CENATAV
Centro de Aplicaciones de Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142
ISSN 2072-6287
Versión Digital

REPORTE TÉCNICO
Reconocimiento de Patrones

SERIE AZUL

Clustering of Simple and Multi-way Spectral Data on the Dissimilarity Representation

Victor Mendiola-Lau, Isneri Talavera Bustamante, and Maria De Marsico

RT_076 octubre 2015

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

CENATAV
Centro de Aplicaciones de
Tecnologías de Avanzada
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142
ISSN 2072-6287
Versión Digital

REPORTE TÉCNICO
Reconocimiento
de Patrones

SERIE AZUL

Clustering of Simple and Multi-way Spectral Data on the Dissimilarity Representation

Victor Mendiola-Lau, Isneri Talavera Bustamante, and Maria De Marsico

RT_076 octubre 2015

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

Table of Content

Clustering of Simple and Multi-way Spectral Data on the Dissimilarity Representation

Victor Mendiola-Lau[1], Isneri Talavera Bustamante[1], and Maria De Marsico[2]

[1] Pattern Recognition Research Team, Advanced Technologies Application Center (CENATAV), Havana, Cuba
{vmendiola, italavera}@cenatav.co.cu
[2] Sapienza University of Rome, Italy
demarsico@di.uniroma1.it

RT_076, Serie Azul, CENATAV
Accepted: September 1, 2015

Abstract. Spectral data are very common in many disciplines such as chemometrics and biometrics, although they can also be found in many other fields like biology, medicine, signal processing, etc. Despite their ubiquity, spectral data can often have more complex forms, either because the same object is described by several spectra or it is represented as a multi-way array, which results from a combined instrumental analysis technique. Whatever the case, spectral data is usually analyzed in the framework of statistical pattern recognition. The connected nature of spectral data, as well as its structural information, e.g. its shape, is often ignored because spectra are represented as feature vectors where relations among variables is not taken into account. The dissimilarity representation approach allows to take into account such information omitted by traditional representations, thus improving supervised classification results. This approach has been widely studied for supervised classification problems, however due to their relevance, unsupervised classification problems should not be left out. In this work, a critical review is provided concerning the dissimilarity representation of simple and multi-way spectral data, as well as clustering strategies for proximity data proposed in the literature. At last, some suggestions are made regarding future research perspectives on these topics.

Keywords: spectral data, multi-way spectral data, dissimilarity representation, clustering.

Resumen. Los datos espectrales son muy comunes en disciplinas como la quimiometría y la biometría, aunque también pueden estar presentes en diversos campos como la biología, la medicina, el procesamiento de señales, etc. A pesar de su ubicuidad, los datos espectrales pueden a menudo tener formas más complejas, ya sea porque el mismo objeto es descrito mediante diversos espectros o porque es representado como un arreglo multi-vía proveniente de una técnica combinada de análisis instrumental. En cualquiera de los casos, los datos espectrales son comúnmente analizados en el marco del reconocimiento estadístico de patrones. La naturaleza continua de los datos espectrales, así como la información de índole estructural, por ejemplo su forma, es frecuentemente ignorada debido a que éstos son representados como vectores de rasgos donde no se tiene en cuenta la relación entre las variables. El enfoque de representación por disimilitudes ofrece la oportunidad de tener en cuenta dicha información omitida por las representaciones tradicionales, permitiendo mejorar los resultados en tareas de clasificación supervisada. Este enfoque ha sido ampliamente estudiado para problemas de clasificación supervisada, no obstante debido a su relevancia, los problemas de clasificación no supervisada no deberían ser ignorados. En este trabajo se proporciona un análisis crítico sobre la representación por disimilitudes de datos espectrales simples y multi-vías, así como las estrategias de agrupamiento para datos basados en proximidades propuestos en la literatura. Finalmente, se realizan algunas sugerencias respecto a perspectivas futuras de investigación sobre estos temas.

Palabras clave: datos espectrales, datos espectrales multi-vías, representación por disimilitudes, agrupamiento.

1 Introduction

Nowadays, several instrumental techniques allow the analysis of a variety of experimental data produced in the context of different case studies in many branches of science, including biology, medicine, material characterization, food, environment, physics, geophysics, pharmaceutical and forensics. Spectral information can be seen as a distinctive signature of objects and in many scenarios, spectra are often a source of challenging supervised and unsupervised classification problems. This kind of data is collected from a wide variety of sensors, e.g., spectroscopic equipment like NIR Spectroscopy, UV spectroscopy, Gas chromatography (GC), NMR Spectroscopy, Mass spectroscopy, Thermal Analysis, Atomic Spectroscopy, and others.

The instrumental techniques previously mentioned result in *simple*[1] spectral data for each object or sample. Over time, progress has been achieved in fields such as spectral data acquisition and new strategies for the analysis and computational representation of this kind of data have been developed. At first, only the information regarding a single analytical technique was available, but lately, with the development of new analytical techniques, each object or sample is usually analyzed by means of several techniques, i.e., several spectral data are acquired. Taking this fact into consideration poses the need of finding a *proper combination* of such complementary information in a learning process. In this context, some advanced tools might be required in the learning stage such as *combination of classifiers*[1] if data is properly labeled, or *clustering ensemble*[2] if data is unlabeled.

More recently, due to the development of *combined* instrumental analysis techniques such as gas chromatography-mass spectrometry and excitation-emission fluorescence, among others; objects can be represented as a multi-dimensional array. Such structures, containing information about the relation between the different types of measurements, can be useful for a better understanding of the problem at hand. The field dedicated to study and develop proper tools to analyze this kind of data sets is known as multi-way data analysis[3,4]. From the existing designs of multi-way data, we will focus in profile data[3], which is tailored to the *combined* spectral data mentioned earlier. In this design, the objects lie in one of the modes while the features characterizing such objects can be found in the rest of the modes. This multi-way data can then be used in subsequent learning tasks over the measured objects. It is important to stress that for this kind of data, proper learning frameworks should be used to ensure a correct and reliable analysis.

In practice, there are often very few samples due to the high cost of the instrumental study (equipment, materials, etc.), the time required to generate such samples, etc. Additionally, the results of the measurements are continuous spectral data that usually have a very high dimension[2]. It is known that a classical *statistical pattern recognition* (SPR) approach usually fails with this kind of data for two main reasons: it neglects key aspects like the *connected* nature of the spectral data and it will almost surely face the problem of the ***curse of dimensionality***[5,6] due to high data dimension. In this work, the term *curse of dimensionality* is referred to as the difficulties experienced by learning algorithms when dealing with few high dimensional data.

Classical SPR traditionally relies on vector-based representations. If the available domain knowledge is sufficient, then a small set of well-discriminating features can be determined. However, if no such knowledge is available, a large set of *suboptimal* features may be selected, possibly leading to class overlap and feature dependency. In this scenario, a probabilistic framework is employed to address the learning task and it is *conveniently* assumed that real world objects or phenomena are described by (or in fact, reduced

[1] In this context, the term *simple* refers to 1D or 2D data.

[2] From a few hundreds of components to several hundreds of thousands of components. This is also valid for *matricized* multi-way data.

to) a set of vectors in a *suitable* vector space. Although this framework is mathematically well founded, it relies on some general assumptions[3]. Sometimes, fixed distributions are imposed over the objects/features even in scenarios where such information cannot be determined; for instance, the distribution of non-faces in the task of face detection is unknown, and some classes may have to be artificially built in order to have enough samples for training, etc. Moreover, the available training set is assumed to be representative for the task, which is not necessarily true in all situations.

When constructing SPR systems, researchers not always (or only partially) address many key issues to SPR like the way objects may be represented. Being pattern recognition one of the disciplines with a very strong connection with learning, emphasis in research should be addressed to the development of suitable representations for real world objects. In [7], the authors proposed a more general description of a pattern recognition system, leaving room for solutions that are not necessarily feature-based, but use other representations. Taking all this into consideration, a conclusion could be drawn: a feature vector representation of spectral data might not be *appropriate*. Hence, an alternative way to tackle the problem might be to use a *different* representation that somehow incorporates specific knowledge of the problem at hand.

