{5.9}$$

As a consequence, patterns are mapped by the Gaussian kernel from data space to the surface of a unit hypersphere in feature space. Centroids in the feature space $\mathbf{v}_i^{\Phi}$ are not constrained to the hyperspherical surface as mapped patterns; therefore, centroids lie inside this hypersphere, and due to the lack of competitiveness between clusters (that characterizes the possibilistic clustering framework), centroids of PCM-II-fs often collapse into a single one, with slight dependency on the value of the cluster spreads η_i.

Note that PCM-II-fs retains the principal characteristics of PCM II, including the capability of estimating hyperspherical densities, this time in the feature space. In the data space this corresponds to the capability to model clusters of more general shape, a significant improvement with respect the original PCM II.

5.2 One-Class SVM vs One Cluster PCM in kernel induced space

In this Section we study the connections between the PCM-II-fs with $c = 1$, that we will call the One Cluster PCM II in feature space (OCPCM) and One Class SVM (OCSVM). In particular, we show the formal analogies between the two objective functions, highlighting the robustness of the proposed method against OCSVM.

5.2.1 One-Class SVM

Let's recall some basic concepts about OCSVM. In particular, the optimization problem is the following:

$$\min_{\alpha_1,\ldots,\alpha_n} \left(\sum_r \sum_s \alpha_r \alpha_s k_{rs} - \sum_h \alpha_h k_{hh}\right) \qquad \text{subject to :}$$

$$\sum_h \alpha_h = 1 \qquad \text{and} \qquad 0 \leq \alpha_h \leq C$$

with respect to α_i.

- when $\xi_h > 0$, the image of $\mathbf{x}_h$ lies outside the hypersphere. These points are called *bounded support vectors*. For them, $\alpha_h = C$ holds;
- when $\xi_h = 0$ and $0 < \alpha_h < C$, the image of $\mathbf{x}_h$ lies on the surface of the hypersphere. These points are called *support vectors*.
- when $\xi_h = 0$ and $\alpha_h = 0$, the image of $\mathbf{x}_h$ is inside the hypersphere.

The computation of the center of the sphere:

$$\mathbf{v} = \sum_h \alpha_h \Phi(\mathbf{x}_h) \tag{5.10}$$

leads to the computation of the distance between a pattern and the center:

$$d_h = \|\Phi(\mathbf{x}_h) - \mathbf{v}\|^2 = k_{hh} - 2\sum_r \alpha_r k_{hr} + \sum_r \sum_s \alpha_r \alpha_s k_{rs} \tag{5.11}$$

5.2.2 One-Cluster PCM in Feature Space

We show now the objective function of OCPCM, starting from a formulation in input space. Let's assume the presence of a single cluster, i.e., we consider PCM-II-fs with $c = 1$. We represent the memberships as a vector $\mathbf{u}$, where each u_h is the membership of the h-th pattern to the cluster.

The objective function of OCPCM is:

$$L = \sum_h u_h \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \sum_h (u_h \ln(u_h) - u_h) \tag{5.12}$$

The possibilistic constraint on the memberships is the following:

$$0 \leq u_h \leq 1 \tag{5.13}$$

Setting to zero the derivatives of L with respect to $\mathbf{v}_i$:

$$\frac{\partial L}{\partial \mathbf{v}} = -\sum_{h=1}^{n} u_{ih} (\mathbf{x}_h - \mathbf{v}_i) = 0 \tag{5.14}$$

we obtain the update formula for the centroid $\mathbf{v}$:

$$\mathbf{v} = \frac{\sum_h u_h \mathbf{x}_h}{\sum_h u_h} \tag{5.15}$$

Substituting $\mathbf{v}$ in L:

$$\begin{aligned} L &= \sum_h u_h \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \sum_h (u_h \ln(u_h) - u_h) \\ &= \sum_h u_h \mathbf{x}_h \mathbf{x}_h + \mathbf{v}\mathbf{v} \sum_h u_h - 2\mathbf{v} \sum_h u_h \mathbf{x}_h + \eta \sum_h (u_h \ln(u_h) - u_h) \\ &= \sum_h u_h \mathbf{x}_h \mathbf{x}_h - \mathbf{v}\mathbf{v} \sum_h u_h + \eta \sum_h (u_h \ln(u_h) - u_h) \\ &= \sum_h u_h \mathbf{x}_h \mathbf{x}_h - \frac{\sum_r \sum_s u_r u_s \mathbf{x}_r \mathbf{x}_s}{\sum_h u_h} + \eta \sum_h (u_h \ln(u_h) - u_h) \end{aligned}$$

The last equation can be extended by mean of positive semidefinite kernels, leading to the following optimization problem:

$$\min \left(\sum_h u_h k_{hh} - \frac{\sum_r \sum_s u_r u_s k_{rs}}{\sum_h u_h} + \eta \sum_h (u_h \ln(u_h) - u_h) \right) \qquad \text{subject to :}$$

$$0 \leq u_k \leq 1$$

With this extension, the proposed algorithm models all data points in a single cluster in features space. If we add the constraint $\sum_h u_h = 1$, the problem becomes the following:

$$\min \left(\sum_h u_h k_{hh} - \sum_r \sum_s u_r u_s k_{rs} + \eta \sum_h u_h \ln(u_h) \right) \qquad \text{subject to :}$$

$$0 \leq u_h \leq 1 \qquad \text{and} \qquad \sum_h u_h = 1$$

This is the optimization problem of the OCPCM; in the next sub-section we will study the optimization procedure. In the next Section, we will see that the constraint on the sum of the memberships corresponds just to scale the values of the memberships, and that the position of the centroid is not affected by that.

The result just obtained shows that the objective function of the One-Cluster PCM-FS has a close relationship with that of One-Class SVM. In particular, the role of the α_h is the dual with respect to the u_h. In One-Class SVM, the center of the sphere is computed as combination of outliers; in One-Cluster PCM-FS, the patterns contribute to the position of the centroids proportionally to their memberships, that is very low for the outliers. This can lead to a more reliable estimation for the centroid $\mathbf{v}$ in One-Cluster PCM-FS. Moreover, in One-Cluster PCM-FS we can see the presence of a regularization term, which is an entropy based score of the memberships. In the experimental analysis, we will show the implication of these facts.

We note that the algorithms we are comparing follow different approaches. One-Class SVM looks for the center $\mathbf{v}$ and the radius R of the enclosing sphere, One-Cluster PCM-FS looks for a centroid in feature space and computes the memberships on the basis of $\mathbf{v}$. The parameter η works as the width of the membership function, and corresponds to the square of the radius R^2. One-Cluster PCM-FS yields the memberships of the patterns, and it is possible to set a threshold to obtain a decision boundary. This corresponds to select a sphere in feature space that is the intersection between the multi-dimensional Gaussian describing the memberships and an hyperplane.

5.2.2.1 Optimization Algorithm - The Unconstrained Case

Let's analyze the procedure to optimize the Lagrangian:

$$L = \sum_h u_h \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \sum_h (u_h \ln(u_h) - u_h) \tag{5.16}$$

The optimization technique that we use is the so called Picard iteration technique. L depends on two groups of variables $\mathbf{u}$ and V related to each other, namely $\mathbf{u} = \mathbf{u}(\mathbf{v})$ and $\mathbf{v} = \mathbf{v}(\mathbf{u})$. In each iteration one of the two groups of variables is kept fixed, and the minimization is performed with respect to the other group. The update equation can be obtained setting the derivatives of L to zero:

$$\frac{\partial L}{\partial \mathbf{v}} = 0 \tag{5.17}$$

$$\frac{\partial L}{\partial u_h} = 0 \tag{5.18}$$

We iterate these two equations until convergence:

$$u_h = \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}\|^2}{\eta}\right) \tag{5.19}$$

$$\mathbf{v} = \frac{\sum_{h=1}^{n} u_h \mathbf{x}_h}{\sum_{h=1}^{n} u_h} \tag{5.20}$$

The constraint $0 \leq u_k \leq 1$ is satisfied since the form assumed by their update equation.

5.2.2.2 Optimization Algorithm - The Constrained Case

We show that constraining the sum of the memberships does not affect the behavior of the optimization procedure. In other words, the results of the constrained and unconstrained case differ only in the scaling factor of the memberships. Let's start with the Lagrangian:

$$L = \sum_h u_h \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \sum_h (u_h \ln(u_h) - u_h) \tag{5.21}$$

subject to:

$$\sum_h u_h = 1 \tag{5.22}$$

$$L = \sum_h u_h \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \sum_h (u_h \ln(u_h) - u_h) + \gamma \left(\sum_h u_h - 1 \right) \tag{5.23}$$

$$\frac{\partial L}{\partial u_h} = \|\mathbf{x}_h - \mathbf{v}\|^2 + \eta \ln(u_h) + \gamma = 0 \tag{5.24}$$

$$u_h = \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}\|^2}{\eta} \right) \exp\left(-\frac{\gamma}{\eta} \right) \tag{5.25}$$

Substituting in the constraint:

$$\sum_h \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}\|^2}{\eta} \right) \exp\left(-\frac{\gamma}{\eta} \right) = 1 \tag{5.26}$$

gives:

$$\gamma = \eta \ln\left(\sum_h \exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}\|^2}{\eta} \right) \right) \tag{5.27}$$

Finally:

$$u_h = \frac{\exp\left(-\frac{\|\mathbf{x}_h - \mathbf{v}\|^2}{\eta} \right)}{\sum_r \exp\left(-\frac{\|\mathbf{x}_r - \mathbf{v}\|^2}{\eta} \right)} \tag{5.28}$$

The update of $\mathbf{v}$ is the same as in the unconstrained case. The normalization in Eq. 5.28 cancels out in the computation of the $\mathbf{v}$. This means that starting with the same memberships, the constrained and unconstrained case give the same $\mathbf{v}$, and the memberships are only scaled to sum up to one.

5.2.3 Applications of OCPCM

The *Core* step produces a fuzzy-possibilistic model of densities (membership function) in the feature space. It is initialized by selecting a *stop criterion* (e.g., when memberships change less than an assigned threshold, or when no significant improvements of $J^{\Phi}(U, V^{\Phi})$ are noticed), setting the value of σ for the Gaussian kernel (in order to define the spatial resolution of density estimation), and initializing the memberships u_h (usually as $u_h = 1$). Then, after estimating the value of η using Eq. 5.7, we perform the Picard iteration using Eq. 5.6.

Density Estimation At the end of the *Core* step, we have modeled the density of patterns in feature space. These memberships, back to the input space, represent a density estimation in input space based on a specific kernel. In this framework, the value of the parameter η plays the role of a scaling factor on the range of density values that can be obtained by the algorithm.

Outlier Detection Once the memberships are obtained, it is possible to select a threshold $\alpha \in (0, 1)$ and use it to define an *α-cut* (or *α-level set*) on data points:

$$A_\alpha = \{\mathbf{x}_h \in X \mid u_h > \alpha\} \tag{5.29}$$

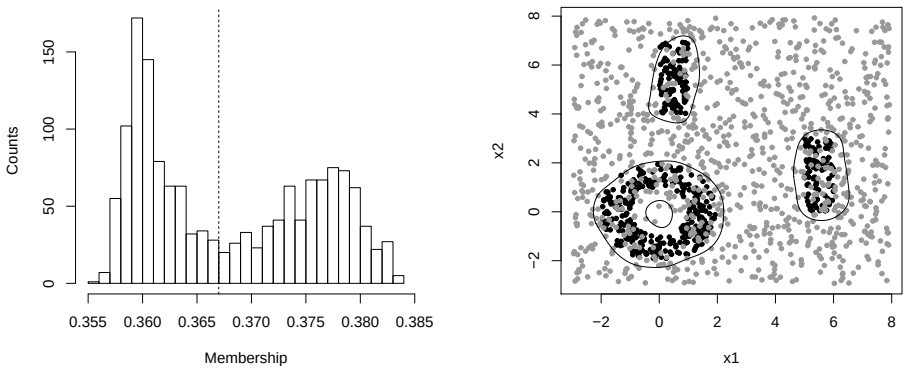
This can be considered as a *Defuzzification* step. Note that given the form of u_h (Eq. 5.3) the threshold α defines a hypercircle which encloses a hyperspherical cap. A_α is then the set of data points whose mapping in feature space lies on the cap, whose base radius depends on α. Points outside the α-cut are considered to be outliers.

Clustering The *Labeling* step separates the data points belonging to the single cluster in feature space, in a number of "*natural*" clusters in data space. It uses a convexity criterion derived from the one proposed for One-Class SVM [HHSV01] assigning the same label to a pair of points only if all elements of the linear segment joining the two points in data space belong to A_α.

The *Defuzzification* and *Labeling* steps can be iterated with different values of α, thus performing a very lightweight *model selection*, without involving new runs of the *Core* step. Often, such as in the case of experiments presented in next section, an a-priori analysis of the memberships histogram permits to obtain a good evaluation of α without performing a true model selection. Indeed, the presence of multiple modes in the membership histogram indicates the presence of different structures in feature space, and allows to find several levels of α discriminating the different densities of data in feature space.

5.3 Experimental Analysis

In this Section, we report the experimental analysis showing the properties of OCPCM. In the first part, we show its ability to model densities in feature space and to perform clustering. In the second part, we present a comparison of OCSVM and OCPCM as outlier detection algorithms, by means of a stability validation test.



(a) (b)

Figure 5.1: (a) Histogram of the memberships obtained by OCPCM with $\sigma = 0.5$. The dotted line gets through to membership value .3667 that separates the two modes of the graph; the value of α is then taken as $\alpha = .3667$. (b) Data space: black dots belong to the dense regions and the Gray ones are the noisy patterns. The contours correspond to points with membership equal to .3667.