For spectral data, some alternative representations such as *functional data analysis* (FDA)[8] and the *dissimilarity representation* (DR)[9] have been used[10]. Although both representations capture the essential characteristics of spectral data, the dissimilarity representation allows that any additional knowledge regarding the problem can be included into the dissimilarity measure, thus allowing the expert to build a more robust measure. It has also a major impact when dealing with a few samples represented as high dimensional vectors, which is the case of the aforementioned spectral data, since the space complexity is reduced to $O(n^2)$ instead of the original $O(nm)$. This phenomenon has great implications in efficiency because the *complexity* of the data set is drastically reduced. Some works have been reported regarding the comparison of chemical spectral data by means of this approach. In [11] infrared (IR) spectrometry is studied, in [12] a new dissimilarity measure for Ultraviolet Spectra (UVS) is presented and more recently, in [13], the dissimilarity representation was analyzed for spectral data representation and classification in a wide range of applications.

However, this representation was mainly thought, and it has been used mostly in supervised classification scenarios[14]. Although many procedures for cluster analysis make use of dissimilarities[15] instead of feature spaces, to our knowledge, the problem of unsupervised classification on the dissimilarity representation has not been as extensively explored as the supervised one. Given that cluster analysis plays such an important role in a broad range of pattern recognition applications, we believe it is an issue of great importance and future efforts should be devoted to its study under this representation.

The rest of this work is structured as follows: in section 2 an overview of multi-way data analysis is provided. This is essential to achieve a proper understanding of multi-way spectral data. Next, in section 3 the key aspects of the dissimilarity representation and the dissimilarity measures for spectral data proposed in the literature are discussed. In section 4, several strategies and algorithms to perform unsupervised classification based on dissimilarity information are analyzed. Finally in section 5 after a brief discussion, some conclusions of this research are drawn and several proposals for future works are outlined.

2 Multi-way Data Analysis. An Overview

The standard way of representing objects is in a two-way structure (matrix), where a number of objects (rows) are simply characterized by feature vectors (columns). Nevertheless, in many research areas such as

[3] As the modeling of objects as vectors in a *suitable* vector space.

process monitoring[16] (specially in analytical chemistry), in the field of signal processing, environmental studies[17], social network analysis[18] and neuroscience[19], objects observed by sensors are represented by higher-order generalizations of vectors and matrices, i.e., several types of information are measured over the objects, as for example, data collected at different times or conditions. The structure in which a set of objects with this kind of representation are organized is called multi-way data.

Multi-way data contains information about the relation between the different types of measurements, which can be useful for a better comprehension of the problem at hand. However, despite the ubiquity of this kind of data, the real challenge lies in how to make a proper analysis. The field dedicated to study and develop proper tools to analyze this kind of data sets is known as multi-way data analysis [4,3]. This field is an extension of multivariate analysis when the analyzed data are high-order arrays. It originated in 1952 [20], when Raymond Cattell first introduced the very important term of multi-way arrays and was further investigated later, in 1964 [21], when Tucker introduced the three-mode component analysis. It is often used to capture the underlying structure of the data and explore its interrelations. On the other hand, it has been shown that this kind of information from the data may not be accurately obtained or identified uniquely by two-way analysis methods because they do not respect the multi-way design of the data.

2.1 General Concepts

In order to comprehend multi-way data, we must first understand simpler data and then extend the concepts to higher-order arrays. Traditional 2-way data are represented through matrices (see figure 1 left) where rows usually correspond to objects and columns to variables. A *cube* is represented by a 3-dimensional array (see figure 1 right). In figure 1, a cube with ***I*** objects, ***J*** variables of type 1 and ***K*** variables of type 2 can be observed.

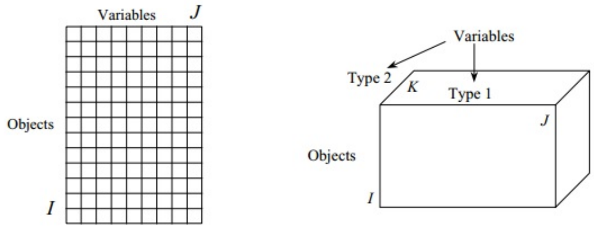


Fig. 1. (Source: [4]) Examples of 2-way data (left) and 3-way data (right).

Classical concepts like rows and columns of 2-way data are *replaced* by the concept of *slice* (or *slab*) (see figure 2). Each horizontal slice contains the information related to an object of the 3-way data, each vertical slice contains the information related to a variable of type 1 and finally, every frontal slice contains the information related to a variable of type 2.

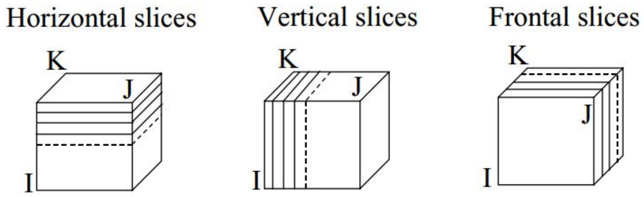


Fig. 2. (Source: [4]) Splitting 3-way data into *slices* (2-way matrices).

Nevertheless, the concepts of row, column and tube can be defined in the context of a 3-way array (see figure 3). In a 3-way array $\boldsymbol{X}(I \times J \times K)$ there are $J \times K$ columns x_{ij} $(I \times 1)$; $I \times K$ rows x_{ik} $(J \times 1)$ and $I \times J$ tubes x_{ij} $(K \times 1)$.

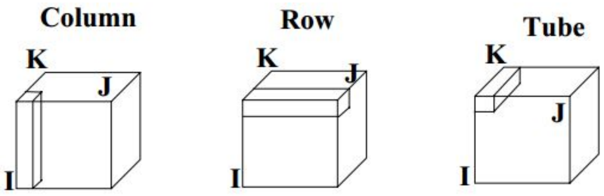


Fig. 3. (Source: [4]) Rows, columns and tubes defined over a 3-way array.

The process of transforming a 3-way array into a 2-way matrix is commonly known as *matricization*[4] and it is described in figure 4 [22]. It is important to point out that there are, in this case, four other ways to matricize $\boldsymbol{X}$, i.e. matrices of dimensions: $(J \times IK)$, $(J \times KI)$, $(K \times IJ)$ and $(K \times JI)$.

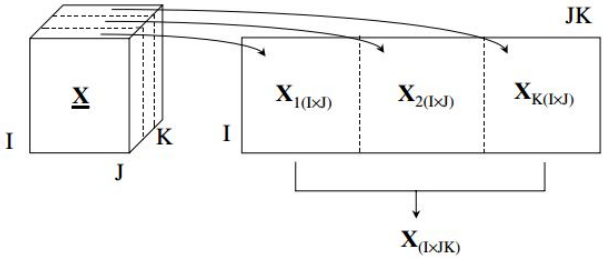

JK
$\underline{\mathbf{X}}$
I
K
J
I
$\mathbf{X}_{1(I\times J)}$
$\mathbf{X}_{2(I\times J)}$
$\mathbf{X}_{K(I\times J)}$
$\mathbf{X}_{(I\times JK)}$

Fig. 4. (Source: [4]) The matricization of a 3-way array $\boldsymbol{X}(I \times J \times K)$ is a 2-way matrix $\boldsymbol{X}(I \times JK)$.

In many scenarios, there is the need to preprocess the data in order to correct deficiencies in the acquisition stage. As in multivariate analysis, mean centering and scaling are the most common operations. Both centering and scaling can be applied to the whole 3-way array, but also, they can be applied by slices (horizontal, vertical or frontal) or by vectors (rows, columns or tubes). For a deeper analysis of these techniques, [4] can be consulted.

2.2 Multi-way Decomposition Models

Multi-way decomposition models are an extension to multi-way data of classical multivariate decomposition models. These models are usually divided into two parts: a *structural part* describing the underlying structure of the data and a *residual part*, which accounts for the information that could not be captured by the structural part. Next, two families of decomposition models commonly used are discussed: the PARAFAC family and the Tucker family.

2.2.1 The PARAFAC Family

Within this family PARAFAC is the fundamental model, since the other models in this family are simply extensions or *amendments* to this model. PARAFAC [23] in turn, is an extension for multi-way data of bilinear factor models and it is based on the principle of Parallel Proportional Profiles [24].

[4] The conversion of an object into a matrix.

Let $\mathbf{X} \in R^{I \times J \times K}$ be a 3-way array. Then, a PARAFAC model of R[5] components is mathematically defined as in equation 1, where a_i, b_i and c_i account for the i-th column of the component matrices $A \in R^{I \times R}$, $B \in R^{J \times R}$ and $C \in R^{K \times R}$ respectively. $\mathbf{E} \in R^{I \times J \times K}$ is a 3-way array containing the residual terms and the operator $\circ$ represents the outer product between vectors

$$\mathbf{X} = \sum_{r=1}^{R} a_r \circ b_r \circ c_r + E \ . \tag{1}$$

The component matrices A, B y C are known as *loading* matrices, although in many cases the matrix A is also known as the *score* matrix[6], since it is usually associated with the samples or objects. Once a PARAFAC decomposition model is determined, each sample could be represented by a *score* vector, being this a very common strategy to deal with multi-way data using techniques of multivariate analysis. An illustration of a PARAFAC model can be observed in figure 5.