5.3.1 Density Estimation and Clustering

We present some results obtained on a synthetic set (Fig. 5.1) composed by three disjoint dense regions (black dots) on a 10x10 square: two rectangular regions, each of them corresponding to 1.24 % of the square and composed by 100 patterns uniformly distributed, and a ring shaped region, corresponding to 7.79 % of the square, that contains 300 patterns uniformly distributed. An uniformly distributed noise of 1000 gray dots is superimposed to the square.

We used a Gaussian kernel with standard deviation $\sigma = 0.5$ estimated as the order of magnitude of the average inter-data points distance. The memberships u_h were initialized to 1. The stop criterion was $\sum_h \Delta u_h < \varepsilon$ with $\varepsilon = 0.01$.

In the *Defuzzification* step we evaluated α using the histogram method. As shown in Fig. 5.1(a), choosing $\alpha = .3667$ that is the value of membership separating the two modes of the histogram, we obtain a good separating surface in the data space (Fig. 5.1(b)), with no need to perform any iteration for model selection.

As shown in the experiment, OCPCM shows a high robustness to outliers and a very good capability to model clusters of generic shape in the data space (modeling their distributions in terms of fuzzy memberships). Moreover, it is able to find *autonomously* the *natural*

number of clusters in the data space. The outliers rejection ability is shared also by the PCM II, but is limited to the case of globular clusters.

In all the runs of One-Cluster PCM-FS the *Core* step, which involves the minimization of $J^{\Phi}(U, V^{\Phi})$ (Eq. 5.1), resulted to be very fast, since less than a tenth of iterations of Eq. 5.6 where enough.

5.3.2 Stability Validation for Outlier Detection

The method that we will use to compare the stability of OCSVM and OCPCM solutions for outlier detection is the one proposed in Ref. [LRBB04]. This approach has been used to estimate the natural number of clusters in a data set. We report the general ideas and we will detail how we intend to use the method. In particular, we will apply a modified version of it making use of the Jaccard coefficient, to compare the ability of the two algorithms to identify the outliers.

The general procedure starts by splitting the original data set in two disjoint subsets X and X'. The cardinality of X and X' is half the cardinality of the original data set. By applying a clustering algorithm to X we obtain the cluster labels $\mathbf{z} = (z_1, z_2, \ldots, z_{n/2})$. On the basis of the clustering model build on X it is possible to assign the cluster labels to X'. This mechanism is called *Transfer by Prediction* and can be formalized by a classifier ϕ trained on X that allows to predict the labels of X'. In our case, the classifier will be the one that labels a pattern as outlier when it is outside the hypersphere in feature space for OCSVM and when it has a membership lower than the selected threshold in OCPCM. On the other hand, we can directly apply the clustering algorithm to X' obtaining a set of labels $\mathbf{z}' = (z'_1, z'_2, \ldots, z'_{n/2})$. The labels $\phi(X')$ and $\mathbf{z}'$ can now be compared using, for instance, the Hamming distance. Such distance must take into account the possible permutations of the cluster labels. Indeed, the $\phi(X')$ and $\mathbf{z}'$ are not necessarily in a direct correspondence. For that reason, the minimum distance over all the permutations has to be taken. The expected value of this distance can be considered as a stability measure of the clustering solution. The computation is made by averaging the distance over a finite number of resamplings. The number of clusters k clearly affects the bounds on this score; this suggests a normalization based on the expected value of the Hamming distance between two random clustering labelings.

To compare OCSVM and OCPCM for outlier detection, we propose a matching based on the Jaccard coefficient. For two binary variables, Jaccard coefficient is a measure of their concordance on positive responses. Given the confusion matrix:

Table 5.1: Pseudocode of the stability validation procedure for outlier detection

1. Repeat r times:
   (a) split the given data set into two halves X and X';
   (b) apply the outlier detection algorithm to X and predict the labels on X' obtaining $\phi(X')$;
   (c) apply the outlier detection algorithm to X' obtaining $\mathbf{z}'$;
   (d) compute the Jaccard coefficient between $\phi(X')$ and $\mathbf{z}'$;

		B	
		0	1
A	0	a_{00}	a_{01}
	1	a_{10}	a_{11}

Jaccard coefficient is defined as:

$$J = \frac{a_{11}}{a_{01} + a_{10} + a_{11}} \tag{5.30}$$

It is clear the difference with the simple matching, where we would have taken into account also the occurrences of negative responses. In some applications, the value of positive and negative responses do not have equal information (asymmetry). For example, if the negative value is not important, counting the non-existence in both the variables may have no meaningful contribution to the similarity or dissimilarity. In our case, the use of such coefficient is particularly suitable, since we want to measure the concordance between the solutions $\phi(X')$ and $\mathbf{z}'$ on the identification of outliers. We want to give more importance to the fact that $\phi(X')$ and $\mathbf{z}'$ match on the outliers, instead of normal patterns. The use of the normalization of this score is not needed since we are dealing with two classes (outlier vs. non-outliers) in both the algorithms. The steps of the stability validation procedure for outlier detection are outlined in Tab. 5.1.

We decided to evaluate the stability for different values of ν in OCSVM. Different values of ν lead to different numbers of outliers. For this reason, to compare correctly OCSVM with OCPCM for different values of ν, we decided to set a threshold in the memberships obtained by OCPCM, in order to reject exactly the same number of patterns rejected by OCSVM with that particular value of ν.

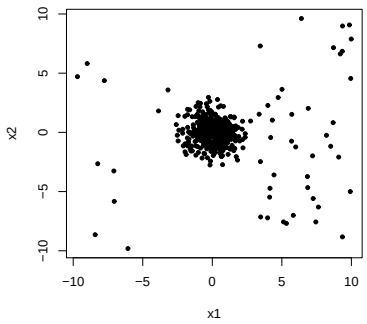


Figure 5.2: Synthetic data set.

5.3.3 Results

5.3.3.1 Synthetic data set

The synthetic data set used in our experiments is shown in Fig. 5.2. It is a two-dimensional data set composed by 400 points. They have been generated using a Gaussian distribution centered in $(0, 0)$ having unitary variance along the two axes. Other 20 points have been added sampling uniformly the set $[3, 10] \times [-10, 10]$ and 10 points sampling uniformly the set $[-10, -3] \times [-10, 10]$ to obtain a non-symmetric outlier distribution.

We tested the stability of OCSVM and OCPCM for outlier detection using the algorithm presented in Tab. 5.1. We used a Gaussian kernel with three different values of σ: 0.1, 1, and 10; the regularization parameter η has been set to two different values: 1 and 10. The results are summarized in Fig. 5.3, where the box-and-whisker plot of the Jaccard coefficient over 500 repetitions ($r = 500$) for different values on ν. The median is denoted by the thick line while the box contains the values between the first and third quartile. The wiskers extend to 1.5 times the range of the difference between the first and third quartile. In the plots, we omitted the values of the Jaccard coefficient exceeding the wiskers. The three rows of Fig. 5.3 correspond to the three different values of σ. The first plot in each row refers to OCSVM, the other two represent those of OCPCM with two different values of the regularization parameter $\eta = 1$ and $\eta = 10$.

5.3.3.2 Real data sets

We compared the stability of OCSVM and OCPCM for outlier detection on three real data sets: Breast, Glass, and Ionosphere (see Chapter 3 for a description of these data sets).

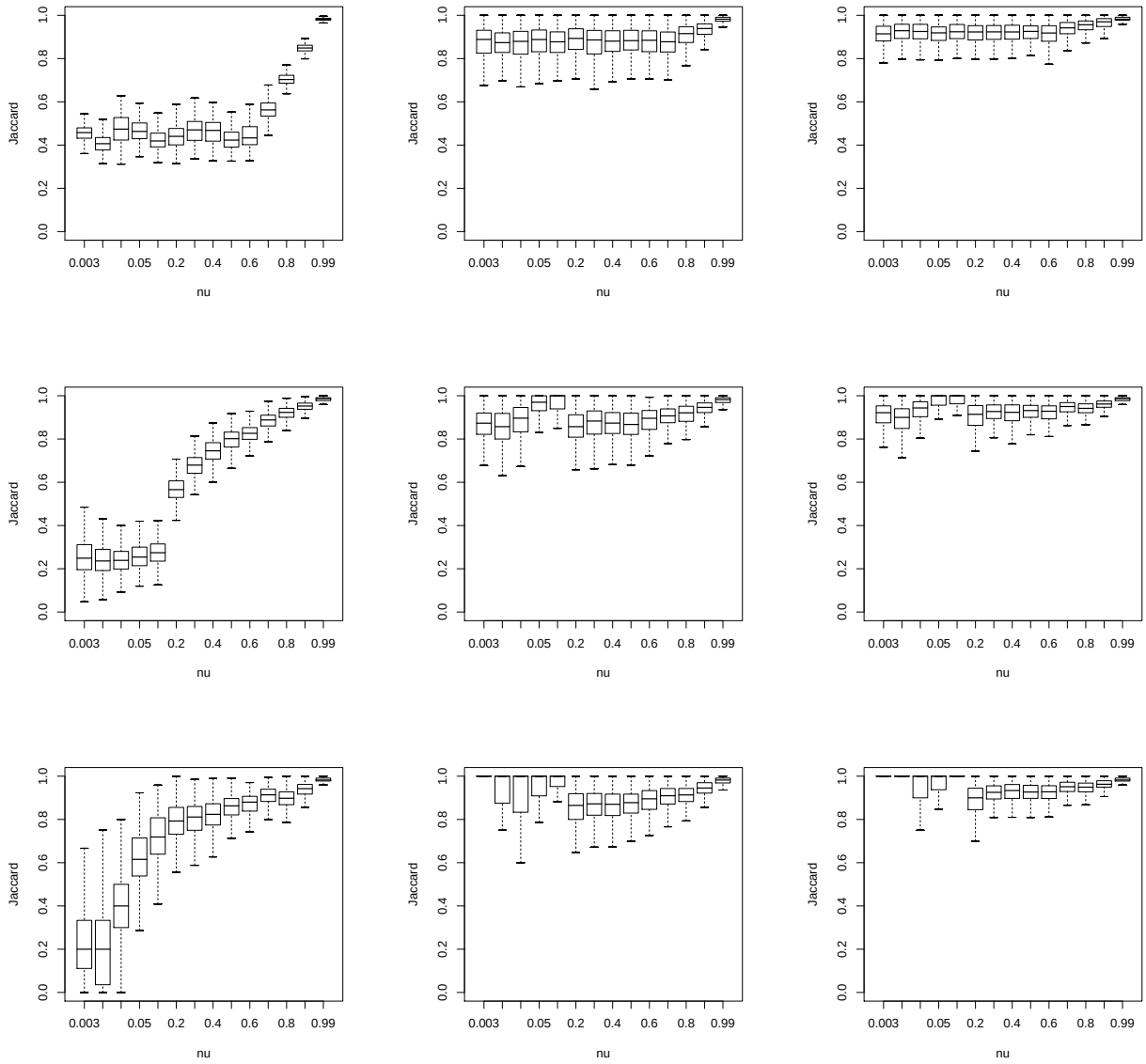


Figure 5.3: Comparison of OCSVM and OCPCM (both with Gaussian kernel) using box-and-whisker plot of the Jaccard coefficient over 500 repetitions. First row: $\sigma = 0.1$ OCSVM, OCPCM $\eta = 1$, and OCPCM $\eta = 10$. Second row: $\sigma = 1$ OCSVM, OCPCM $\eta = 1$, and OCPCM $\eta = 10$. Third row: $\sigma = 10$ OCSVM, OCPCM $\eta = 1$, and OCPCM $\eta = 10$.