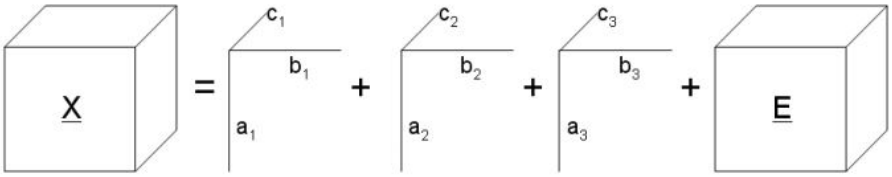


Fig. 5. (Source: [25]) Illustration of a 3-component PARAFAC model.

Another relevant model in this family is the PARAFAC2 model [26], which is a less restrictive model than PARAFAC. The main advantage of PARAFAC2 over the PARAFAC model, is that PARAFAC is not able to extract oblique factors (non orthogonal factors), while PARAFAC2 performs well in both situations. There are other important models in this family and they are briefly discussed in [25].

2.2.2 The Tucker Family

The models from the PARAFAC family can be considered as restricted versions of models belonging to the Tucker family. The simplest model in this family, which is the base for the rest, is the Tucker3 model [27]. Similar to PARAFAC, Tucker3 is also a generalization of bilinear factor models to higher-order data sets. However, unlike a PARAFAC model, a Tucker3 model can extract a different number of components from each mode and the factors from different modes can interact with each other.

Let $\mathbf{X} \in R^{I \times J \times K}$ be a 3-way array. Then, a Tucker3 model is mathematically defined as in equation 2, where $A \in R^{I \times P}$, $B \in R^{J \times Q}$ and $C \in R^{K \times R}$ are the component matrices corresponding to the first, second and third mode respectively[7]. $G \in R^{P \times Q \times R}$ is known as the *core array* and $E \in R^{I \times J \times K}$ is a 3-way array containing the residual terms. An illustration of a Tucker3 model can be observed in figure 6

$$x_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr} + e_{ijk} \ . \tag{2}$$

Tucker3, as Principal Components Analysis (PCA), can always *represent* a data set to some extent by including a sufficient amount of components, so Tucker3 does not suffer from the numerical and math-

[5] The amount of components R (a natural number) is also denoted as F or K in the literature.

[6] The *scores* are the *loadings* in the object mode.

[7] Similar to PARAFAC, P, Q and R are the amount of components extracted for each mode respectively.

ematical problems that sometimes make applications of the PARAFAC model difficult. There are other important models in this family and they are briefly discussed in [25].

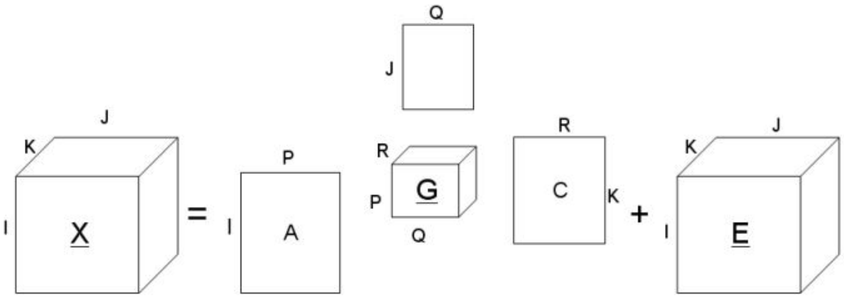


Fig. 6. (Source: [25]) Illustration of a (P, Q, R)-component Tucker3 model.

2.3 Multi-way Learning Strategies

There are three main strategies used in the literature to tackle learning problems when the data has a multi-way structure[28]. These strategies, as well as their shortcomings and benefits are discussed next.

2.3.1 Matricization + Two-way Learning

There are still many works that instead of using the entire multi-way array, create a two-way matrix out of it to analyze it later with two-way models. Any traditional multivariate learning technique can be used afterwards. Sometimes this may do no harm, because it is possible to draw the conclusion that a 2-way analysis could be sufficient, but most of the time it is inefficient. This approach might lead to overfitting given that the lower level model does not interpret the data properly and it neglects its natural composition. Another limitation is that the interpretation of the results obtained by a multivariate learning technique, in terms of the original variables, can be very difficult if not impossible.

2.3.2 Multi-way Decomposition + Two-way Learning

Another approach commonly used to perform learning tasks over multi-way, data without resorting to the previous strategy, is to decompose the data array by a multi-way decomposition technique (for example PARAFAC or Tucker3) and then use the sample scores as a new set of variables. This new set of variables can afterwards be subjected to two-way classification methods. The first step (data decomposition), takes full advantage of the multi-way nature of data, while the second step of building classification rules is disconnected from the multi-way model. Hence, it will not be possible to have a direct interpretation of the raw data in terms of their contributions to class separation.

2.3.3 Multi-way Learning Methods

The third and last strategy is to use learning techniques that exploit the natural (multi-way) structure of the data[4,3,29]. This kind of techniques take advantage of the interrelations between the modes in the complex multi-way structure for the learning task of interest, meaning that potentially better results could be obtained from this approach.

2.4 Multi-way Classification

In the context of multi-way classification, all the strategies discussed earlier in section 2.3 have been applied. However, the application of truly multi-way classification methods, is still limited. In this section, the most relevant multi-way classifiers will be discussed.

There are few multi-way classifiers developed until today and, for these classifiers, the representation and classification are seen as two different steps. This is the case of *multi-way partial least squares discriminant analysis* (NPLS-DA) [30] and N-SIMCA [31] classifiers. They are both extensions to multi-way data of the former PLS-DA and SIMCA methods developed for two-way data.

In the case of NPLS-DA, the N-PLS regression method aims at establishing a linear relationship between a set of explanatory variables, i.e., measurements performed over the objects, and response/dependent variables, e.g., some type of properties. Although the PLS method is primarily regarded as a calibration method, it can also be used for classification. In such case, the dependent variables are the classes, so the regression is performed to class labels.

For the N-SIMCA classifier, like for two-way SIMCA [32], a separate classification model by means of a component analysis is built for each of the classes and class boundaries are estimated to classify the new objects. In this case, multi-way decomposition methods, e.g., PARAFAC or Tucker3 are applied.

Despite the capabilities of the aforementioned multi-way classifiers, it should be emphasized that they do not consider context/background dependent information from objects into their analysis. For example, in the particular case of multi-way spectral data or any other continuous nature data, a key feature like shape is usually not taken into account. Therefore, discriminative context information deriving from their original nature is not reflected in the representation, nor in the classification stage.

Recently in [13], a new classification approach based on the dissimilarity representation was proposed. This proposal overcomes the problems mentioned earlier because such information can be *encoded* in the dissimilarity measure. The classification is accomplished on the dissimilarity space, where a wide variety of learning techniques can be applied. In spite of the remarkable classification performance, this proposal does not allow model interpretation in terms of the set of variables/spectral features used to characterize the samples. Therefore, one might say that among current multi-way classifiers, there is a trade off between classification performance and model interpretation.

3 The Dissimilarity Representation. An Overview

The dissimilarity representation aims at treating objects as a whole[8], avoiding the use of absolute features. It can be regarded as an alternative approach to the use of feature-based representations in the recognition of real world objects like images, spectra and time-signal data. Instead of a feature-based characterization, a measure that estimates the dissimilarity between pairs of objects is defined. The similarity/dissimilarity measure can be defined over any kind of objects like images, feature vectors and even structural representations, e.g., graphs, strings, etc. This ability to effectively represent structural data is the reason why the dissimilarity representation is said to potentially being able to bridge the gap between structural and statistical pattern recognition.

Under this framework, proper knowledge of class densities is not needed and the training set is only required to be representative for the domain of the classes. Some authors[33,34,35] believe that the underlying concept behind the dissimilarity representation, the notion of *similarity*, is more fundamental than that of a *feature* or of a *class*, since it is the similarity which groups the objects together and, thereby, it

[8] By means of a characterization in terms of proximities/dissimilarities to other objects.

should play a crucial role in the class description. Using the notion of *proximity* (instead of features) as a core concept, the dissimilarity representation renews the area of statistical learning in one of its foundations: the representation of objects[36].