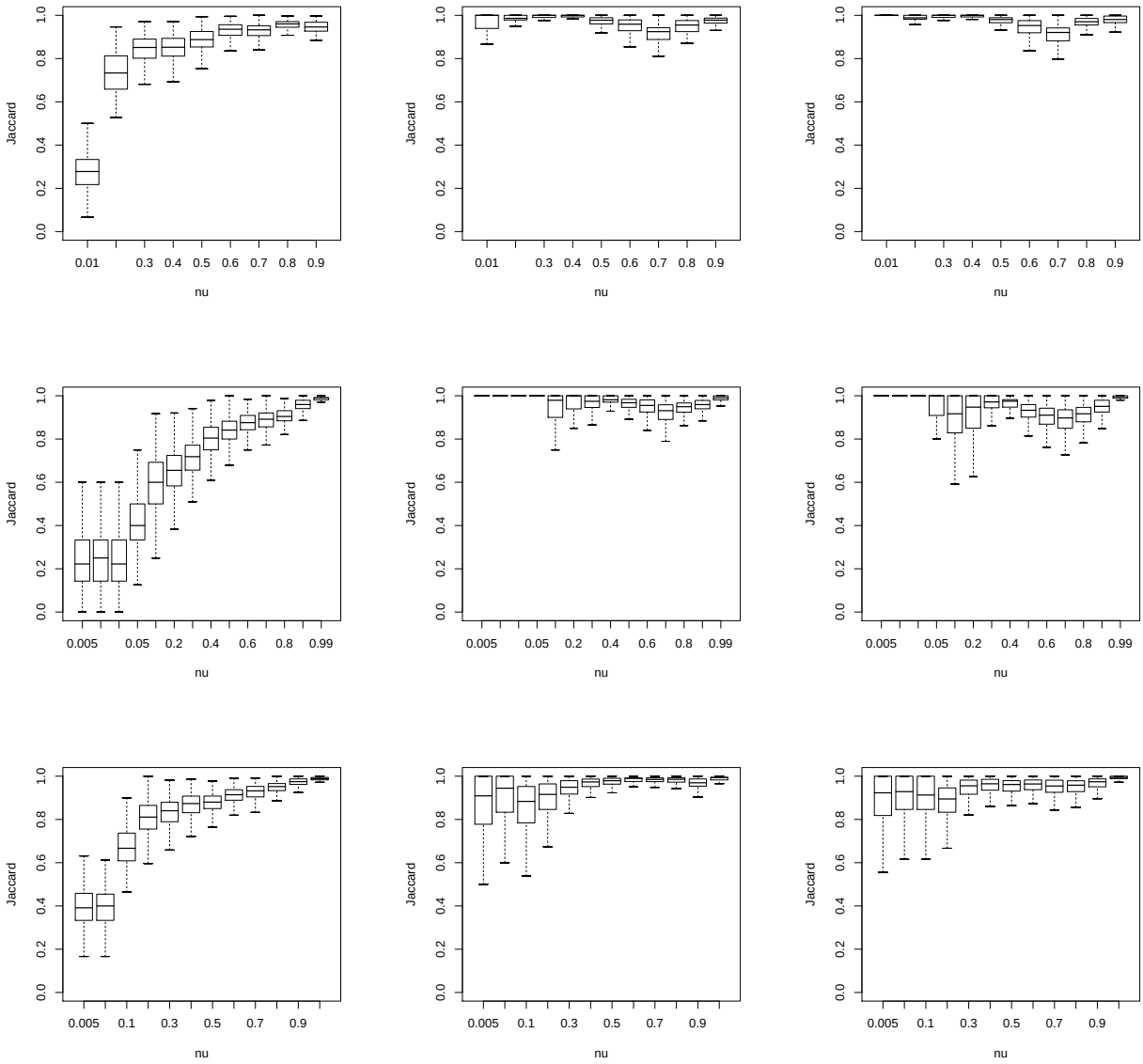


Figure 5.4: Comparison of OCSVM and OCPCM (both with Gaussian kernel) using box-and-whisker plot of the Jaccard coefficient over 500 repetitions. First row: Breast $\sigma = 10$ OCSVM, OCPCM $\eta = 10$, and OCPCM $\eta = 20$. Second row: Glass $\sigma = 5$ OCSVM, OCPCM $\eta = 1$, and OCPCM $\eta = 10$. Third row: Ionosphere $\sigma = 5$ OCSVM, OCPCM $\eta = 1$, and OCPCM $\eta = 10$.

For the Breast data set, we used a Gaussian kernel with $\sigma = 10$; for Glass and Ionosphere we used a Gaussian kernel with $\sigma = 5$. Fig. 5.4 shows the box-and-whisker plot of the Jaccard coefficient over 500 repetitions for different values on ν. The three rows of Fig. 5.4 refer respectively to the Breast, Glass, and Ionosphere data sets. The first plot in each row refers to OCSVM, the other two represent those of OCPCM with two different values of the regularization parameter η.

5.4 Discussions

In this Chapter, we introduced the possibilistic clustering in feature space. The analogies between OCPCM and OCSVM have been studied. In particular, we showed that the role of the Lagrange multipliers in OCSVM is dual with respect to the memberships in OCPCM. This fact plays a crucial role in the application of these algorithms. In OCSVM, the Lagrange multipliers α_h are zero for patterns inside the sphere and non-zero for the outliers. The memberships in OCPCM are high for patterns in dense areas and are low for outliers. The center of the sphere in OCSVM and the Gaussian in OCPCM are computed as the weighted sum of the patterns, and the weights are respectively the Lagrange multipliers and the memberships. This leads to an estimation of the center of the sphere for OCSVM as a weighted sum of the outliers. The estimation of the center of the Gaussian is thus more reliable for OCPCM. Moreover, OCPCM objective function contains a regularization term that is an entropy based score computed on the memberships. This gives to OCPCM the ability to avoid overfitting.

All these considerations are fully confirmed by the tests conducted on synthetic and real data sets (Figs. 5.3 and 5.4). Especially for small values of ν, that correspond to low rejection of outliers, the stability of OCPCM is very high with respect to that of OCSVM. In OCPCM, the selection of the regularization parameter is not critical, and the stability is achieved for η ranging in a wide range of values. Moreover, the optimization procedure is iterative and very fast, since few iterations are needed. In OCSVM, it is necessary to solve a quadratic optimization problem.

Chapter 6

Conclusions

In this thesis, we proposed some advances in the use of kernel in clustering. Such advances involve both the theoretical foundation of kernel methods for clustering and their extensive application to real problems.

The theoretical contributions originated from a classification of kernel methods for clustering. We reported from literature the formal equivalence between the objective functions of the spectral clustering with the ratio association as objective function and K-means in feature space. We studied in detail the relational dual of four fuzzy central clustering algorithms, proposing theoretical studies on the applicability of these algorithms in situations where patterns are described in terms on non-metric pairwise dissimilarities. In particular, it has been studied how the symmetrization and the shift operation on the dissimilarities affect their objective function. The main results include the proof of the invariance of the objective function to symmetrization and the lack of invariance to shift operations. Moreover, the four considered clustering algorithms have been presented under a more general framework, highlighting their connections with clustering in the space induced by positive semidefinite kernels. We proposed a novel clustering algorithm, the Possibilistic c-means in feature space, studying the analogies between OCPCM and OCSVM. In particular, we showed that the role of the Lagrange multipliers in OCSVM is dual with respect to the memberships in OCPCM, and the regularized properties of OCPCM objective function. In OCSVM, the estimation of the center of the hypersphere is based on the outliers; this drawback is avoided in OCPCM. The regularization properties of OCPCM play an important role in avoiding overfitting. These facts, along with the simple optimization procedure, suggest the potentiality of OCPCM in applications.

Regarding the applications, we conducted a comparative study of several clustering methods based on kernels and spectral theory. In general, methods in feature space performed better than methods with the kernelization of the metric and SVC. The spectral algorithm

proposed by Ng et al. achieved good results in almost all the data sets. This means that a low dimensional representation, based on the eigenvectors of the Laplacian of the graph obtained on the data, is quite effective to highlight structures in data. Clustering in kernel induced spaces and spectral clustering outperformed standard clustering algorithms. This is one of the motivations that support the interest of the Machine Learning community for these recent clustering techniques. On the other hand, methods based on kernels and spectral theory require tuning the kernel or the adjacency function; good performances are often achieved only for values of the parameters ranging in a narrow interval. Such comparative study required the use of several clustering algorithms that have been implemented in these years and collected in a software package written in R language.