This representation has been successfully applied in several disciplines like face recognition[37,38,39], video surveillance[40,41], chemometrics[10,13] (particularly in the classification of spectral data), and many others. Next, the fundamental aspects of this representation are discussed.

3.1 General Concepts

As already mentioned, the notion of *proximity* might play a key role in class formation. Essentially, *similar* objects can be grouped together to form a class, and consequently a class is a set of *similar* objects. A similarity can be modeled by a similarity measure and also by a dissimilarity measure. Both are closely related; a small dissimilarity and a large similarity both imply a close resemblance of objects. In this work our attention will be centered on dissimilarities, which focus on class and object differences.

In SPR, objects are usually analyzed by means of a feature representation. A feature could be described as the *combination* of measured characteristics of an object: for instance, the weight is an attribute for the class of apples, then a feature consists of the measured weights for a number of apples. For a set T of N objects, a feature-based representation relying on a set F of m features is encoded as an $N \times m$ matrix $A(T,F)$, where each row is a vector describing the feature values for a particular object. Instead, a dissimilarity representation of objects is based on pairwise comparisons and is expressed as an $N \times N$ matrix $D(T,T)$. Each cell of D corresponds to a dissimilarity value computed between pairs of objects. Hence, each object x is represented by a vector of *proximities* $D(x,T)$ to the objects in T. In figure 7 both approaches to object representation can be observed.

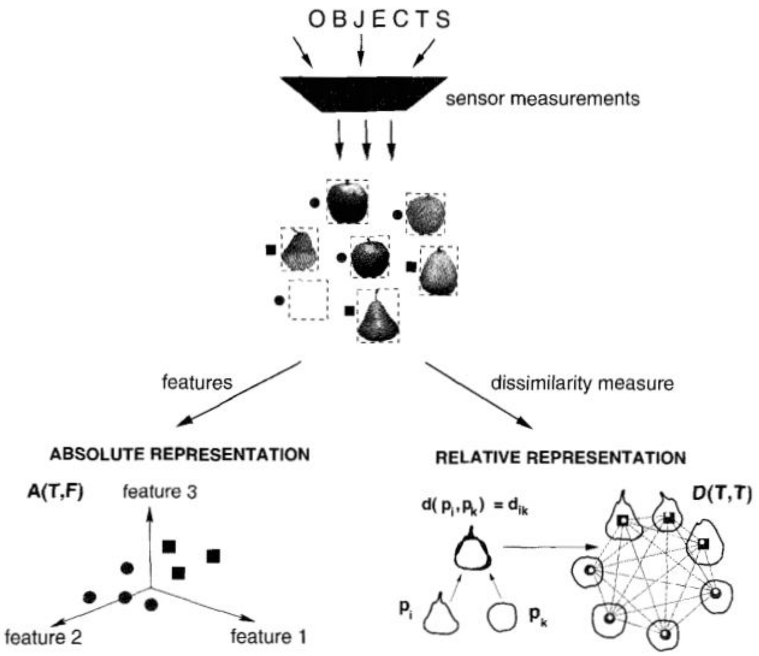
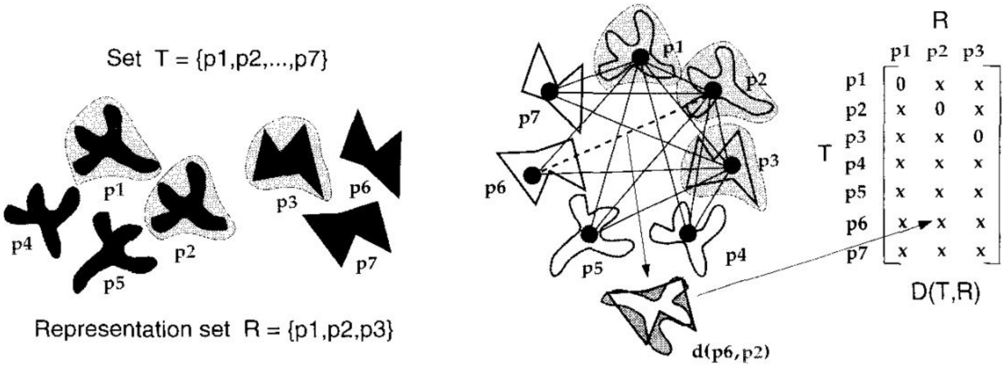


Fig. 7. (Source: [9]) Feature-based (absolute) representation vs. dissimilarity-based (relative) representation.

In the context of dissimilarity representations, two main types of representations are considered: the ones which are learned and the ones which are optimized (or fixed). In this work, special attention is devoted to the second type[9], which will be denoted simply as *proximity representations* from now on. Proximity representations can be further divided in two types: *relative* and *conceptual*. In a *relative* representation, each object is described by a set of proximities to other objects. On the other hand, in a *conceptual* representation, each object is described by the proximity to a model (or concept)[10]. In this work, our main focus is on relative dissimilarity representations. Next, a definition of such dissimilarity representation is given.

Definition 1 (Dissimilarity representation). *Let $T = \{p_1, p_2, \cdots, p_N\}$ be a collection of N objects and let d be a dissimilarity measure computed or derived from the objects directly, their sensor representations, string representations or other intermediate representations. A dissimilarity representation of an object x is a set of dissimilarities/proximities between x and the objects in T denoted by a vector $D(x,T) = [d(x,p_1), d(x,p_2), \cdots, d(x,p_N)]$; and the set of objects T will be represented by the dissimilarity matrix $D(T,T)$. Usually, a relatively small set of representative objects for the domain considered is used for representation. Given a representation set $R = \{p_1, p_2, \cdots, p_M\}$, an object x is now represented as $D(x,R) = [d(x,p_1), d(x,p_2), \cdots, d(x,p_M)]$, where x is related to each* ***prototype*** *in the representation set. Therefore, instead of the dissimilarity matrix $D(T,T)$, the set of objects T is now represented by a dissimilarity matrix $D(T,R)$ computed over the representation set R.*



	p1	p2	p3
p1	0	x	x
p2	x	0	x
p3	x	x	0
p4	x	x	x
p5	x	x	x
p6	x	x	x
p7	x	x	x

Fig. 8. (Source: [9]) Dissimilarity representation $D(T,R)$. The representation objects are elements of the set T.

Two important aspects should be taken into consideration. First, as can be seen in figure 8, the representation set R might be a subset of $T\,(R \subseteq T)$ or they might be completely distinct sets. If computationally tractable, one may also use the complete representation $D(T,T)$ and try to select R to optimize a particular classifier on $D(T,R)$. And second, one should be very careful not to confuse resemblance between dissimilarity and feature-based representations regarding their matrix notation, because the meaning is different (see figure 9).

3.2 Learning Frameworks

Once we have a dissimilarity representation $D(T,R)$ of our objects as a starting point, a proper interpretation of the dissimilarity information is needed for supervised and unsupervised learning tasks. In the

[9] The learned proximity representations are still an open issue.

[10] For example, the *similarity* of an object to the *class* of all kinds of apples.

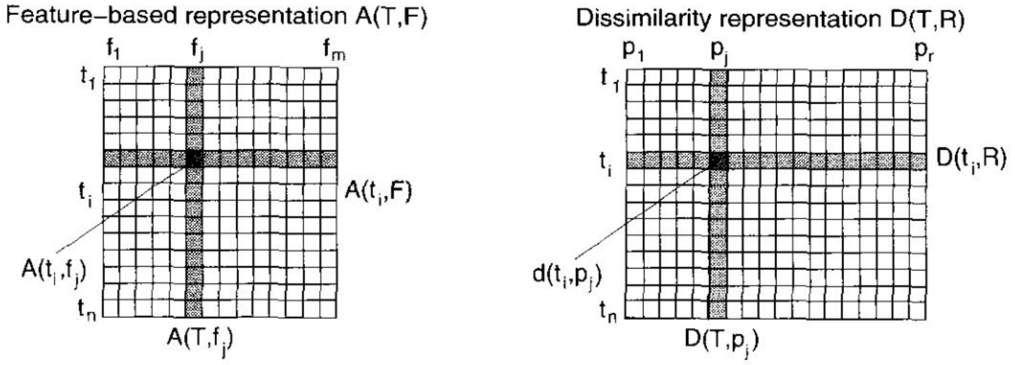

Feature–based representation A(T,F)
f1
fj
fm
t1
ti
tn
A(ti,F)
A(ti,fj)
A(T,fj)
Dissimilarity representation D(T,R)
p1
pj
pr
t1
ti
tn
D(ti,R)
d(ti,pj)
D(T,pj)

Fig. 9. (Source: [9]) Feature-based representation $A(T,F)$ (left). Dissimilarity representation $D(T,R)$ (right). Let $T = \{t_1,\cdots,t_n\}$ be a set of training objects and $F = \{f_1,\cdots,f_m\}$ be the features. An object t, is represented as a vector of its feature values $a(t_i, f_j)$ i.e. $A(t_i,F) = [a(t_i, f_1),\cdots,a(t_i, f_m)]$ and the feature f_j is represented as a vector $A(T, f_j)$. A dissimilarity representation describes the relations between objects, hence additionally, a collection of representatives $R = \{p_1,\cdots,p_r\}$ is needed. An object t_i is represented by a vector of its dissimilarities $d(t_i, p_j)$ to the objects from R, i.e. $D(t_i,R) = [d(t_i, p_1),\cdots,d(t_i, p_r)]$. $D(T, p_j) = [d(t_1, p_j),\cdots,d(t_n, p_j)^T$ refers to dissimilarities to a particular object p_j.

literature, three learning frameworks have been studied for the dissimilarity representation: the *pretopological approach*, the *embedding approach* and the *dissimilarity space approach*.

3.2.1 Pretopological Approach

The *pretopological approach* makes a direct interpretation of dissimilarities between the objects making use, directly or not, of pretopological spaces. This means that a dissimilarity representation in this case describes a space where the notion of a norm or a distance is not yet available and therefore, the basic neighborhoods play a key role. These neighborhoods are defined by the use of dissimilarities to the objects from R[9], and they can be considered as basic concepts to build a space that can effectively express the relations between the objects in T.

The traditional way of classifying a new object x represented by $D(x,R)$ is by means of the nearest neighbor rule[42,5]. The object x is classified into the class of its nearest neighbor, that is the class of the representation object p_i given by $p_i = \arg min_{p_j \in R} d(x, p_j)$. Moreover, a more refined strategy could be to assign x to the class that most occurs among its k nearest neighbors, by interpreting the dissimilarities directly. The information stored in $D(T,R)$ is not used by the classification procedure (in fact, there is no training), that is why if $R = T$ is used, the classification is the most reliable that can be achieved as it is based on the entire *training set*. However, several approaches exist to select a *smaller* representation set R ($|R| < |T|$) to speed up the classification. Under some circumstances, the recognition may even be improved by this reduction.

3.2.2 Embedding Approach

Embeddings are a useful approach in practical problems where finite dissimilarity representations are considered, that is, finite metric spaces (X,d) defined by the corresponding dissimilarity matrix D. The main goal with this strategy is to find an alternative representation space suitable for learning, where the notion of closeness between objects is somehow preserved.

The *embedding approach* computes a spatial representation of a symmetric $D(R,R)$[11], i.e., a vector space V, where the objects are mapped as points such that their distances reflect the actual dissimilarities.

[11] This approach requires a symmetric dissimilarity measure.

Then, the remaining objects $T \setminus R$, if any, are projected to the computed space. Next, the learning process takes place in this embedded vector space (see figure 10), which is possible because the embedded vector space is equipped with desirable properties[12], if needed.

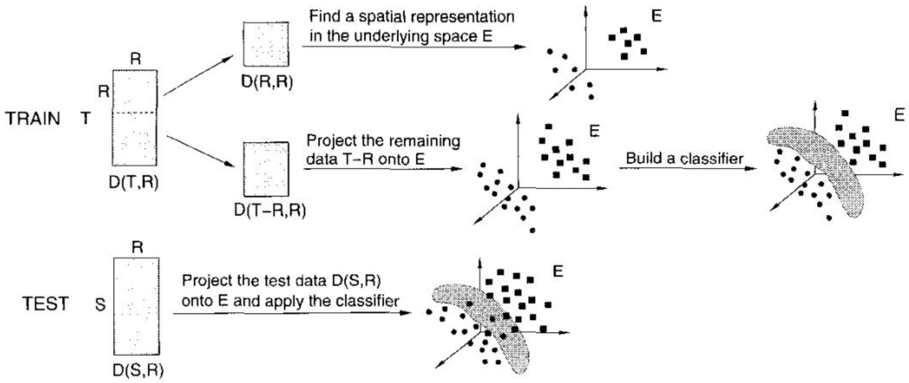

TRAIN
T
R
R
D(T,R)
D(R,R)
Find a spatial representation in the underlying space E
E
D(T−R,R)
Project the remaining data T−R onto E
E
Build a classifier
E
TEST
S
R
D(S,R)
Project the test data D(S,R) onto E and apply the classifier
E

Fig. 10. (Source: [9]) Classification in a embedded space.

Many representation spaces have been investigated for this purpose such as Hilbert and Euclidean spaces. However, a Euclidean space is not always capable to accommodate such a dissimilarity-preserving mapping, but a pseudo-Euclidean space is[43]. Due to their properties, a probabilistic framework and many theoretical models exist, which are used to solve pattern recognition problems formulated in such spaces.

3.2.3 Dissimilarity Space Approach

The *dissimilarity space approach* considers the dissimilarities as the *features* of the objects instead of the distances between them. In this context, the dissimilarity representation is regarded as a data-dependent mapping specified by the representation set R. Such mapping $\phi(\cdot, R) : T \to R^n$ is defined as:

$$\phi(x,R) = [d(x,p_1), d(x,p_2), \cdots, d(x,p_n)] \ . \tag{3}$$

Note that T denotes either the objects themselves or a feature-based vector representation of them. Once again, R can be T itself or a set of representative objects (prototypes) $R \subseteq T$, which can be found by a prototype selection method [9,44]. The learning task can now take place as in a traditional vector space, in which each dimension corresponds to a dissimilarity to a representation object $d(\cdot, p_i)$ (see figure 11).

The assumption that dissimilarities should be small for similar objects belonging to the same class and large for objects belonging to different classes, gives a possibility for discrimination. Besides, a dissimilarity vector space is assumed to be endowed with the traditional inner product and the associated norm and Euclidean metric, thus allowing for the traditional learning to take place. In this space, supervised learning techniques usually achieve outstanding results while maintaining at the same time a low computational cost determined by the number of objects in R.

3.2.4 Choosing a Learning Framework

In spite of the many advantages provided by the three learning frameworks[13] mentioned earlier, there are some shortcomings that should not be ignored. The main reason for such limitations is in many cases, a

[12] For example like an inner product, additional algebraic structures, etc.

[13] Also referred to as learning paradigms.

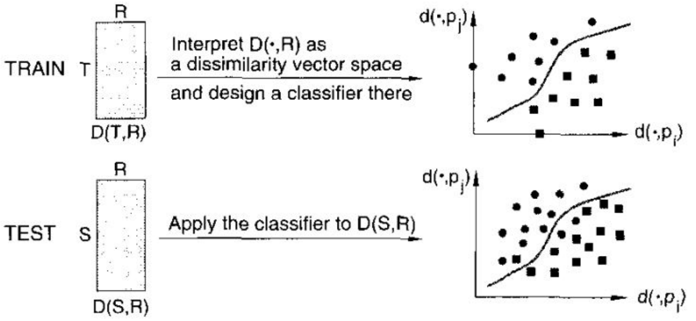

R
TRAIN T
D(T,R)
Interpret D(•,R) as
a dissimilarity vector space
and design a classifier there
d(•,p_j)
d(•,p_i)
R
TEST S
D(S,R)
Apply the classifier to D(S,R)
d(•,p_j)
d(•,p_i)

Fig. 11. (Source: [9]) Classification in a dissimilarity space.

result of the same assumptions made by such learning paradigms. With the exception of the *pretopological approach*, which does not assume much, the other two approaches assume that the target vector spaces are endowed with traditional favorable properties. The geometry is simply imposed beforehand by the nature of the Euclidean distance between reduced descriptions of objects, i.e., between vectors in a Euclidean space. The existence of a well-established theory for Euclidean metric spaces made researchers place the learning paradigm in that context[9], but sometimes, the fact that *reality* is not modeled properly is neglected.

In the particular case of the *pretopological approach*, the nearest neighbor rule does not use the dissimilarities between objects of the training set when classifying new objects, hence discarding the chance of gaining knowledge from them. As a consequence, this approach is usually outperformed by the other two approaches[45,9]. Also, this rule relies on the fact that the representation set is properly chosen and a poor selection of elements in R might lead to unreliable results.