The experimental analysis conducted on the relational duals of four fuzzy central clustering algorithms showed that FCM II is the least sensitive to shift operations. The cluster labels obtained by defuzzifying the memberships in both FCM I and FCM II were the same as the unshifted case, even for large shifts. This suggests that FCM I and FCM II could be useful to perform the optimization stage to obtain the cluster labels, even though the value of the memberships are distorted by the shift. The possibilistic clustering algorithms are strongly affected by the shift operation due to their inability to deal with sparse data sets.

OCPCM has been tested on synthetic and real data sets. We performed a comparison with OCSVM in the context of outlier detection. Especially for small values of ν, that correspond to low rejection of outliers, the stability of OCPCM is very high with respect to that of OCSVM. The selection of the regularization parameter in OCPCM is not critical, and the stability is achieved for η ranging in a wide range of values. Moreover, the optimization procedure is iterative and very fast, since few iterations of the memberships update equation are needed; in OCSVM, instead, it is necessary to solve a quadratic optimization problem.

Many questions are still open. In the present studies we used a Gaussian kernel, that is a very common choice. Such kernel maps the given data to an infinite dimensional space, where the patterns are definitely linearly separable. This is the reason why it is used with success in classification problems. In clustering applications, however, this may not be the best choice, especially for central clustering algorithms. Indeed, they minimize an objective function that favors the clustering of hyperspherical clusters, and this may have nothing in common with the representation given by the Gaussian kernel. Despite that, this choice led to better performances with respect to the standard clustering algorithms. It would be interesting to study the impact of other kernels in the performances. In particular, some kernels able to highlight structures in data have been proposed [FRB04, KSC07], and are worth of further investigations in the context of fuzzy clustering. Also, the connection between spectral and kernel methods for clustering, is leading to an interesting research area on hybrid optimization strategies [DGK07].

The complexity of these methods is one of the limitations on their applicability to real problems. The complexity of a single iteration of central clustering algorithms in kernel

induced space is quadratic with the cardinality of the data set, while it is linear for standard central clustering algorithms. For all these iterative algorithms, we cannot take into account the number of iterations that can have a strong impact on the running time of the algorithms. Their convergence depends on the particular data set and the choice of the parameters. The computation of the eigenvectors in spectral methods is affected by the selection of the data set and parameter selection as well. The sparsity of the matrix has a big impact on the time required to solve the eigenproblem. For these reasons, it is very difficult to identify the best approach in terms of both accuracy and complexity.

Appendix A

Software Package - kernclust

The software implemented during these years has been collected in a software package called *kernclust*[1]. This package has been implemented in R language. R is a programming language and environment for statistical computing, which was developed at Bell Laboratories [R D06]. It provides a large set of tools optimized for a wide range of problems. It is based on objects such as vectors, matrices, and more complex structures (data frames, lists). There are many operators acting directly on these objects, which make computations fast and expressed in a compact way. These properties, its GNU license[2], and a generic resemblance to Matlab, have boosted its diffusion in the statistical and machine learning communities. The peculiar conventions adopted make it a straightforward task and allow even very complicated constructs to be expressed compactly. The drawback of the fact that R works as an interpreter can lead to some limitations in terms of speed. This problem may be overcome by calling external C, C++, or Fortran routines from within an R program. This is useful when parts of the code are computationally intensive and difficult to optimize in R. Moreover, an R to C compiler has been recently released[3].

The main algorithms contained in *kernclust* are the following:

clustfs Clustering in feature space;

assign_clustfs Assign new patterns to the clusters obtained by clustfs;

clustkm Clustering with the kernelization of the matric;

assign_clustkm Assign new patterns to the clusters obtained by clustkm;

[1] the package can be downloaded at ftp://ftp.disi.unige.it/person/FilipponeM/Rpackage/kernclust_1.0.tar.gz

[2] R language is available at `http://www.r-project.org/` for the most common computer platforms (Windows, Linux, Mac OS).

[3] `http://hipersoft.cs.rice.edu/rcc/`

oneclass Algorithm to find the sphere enclosing almost all data, excluding the outliers;

pathoneclass Labeling algorithm for One Class SVM;

ngjordan Ng-Jordan spectral clustering algorithm;

shimalik Shi-Malik spectral clustering algorithm;

minimalshift Obtain a positive semidefinite kernel matrix from a dissimilarity matrix.;

nmi, ce, mis Normalized Mutual Information, Conditional Entropy and Count of Misclassified Patterns.

All these algorithms have a core part written in C language. This gives to these algorithms good performances in terms of speed. R is used as a frontend, allowing the user to perform easily the analysis of the results and produce high quality plots.

For the complete package documentation, the reader is referred to the manual contained in the package itself.

Appendix B

Other Activities

During these years, my activity has been funded by these fellowships:

- From 01-03-2007 to 30-10-2007
  at Department of Information and Software Engineering - George Mason University
  4400 University Drive, Fairfax, VA 22030 - USA
  Grant: Detecting Suspicious Behavior in Reconnaissance Images
  PI, co-PI: Prof. Daniel Barbarà and Prof. Carlotta Domeniconi

- From 20-07-2006 to 30-10-2006
  at Consorzio Venezia Ricerche
  Via della Libertà 12, 30175 Marghera, Venezia - Italy
  Topic of the fellowship:
  Tide level forecasting in the lagoon of Venezia
  Tutor: Prof. Elio Canestrelli

- From 01-09-2005 to 30-04-2007
  at Department of Computer Science - University of Genova
  Via Dodecaneso 35, 16146 Genova - Italy
  Topic of the fellowship:
  Novel clustering techniques with applications in image segmentation and analysis
  Tutor: Prof. Stefano Rovetta

- From 01-06-2005 to 31-08-2005
  at Department of Computer Science and Department of Endocrinologic and Metabolic Sciences - University of Genova
  Via Dodecaneso 35, 16146 Genova - Italy
  Topic of the fellowship:

Application of advanced clustering techniques in diagnostic problems in rheumatology
Tutor: Prof. Guido Rovetta

In these projects I dealt with other machine learning problems; some other contributions, that are not mentioned in this thesis, have been proposed:

- feature selection [FMR06c, FMR06b, FMR05, FMRC06];
- biclustering [FMR$^+$06d];
- time series analysis and forecasting [CF07, CCC$^+$07];
- other clustering approaches [RMF07, FMR06a, MRF05, RMF05].

Bibliography

[ABN+99] U. Alon, N. Barkai, D. A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, June 1999.

[ABR64] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[AHJ07] Bruno Apolloni, Robert J. Howlett, and Lakhmi C. Jain, editors. *Knowledge-Based Intelligent Information and Engineering Systems, 11th International Conference, KES 2007, XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 12-14, 2007, Proceedings, Part III*, volume 4694 of *Lecture Notes in Computer Science*. Springer, 2007.

[AN07] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[Apo67] T. M. Apostol. *Calculus, 2 vols.* Wiley, 2 edition, 1967.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[BDLR+04] Y. Bengio, O. Delalleau, N. Le Roux, J. F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16(10):2197–2219, 2004.

[Bez81] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[BH03] M. Brand and K. Huang. A unifying theorem for spectral embedding and clustering. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[Bis96] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK, 1996.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[BL94] G. Beni and X. Liu. A least biased fuzzy clustering method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):954–960, 1994.

[BN03] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[BR93] J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[BVP03] Yoshua Bengio, Pascal Vincent, and Jean F. Paiement. Spectral clustering and kernel PCA are learning eigenfunctions. Technical Report 2003s-19, CIRANO, 2003.

[CCC+07] Elio Canestrelli, Paolo Canestrelli, Marco Corazza, Maurizio Filippone, Silvio Giove, and Francesco Masulli. Local learning of tide level time series using a fuzzy approach. In *IJCNN - International Joint Conference on Neural Networks*, Orlando - Florida, 12-17 August 2007.

[CF07] Francesco Camastra and Maurizio Filippone. Svm-based time series prediction with nonlinear dynamics methods. In Apolloni et al. [AHJ07], pages 300–307.

[CH03] Jung-Hsien Chiang and Pei-Yi Hao. A new kernel-based fuzzy clustering approach: support vector clustering with cell growing. *IEEE Transactions on Fuzzy Systems*, 11(4):518–527, 2003.

[Chu97] Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, February 1997.

[CSZ93] Pak K. Chan, Martine Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *Proceeding of the 1993 symposium on Research on integrated systems*, pages 123–142, Cambridge, MA, USA, 1993. MIT Press.

[CTK01] Nello Cristianini, John S. Taylor, and Jaz S. Kandola. Spectral kernel methods for clustering. In *NIPS*, pages 649–655, 2001.

[CV95] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[CV05] F. Camastra and A. Verri. A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):801–804, 2005.

[dCOF06] Miquel de Cáceres, Francesc Oliva, and Xavier Font. On relational possibilistic clustering. *Pattern Recognition*, 39(11):2010–2024, 2006.

[DGK04] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, New York, NY, USA, 2004. ACM Press.

[DGK05] Inderjit Dhillon, Yuqiang Guan, and Brian Kulis. A unified view of kernel k-means, spectral clustering and graph partitioning. Technical Report Technical Report TR-04-25, UTCS, 2005.

[DGK07] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, November 2007.

[DH73a] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17:420–425, 1973.

[DH73b] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.

[Did71] E. Diday. La méthode des nuées dynamiques. *Revue de Stat Appliquée*, 19(2):19–34, 1971.

[FB03] Xiaoli Z. Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 186–193. AAAI Press, 2003.

[FCMR08] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1):176–190, January 2008.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[Fil07] Maurizio Filippone. Fuzzy clustering of patterns represented by pairwise dissimilarities. Technical Report ISE-TR-07-05, Department of Information and Software Engineering, George Mason University, October 2007.

[Fis36] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugenics*, 7:179–188, 1936.

[FMR05] Maurizio Filippone, Francesco Masulli, and Stefano Rovetta. Unsupervised gene selection and clustering using simulated annealing. In Isabelle Bloch, Alfredo Petrosino, and Andrea Tettamanzi, editors, *WILF*, volume 3849 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2005.

[FMR06a] Maurizio Filippone, Francesco Masulli, and Stefano Rovetta. Gene expression data analysis in the membership embedding space: A constructive approach. In *CIBB 2006 - Third International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, Genova - Italy, 29-31 August 2006.

[FMR06b] Maurizio Filippone, Francesco Masulli, and Stefano Rovetta. Supervised classification and gene selection using simulated annealing. In *IJCNN - International Joint Conference on Neural Networks*, Vancouver - Canada, 16-21 July 2006.

[FMR06c] Maurizio Filippone, Francesco Masulli, and Stefano Rovetta. A wrapper approach to supervised input selection using simulated annealing. Technical Report DISI-TR-06-10, Department of Computer and Information Sciences at the University of Genova, Italy, 12th June 2006.

[FMR+06d] Maurizio Filippone, Francesco Masulli, Stefano Rovetta, Sushmita Mitra, and Haider Banka. Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis. In Corrado Priami, editor, *Computational Methods in Systems Biology*, volume 4210 of *Lecture Notes in Computer Science*, pages 312–322. Springer Berlin / Heidelberg, 2006.

[FMR07] Maurizio Filippone, Francesco Masulli, and Stefano Rovetta. Possibilistic clustering in feature space. In *WILF*, Lecture Notes in Computer Science. Springer, 2007.

[FMRC06] Maurizio Filippone, Francesco Masulli, Stefano Rovetta, and Sergiu-Petre Constantinescu. Input selection with mixed data sets: A simulated annealing wrapper approach. In *CISI 06 - Conferenza Italiana Sistemi Intelligenti*, Ancona - Italy, 27-29 September 2006.

[FRB04] Bernd Fischer, Volker Roth, and Joachim M. Buhmann. Clustering with the connectivity kernel. In Sebastian Thrun, Lawrence Saul, and Bernhard schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[GG92] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer, Boston, 1992.

[Gir02] M. Girolami. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[GO98] T. Graepel and K. Obermayer. Fuzzy topographic kernel clustering. In W. Brauer, editor, *Proceedings of the 5th GI Workshop Fuzzy Neuro Systems '98*, pages 90–97, 1998.

[GS88] Paul R. Gorman and Terrence J. Sejnowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1(1):75–89, 1988.

[GST+99] Todd R. Golub, Donna K. Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P. Mesirov, Hilary Coller, Mignon Loh, James R. Downing, Mark A. Caligiuri, Clara D. Bloomfield, and Eric S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, October 1996.

[HB94] Richard J. Hathaway and James C. Bezdek. Nerf c-means: Non-euclidean relational fuzzy clustering. *Pattern Recognition*, 27(3):429–437, 1994.

[HDB89] Richard J. Hathaway, John W. Davenport, and James C. Bezdek. Relational duals of the c-means clustering algorithms. *Pattern Recognition*, 22(2):205–212, 1989.

[HHSV00] Asa B. Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. A support vector method for clustering. In Todd, editor, *NIPS*, pages 367–373, 2000.

[HHSV01] Asa B. Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[HK03] F. Höppner and F. Klawonn. A contribution to convergence theory of fuzzy c-means and derivatives. *IEEE Transactions on Fuzzy Systems*, 11(5):682–694, 2003.

[HN96] Paul Horton and Kenta Nakai. A probabilistic classification system for predicting the cellular localization sites of proteins. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 109–115. AAAI Press, 1996.

[HY91] Zi Q. Hong and Jing Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24(4):317–324, 1991.

[IM04] R. Inokuchi and S. Miyamoto. LVQ clustering and SOM using a kernel function. In *Proceedings of IEEE International Conference on Fuzzy Systems*, volume 3, pages 1497–1500, 2004.