The *embedding approach*, as mentioned before, has its learning paradigms established in vector spaces were certain properties are met. It also requires the symmetry of the dissimilarity measure and, even though any asymmetric dissimilarity measure can be transformed into a symmetric one [9], there are scenarios where non-Euclidean dissimilarities could be relevant [46,47,48]. Moreover, usually an eigendecomposition problem must be solved[36], which could be numerically unstable and expensive for large data sets. Next, the *dominant*[14] eigenvalues must be selected, thus determining the dimension of the embedded space. This procedure allows for a dimensionality reduction but some *relevant* information could be lost in the process. It is important to emphasize that in many cases this selection of *meaningful* eigenvalues is often assessed by the user[9].

The *dissimilarity space approach*, just like the *embedding approach* interprets the dissimilarities in the context of vector spaces. In this approach, a proper selection of the representation set R induces a compact space where any traditional classifier that works in vector spaces can be used. Moreover, due to the selection of a small and representative set of prototypes, the problem of the *curse of dimensionality* is usually avoided. Nonetheless, the construction of classifiers in the dissimilarity space does not use the fact that they deal with dissimilarities instead of features or attributes[36]. Therefore, the *dissimilarity space approach* might not be making the most out of the dissimilarity data either, although it is the most widely used approach in the literature.

In many practical applications, the domain of real world objects might not *follow the rules* of the previously mentioned spaces. For instance, graphs are usually compared by user defined distance measures that are sometimes optimized over the training set in the nearest neighbor scenario [36]. In this context,

[14] Usually the largest p positive and the smallest q negative eigenvalues are chosen.

it might be worth analyzing the suitability of alternative representations for the dissimilarity information that make no further assumptions: for instance, structural representations like graphs or tree models[9].

Most of the arguments given so far have been analyzed in the context of supervised classification. However, not all of them might be valid in an unsupervised environment. Regarding the pretopological approach, several successful unsupervised classification algorithms have been proposed. Within an embedding approach, the computation of the whole embedding (one of the drawbacks of this approach) with the arrival of every new sample for classification might not be needed in unsupervised classification tasks. Therefore, the consideration of alternative representations and the *proper* selection of the learning framework for unsupervised classification tasks are still topics yet to be studied in greater depth.

3.3 Dissimilarity Measures for Spectral Data

Due to the richness and wide variety of real world objects, there is no dissimilarity measure that fits them all. For each problem at hand, and particularly for spectral data, a dissimilarity measure specifically designed for that type of data is preferred. In the case of spectral data, for example, the shape of peaks may be taken into account. In this section, some of the most relevant studies carried out concerning dissimilarity measures for the dissimilarity representation of spectral data are presented.

3.3.1 Measures for 1D Spectral Data

As mentioned before, 1D spectra are usually represented in a feature space spanned by the spectral bands (see figure 12). A significant drawback of this approach is that it usually faces the problem of the *curse of dimensionality* due to the high feature space dimensionality induced by the high resolution of spectral data. Hence, an exponential amount of training data is often required to design proper classifiers.

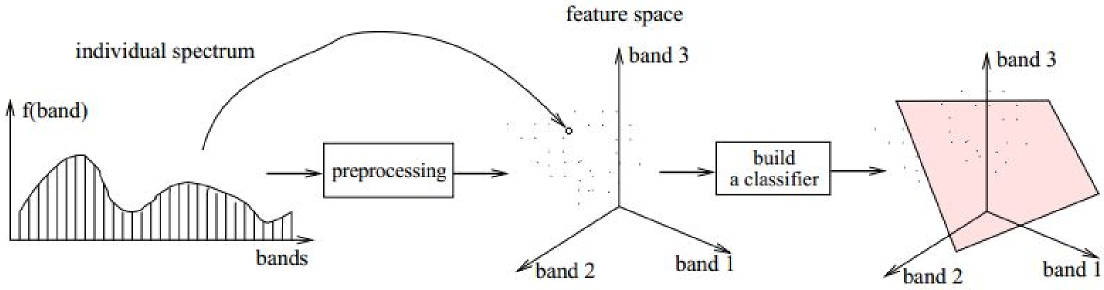


Fig. 12. (Source: [49]) Feature-based representation of spectral data.

Another approach is to represent spectra by dissimilarities to other spectra. A dissimilarity measure specifically designed for spectral data could be able to highlight relevant characteristics like their structures (shape changes) and/or concentration or intensity changes. An important difference between both approaches is that the former represents each spectrum in an absolute way while the later relates the observations to each other (see figure 13).

Some dissimilarity measures are more commonly used in the comparison of spectral data[15] like the very well known Manhattan (L1-norm) and Euclidean distances. However, these measures somehow neglect the key aspects of spectral data. There are several proposals in the literature addressing particularly spectral data.

[15] Mostly for chemical spectral data.

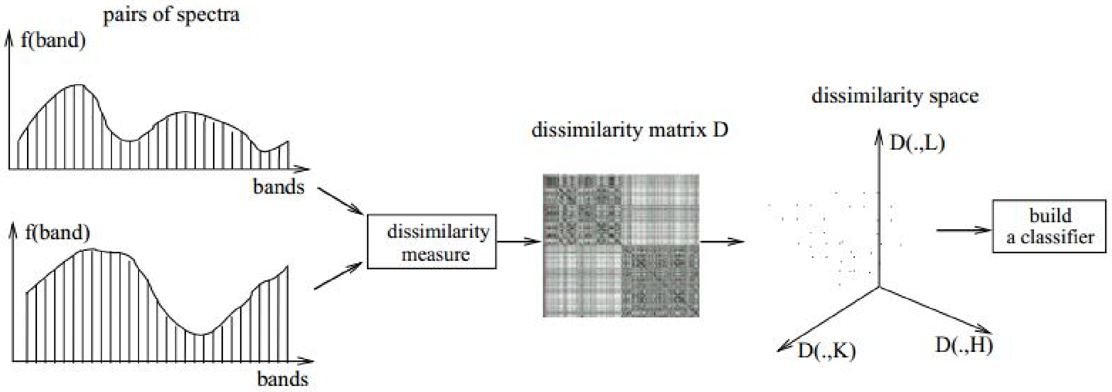


Fig. 13. (Source: [49]) Relative representation of spectral data using dissimilarities.

Let H and K be two spectra with N spectral bands each, $\omega = 1, \cdots, N$. In [50], the Spectral Angle Mapper (SAM) measure (see equation 4) was proposed for spectral data, which is computed as follows:

$$d_{sam}(H,K) = \arccos\left(\frac{\sum_{\omega=1}^{N} h_\omega k_\omega}{\sqrt{\sum_{\omega=1}^{N} h_\omega^2 \sum_{\omega=1}^{N} k_\omega^2}}\right) . \tag{4}$$

Another dissimilarity measure (see equation 5) that also takes into consideration the angle between the two *vectors* has been proposed. This measure is based on the Pearson Correlation Coefficient (PCC). The PCC can also be seen as the cosine of the angle between two mean-centered spectra. Although SAM and PCC are among the most used measures in the comparison of chemical spectral data, the connectivity/ordering of variables is not taken into account in either of them. For the particular case of spectral data, encoding such connectivity existing among variables is key, otherwise significant shape changes in one of the spectra might not be properly reflected in the dissimilarity measure

$$d_{pcc}(H,K) = 1 - \left(\frac{\sum_{\omega=1}^{N} (h_\omega - \bar{H})(k_\omega - \bar{K})}{\sqrt{\sum_{\omega=1}^{N} (h_\omega - \bar{H})^2 \sum_{\omega=1}^{N} (k_\omega - \bar{K})^2}}\right) . \tag{5}$$

Another commonly used measure is the *Minkowski* distance (see equation 6). This measure is computed on a band-to-band basis, neglecting the connectivity in the spectral domain[51]. Minkowski distance between two spectra, as well as SAM and PCC, yields the same dissimilarity value even if all the bands are randomly permuted

$$d_{min}(H,K) = \left(\sum_{\omega} |h_\omega - k_\omega|^r\right)^{\frac{1}{r}} . \tag{6}$$

If spectra are normalized to a unit area, they could be compared using dissimilarity measures developed for probability distributions like the *Kolmogorov-Smirnov* distance (KS):

$$d_{ks}(H,K) = \max_{\omega}(|\hat{h_\omega} - \hat{k_\omega}|) , \tag{7}$$

where $\hat{h_\omega}$ and $\hat{k_\omega}$ are cumulative distribution functions, $\hat{h_i} = \sum_{j \le i} h_j$, similarly for $\hat{k}$. This measure takes into account the distribution shape as it compares the areas under original spectra[51].