[JD88] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[KCT+01] Lukasz A. Kurgan, Krzysztof J. Cios, Ryszard Tadeusiewicz, Marek R. Ogiela, and Lucy S. Goodenday. Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine*, 23(2):149–169, 2001.

[KJNY01] R. Krishnapuram, A. Joshi, O. Nasraoui, and L. Yi. Low-complexity fuzzy relational clustering algorithms for web mining. *IEEE Transactions on Fuzzy Systems*, 9(4):595–607, 2001.

[KK93] R. Krishnapuram and J. M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[KK96] R. Krishnapuram and J. M. Keller. The possibilistic c-means algorithm: insights and recommendations. *IEEE Transactions on Fuzzy Systems*, 4(3):385–393, 1996.

[KL70] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

[KL07] Hyun C. Kim and Jaewook Lee. Clustering based on gaussian processes. *Neural Computation*, 19(11):3088–3107, 2007.

[Koh90] T. Kohonen. The self-organizing map. In *Proceedings of the IEEE*, volume 78, pages 1464–1480, 1990.

[KR90] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[KSC07] Jaehwan Kim, Kwang-Hyun Shim, and Seungjin Choi. Soft geodesic kernel k-means. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–429–II–432, 2007.

[KVV00] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad, and spectral. In *Proceedings of the 41st Annual Symposium on the Foundation of Computer Science*, pages 367–380. IEEE Computer Society, November 2000.

[LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 1:84–95, 1980.

[Lee05] Daewon Lee. An improved cluster labeling method for support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):461–464, 2005.

[LJGP07] Lau, W. John, Green, and J. Peter. Bayesian model-based clustering procedures. *Journal of Computational & Graphical Statistics*, 16(3):526–558, September 2007.

[Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[LM04] Julian Laub and Klaus R. Müller. Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research*, 5:801–818, 2004.

[LRBB04] Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2004.

[LRBM06] Julian Laub, Volker Roth, Joachim M. Buhmann, and Klaus R. Müller. On the information and representation of non-euclidean pairwise data. *Pattern Recognition*, 39(10):1815–1826, 2006.

[Mac67] J. B. Macqueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[Mac02] David J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, June 2002.

[MBS93] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'Neural gas' network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[Mer09] John Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Proceedings of the Royal Society of London*, 209:415–446, 1909.

[MF00] D. Macdonald and C. Fyfe. The kernel self-organising map. In *Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000*, volume 1, pages 317–320, 2000.

[MMR+01] Klaus R. Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, 2001.

[Moh92] Bojan Mohar. Laplace eigenvalues of graphs: a survey. *Discrete Math.*, 109(1-3):171–183, 1992.

[MRF05] Francesco Masulli, Stefano Rovetta, and Maurizio Filippone. Clustering genomic data in the membership embedding space. In *CI-BIO - Workshop on Computational Intelligence Approaches for the Analysis of Bioinformatics Data*, Montreal - Canada, 5 August 2005.

[MS00] Marina Meila and Jianbo Shi. Learning segmentation by random walks. In *NIPS*, pages 873–879, 2000.

[NJW02] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[OMSRA07] Oh, Man-Suk, Raftery, and E. Adrian. Model-based clustering with dissimilarities: A bayesian approach. *Journal of Computational & Graphical Statistics*, 16(3):559–585, September 2007.

[Pla99] John C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

[PSL90] Alex Pothen, Horst D. Simon, and Kan-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, July 1990.

[QS04] A. K. Qinand and P. N. Suganthan. Kernel neural gas algorithms with application to cluster analysis. *ICPR*, 04:617–620, 2004.

[R D06] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.

[RD06] Adrian E. Raftery and Nema Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, March 2006.

[RLBM02] Volker Roth, Julian Laub, Joachim M. Buhmann, and Klaus R. Müller. Going metric: Denoising pairwise data. In *NIPS*, pages 817–824, 2002.

[RLKB03] Volker Roth, Julian Laub, Motoaki Kawanabe, and Joachim M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1540–1551, 2003.

[RMF05] Stefano Rovetta, Francesco Masulli, and Maurizio Filippone. Soft rank clustering. In Bruno Apolloni, Maria Marinaro, Giuseppe Nicosia, and Roberto Tagliaferri, editors, *WIRN/NAIS*, volume 3931 of *Lecture Notes in Computer Science*, pages 207–213. Springer, 2005.

[RMF07] Stefano Rovetta, Francesco Masulli, and Maurizio Filippone. Membership embedding space approach and spectral clustering. In Apolloni et al. [AHJ07], pages 901–908.

[Ros98] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, November 1998.

[Rou78] M. Roubens. Pattern classification problems and fuzzy sets. *Fuzzy Sets and Systems*, 1(4):239–253, October 1978.

[RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[Rus93] E. H. Ruspini. Numerical methods for fuzzy clustering. In D. Dubois, H. Prade, and R. R. Yager, editors, *Readings in Fuzzy Sets for Intelligent Systems*, pages 599–614. Kaufmann, San Mateo, CA, 1993.

[RW05] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005.

[Sai88] S. Saitoh. *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England, 1988.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.

[SS73] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman, San Francisco, 1973.

[SS01] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[SSM98] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[SWHB89] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest*, 10:262–266, 1989.

[TD99] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, 1999.

[Vap95] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[VM05] Deepak Verma and Marina Meila. A comparison of spectral clustering algorithms. Technical report, Department of CSE University of Washington Seattle, WA 98195-2350, 2005.

[War63] J. H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[Win85] Michael Windham. Numerical classification of proximity data with assignment measures. *Journal of Classification*, 2(1):157–172, December 1985.

[WM90] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences,U.S.A.*, 87:9193–9196, 1990.

[WW93] Dorothea Wagner and Frank Wagner. Between min cut and graph bisection. In *Mathematical Foundations of Computer Science*, pages 744–750, 1993.

[WXY03] Zhong D. Wu, Wei X. Xie, and Jian P. Yu. Fuzzy c-means clustering algorithm based on kernel method. *Computational Intelligence and Multimedia Applications*, 2003.

[YEC02] Jianhua Yang, Estivill, and S. K. Chalup. Support vector clustering through proximity graph modelling. In *Proceedings of the 9th International Conference on Neural Information Processing*, volume 2, pages 898–903, 2002.

[YS03] Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, Washington DC, USA, 2003. IEEE Computer Society.

[ZC02] Dao Q. Zhang and Song C. Chen. Fuzzy clustering using kernel method. In *The 2002 International Conference on Control and Automation, 2002. ICCA*, pages 162–163, 2002.

[ZC03] Dao Q. Zhang and Song C. Chen. Kernel based fuzzy and possibilistic c-means clustering. In *Proceedings of the International Conference Artificial Neural Network*, pages 122–125. Turkey, 2003.

[ZC04] Dao Q. Zhang and Song C. Chen. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32(1):37–50, 2004.