In [49], the authors propose to compute the *Manhattan* measure on the first Gaussian derivatives of the curves[16], to take into account the shape information that can be obtained from the derivatives. Shape information is obtained by computing the first Gaussian derivative:

$$H^{\sigma} = \frac{d}{d_{\omega}} G(\omega,\sigma) * H \ , \tag{8}$$

where $*$ denotes the convolution operator and σ stands for a smoothing parameter. The proposed *Shape* dissimilarity measure is then a sum of absolute differences between spectra derivatives H^{σ} and K^{σ} (see equation 9). Good performances have been obtained for chemical spectral data with this measure[45,10]

$$d_s(H,K) = \sum_{\omega} (|h^{\sigma}_{\omega} - k^{\sigma}_{\omega}|) \ . \tag{9}$$

3.3.2 Measures for 2D Spectral Data

In many scenarios, particularly in multi-way analysis, the objects are often described as 2D *signals*[17]. For 2D or any-dimensional continuous data[18], their continuous (functional) nature or shape changes in the surface could be relevant aspects in the analysis. Some 2D measures have been proposed for image comparison[19]; however most of them are just based on the pairwise comparison of objects, ignoring the continuous nature of images like Assembled Matrix Distance (AMD)[52], Frobenius[53], and others.

With the goal of maintaining the notation, H and K are considered now as 2D signals with $l = 1,\cdots,L$ rows and $m = 1,\cdots,M$ columns. The AMD distance can be computed as:

$$d_{amd}(H,K) = \left(\sum_{l=1}^{L} \left(\sum_{m=1}^{M} (h_{lm} - k_{lm})^2 \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} \ , \tag{10}$$

where the weight p is used to emphasize either small or large differences between the elements. If $p < 1$ the largest differences will be reduced. On the other hand, if $p > 1$, the larger differences will be more pronounced.

Later in [54], the *2DShape* dissimilarity measure was proposed, which incorporates the use of first Gaussian derivatives into the AMD measure. In that way, the authors can take the ordering information into account as well as the shape of the spectra[20]. The *2DShape* it defined in three stages: (1) compute the matrix D^1 (see equation 11), (2) compute the matrix D^2 (see equation 12) and (3) combine both dissimilarity matrices (see equation 13)

$$D^1 = \left(\sum_{l=1}^{L} \left(\sum_{m=1}^{M} (h^{\sigma_1}_{l,m} - k^{\sigma_1}_{l,m})^2 \right)^{\frac{p_1}{2}} \right)^{\frac{1}{p_1}}, \quad h^{\sigma_1}_{l,\cdot} = \frac{d}{d_l} G(l,\sigma_1) * h_{l,\cdot}, \quad k^{\sigma_1}_{l,\cdot} = \frac{d}{d_l} G(l,\sigma_1) * k_{l,\cdot} \ , \tag{11}$$

$$D^2 = \left(\sum_{m=1}^{M} \left(\sum_{l=1}^{L} (h^{\sigma_2}_{l,m} - k^{\sigma_2}_{l,m})^2 \right)^{\frac{p_2}{2}} \right)^{\frac{1}{p_2}}, \quad h^{\sigma_2}_{\cdot,m} = \frac{d}{d_k} G(k,\sigma_2) * h_{\cdot,m}, \quad k^{\sigma_2}_{\cdot,m} = \frac{d}{d_k} G(k,\sigma_2) * k_{\cdot,m} \ , \tag{12}$$

[16] Known as *Shape* measure.

[17] And even in some cases by higher order signals.

[18] For example, aligned images, spectroscopic data, etc.

[19] In particular for face and palm-print recognition.

[20] This measure takes into account the information on both directions of the array

$$D = \alpha_1 D^1 + \alpha_2 D^2 \ . \tag{13}$$

The variables $h_{l,\cdot}$ and $k_{l,\cdot}$ stand for the l-th rows of the spectra H and K respectively. In the same way, the variables $h_{\cdot,m}$ and $k_{\cdot,m}$ correspond to the m-th columns of the spectra H and K. Their expressions correspond to the computation of the first Gaussian (that is what G stands for) derivatives of spectra. The dissimilarities in step 1 and step 2 correspond to the first and second directions respectively. This measure has a higher computational complexity and a trade off might be needed between computational complexity and classification accuracy[13].

Nonetheless, the 2DShape measure is based on the combination of 1D dissimilarities; meaning it does not analyzes the combined 2D shape changes. It was originally designed for data sets with features of different nature in the different directions, e.g., data with a continuous and a non-continuous direction. Recently in [55], a new dissimilarity measure named *continuous multi-way shape* (CMS) was proposed, which exploits the information on the whole structure of multi-dimensional continuous data. It is based on the differences between the gradients of objects, thus the shape changes of the surfaces are considered. Moreover, the CMS measure could be used in data with a small amount of missing values[55].

4 Clustering on Dissimilarity Representations

Cluster analysis (or simply *clustering*) plays an important role in a broad range of applications, from information retrieval to image segmentation. The clustering task can be considered as a division/partition of data into groups of *similar*[21] objects. Each group is called a cluster and it consists of objects that are somehow *similar* between themselves and *dissimilar* to objects belonging to other groups[22].

A *clustering*[23] is basically a set of the aforementioned clusters, usually containing all objects in the data set. Additionally, it may specify the relationship between the different groups or clusters. But, what is an optimal data partition and how to find it? How to decide whether a data partition is better or worse?. Although several criteria to evaluate a partition have been proposed[56], it is usually the user who must supply this criterion in such a way that the result of the clustering will fulfill specific application needs. For example, the user could be interested in finding representatives for homogeneous groups (prototype selection), in finding useful and suitable groupings (*useful* data classes) or in finding unusual data objects (outlier detection).

There is a very extensive material concerning cluster analysis among which we can mention works like [57,58,59,60,61,62,63,64], etc. According to [65], the notion of a *cluster* can not be precisely defined in itself, which is one of the reasons why there are so many clustering algorithms. Different researchers employ different cluster models, and for each of these cluster models again different algorithms have been proposed. Clustering algorithms can be further categorized according to several criteria: *partitional clustering* (exclusive clusters) vs. *hierarchical clustering* (nested clusters), *hard clustering* (each object belongs to a cluster or not) vs. *fuzzy clustering* (each object belongs to each cluster to a certain degree), *strict partitioning clustering* (each object belongs to exactly one cluster) vs. *strict partitioning clustering with outliers* (objects can also belong to no cluster, and are therefore considered outliers), and more. An extensive review can be found, for instance, in [63] and [66].

In spite of the great amount of strategies used in the literature to *define* the clusters, they all have something in common (a common factor): the underlying concept of data and cluster *similarity/dissimilarity*[67].

[21] The concept of *similar* is a relative concept depending on the application.

[22] This could be interpreted as labeling the objects of the data set.

[23] Also called *partition* in many scenarios. Both terms will be used interchangeably in this work.

An alternative approach to data clustering is referred to as *pairwise data clustering*[24][68], and it focuses precisely on clustering data characterized by its pairwise comparisons instead of the original objects. It is worth noting that the characteristics of the data set are hidden in these pairwise relations or proximity values[25][69], and it provides the ideal framework to cope with the clustering task in the context of the dissimilarity representation. Therefore, most of our efforts will be centered in the analysis of this kind of clustering algorithms.

On the other hand, some additional tasks could be tackled by means of a clustering task in the dissimilarity representation, for instance, tasks like *anomaly detection*[70] and *prototype selection*[44]. In the case of prototype selection, recent proposals[71,39] have shown the feasibility of incorporating the concept of *entropy* into this scenario. This selection of representative prototypes has a great impact in the performance and accuracy of classifiers under this representation, although these advantages might also apply to clustering. In light of this, incorporating *information theoretic* knowledge into this tasks seems beneficial and it might be worthwhile to do some research regarding this topic.

4.1 Adaptations from Traditional Clustering

In this section, clustering techniques derived for dissimilarity representations[26] will be discussed. As usual, dissimilarity representations will be interpreted in three frameworks: *pretopological approach*, *embedding approach* and *dissimilarity space approach*[9].

4.1.1 Pretopological Approach Adaptations

Under this paradigm, an intuitive idea is to assign objects with small dissimilarities to other objects or which are in close neighborhoods of the selected representatives[27] to the same cluster. This is in fact the core idea behind the adaptation of classical clustering algorithms for dissimilarity representations presented here. The proposals discussed here are divided into two groups: *hierarchical methods* and *partitioning methods*.

Hierarchical methods

In the context of hierarchical clustering, agglomerative methods are very popular[28]. Initially, every object is considered as a single cluster. The *closest* two clusters, according to a conceptual dissimilarity measure ρ are merged until all objects belong to one cluster or a specified number of clusters k is reached.

The key difference between the different versions of agglomerative algorithms relies in the *linkage* function, which determines how clusters are combined. Adaptations of existing *linkage* functions for dissimilarity representations will be discussed next. Let C_k and C_l be two clusters where $|C_k| = n_k$ and $|C_l| = n_l$. Let ρ_{kl} be a measure of dissimilarity between clusters C_k and C_l. The clusters can be combined by one of the following criteria [9]:

- **Single Linkage (SL).** The dissimilarity ρ_{kl} between two clusters is the dissimilarity between their nearest neighbors computed as $\rho_{kl} = \min_{p_i \in C_k, p_j \in C_l} d(p_i, p_j)$. This rule emphasizes cluster connectedness, resulting in elongated, chain-like clusters.

[24] Also referred in the literature as *proximity based clustering*.

[25] They often violate the requirements of a distance measure, i.e., the triangular inequality does not necessarily hold, the self-dissimilarity may not vanish, etc.

[26] And in general for pairwise proximity data.

[27] Prototypes of the representation set.

[28] Divisive ones are not too popular probably due to their high computational burden, which is $O(2^n)$.

- **Complete Linkage (CL).** The dissimilarity ρ_{kl} is defined by the dissimilarity of furthest neighbors of the two clusters computed as $\rho_{kl} = \max_{p_i \in C_k, p_j \in C_l} d(p_i, p_j)$. This usually performs well when the objects form naturally distinct clouds, since it emphasizes the compactness. It is inappropriate if the clusters are somehow elongated or of a chain type.
- **Average Linkage (AL).** The dissimilarity ρ_{kl} is the average between-cluster dissimilarity computed as $\rho_{kl} = \frac{1}{n_k n_l} \sum_{p_i \in C_k} \sum_{p_j \in C_l} d(p_i, p_j)$. This performs well in both cases, when the objects form natural distinct clouds and when they form elongated clusters. It tends to produce clusters of a similar spread (or variance in a vector space).
- **Density Linkage.** This criterion derives a new dissimilarity d_{dens} based on the density estimates and adjacencies, which is then used by the single linkage clustering. For instance, in the k-nearest neighbor approach, the estimated density $f(p_i)$ at p_i is the number of objects within the k-nearest neighbor ball divided by its volume. The new d_{dens} is computed as $d_{dens}(p_i, p_j) = \frac{1}{2}\left(\frac{1}{f(p_i)} + \frac{1}{f(p_j)}\right)$ if $d(p_i, p_j) \leq max\{d_{k-NN}(p_i), d_{k-NN}(p_j)\}$ and ∞, otherwise.

So far, the previously discussed methods work directly on dissimilarities. Two other popular criteria require a Euclidean vector space representation, since they work with the estimated cluster means[72,73]:

- **Centroid Linkage.** The dissimilarity ρ_{kl} between two clusters is the square (Euclidean) distance between their estimated mean vectors, $\bar{x_k}$ and $\bar{x_l}$, $\rho_{kl} = d^2(\bar{x_k}, \bar{x_l}) = \left\|\bar{x_k} - \bar{x_l}\right\|_2^2$.
- **Ward's Linkage.** The two clusters selected to be merged are the ones that give the smallest increase in the intra-cluster sum of squares, which is the sum of the squared Euclidean distances between vectors and their cluster means, $\rho_{kl} = \frac{n_k n_l}{n_k + n_l} d^2(\bar{x_k}, \bar{x_l}) = \frac{n_k n_l}{n_k + n_l} \left\|\bar{x_k} - \bar{x_l}\right\|_2^2$. This tends to create clusters of similar sizes.

The centroid linkage and Ward's linkage can be generalized to the case when only dissimilarity representations are provided. This can be done in the three interpretation frameworks:

- **Generalized centroid linkage (GCL).** The extension of the *centroid linkage* may refer to:
  (a) *neighborhood relation*. Cluster centers are used instead of the vectors means. The centers c_k and c_l of the two clusters are defined as objects for which the maximum distance to all other objects within the clusters is minimum[29]. The conceptual dissimilarity becomes then $\rho_{kl} = d^2(c_k, c_l)$.
  (b) *embedded space*. The *square distance* between the cluster means can be approximated as $\rho_{kl} = \rho_{avg}(C_k, C_l) - \frac{1}{2}\rho_{avg}(C_k, C_k) - \frac{1}{2}\rho_{avg}(C_l, C_l)$ where ρ_{avg} is the average square dissimilarity $\rho_{avg} = \frac{1}{n_k n_l} \sum_{p_i \in C_k} \sum_{p_j \in C_j} d^2(p_i, p_j)$. This is the merging criterion.
  (c) *dissimilarity space*. The centroid linkage is applied in a dissimilarity space. The dissimilarity between the clusters is computed as $\rho_{kl} = \left\|\bar{\mathbf{d_k}} - \bar{\mathbf{d_l}}\right\|_2^2$, where $\bar{\mathbf{d_k}}$ and $\bar{\mathbf{d_l}}$ are the mean vectors of the two classes computed in a dissimilarity space $D(\cdot, R)$.
- **Generalized ward's linkage (GWL).** The extension of the *Ward's linkage* may refer to:
  (a) *neighborhood relation*. Cluster centers are used instead of the vectors means. Then, the dissimilarity between the clusters is computed as $\rho_{kl} = \frac{n_k n_l}{n_k + n_l} d^2(c_k, c_l)$.
  (b) *embedded space*. The *distance* of a single point to the mean of the cluster C_k in an embedded space is determined as $d^2(p_i, me_k) = \frac{1}{n_k} \sum_{p_j \in C_k} d^2(p_i, p_j) - \frac{1}{2n_k^2} S$. The term S is computed as $S = \sum_{p_z \in C_k} \sum_{p_t \in C_k} d^2(p_z, p_t)$ and the GWL criterion relies now on the estimated within-cluster sum $\sum_{p_i \in C_k} d^2(p_i, me_k) = \frac{1}{2n_k} S$[9].
  (c) *dissimilarity space*. The Ward's linkage is applied in a dissimilarity space.

[29] Clusters centers could have also be chosen as the objects which minimize the average square dissimilarity within the clusters.

Partitioning methods

Regarding the extensions of classical partition methods, the *k*-centers[74] (an extension of *k*-means[75]) and the *mode-seeking*[76] algorithms will be described in this section.

k-centers

The *k*-centers technique works directly on a dissimilarity representation $D(R,R)$. First, k objects evenly distributed with respect to the dissimilarity information are drawn from R. Next, the algorithm proceeds as follows:

1. Select an initial $J = \{p_1^{(j)}, p_2^{(j)}, \cdots, p_k^{(j)}\}$ of k objects chosen from R (for example, performing random selection).
2. Find the nearest neighbor in J for each $p_z \in R$. Let J_i be a subset of R consisting of objects that yield the same nearest neighbor $p_i^{(j)}$ in J, where $i = 1, 2, \cdots, k$. Hence, $R = \cup_{i=1}^{k} J_i$.
3. For each J_i find its center c_i, i.e., an object in J, for which the maximum distance to all other objects in J_i is minimum[30].
4. For each center c_i, if $c_i \neq p_i^{(j)}$, then replace $p_i^{(j)}$ by c_i in J. If any replacement is done, then return to step 2, otherwise STOP.

Except for step 3, this routine is identical to the *k*-means algorithm performed in a vector space. The result of the k-centers procedure heavily depends on initialization. To determine the initial set J of k objects, we start from a chosen center for the entire set and then, gradually, more centers are added until k centers are determined by means of splitting existing groups[31]. The entire procedure is repeated M times, resulting in M potential sets from which one yielding the minimum of the largest final subset radius is selected.

mode-seeking

This technique focuses on modes in dissimilarity data, which are determined by focusing on a specified neighborhood size s. The algorithm proceeds as follows:

1. Set a relative neighborhood size to an integer $s > 1$.
2. For each object $p_i \in R$ find the dissimilarity $d(p_i, \mathbf{nn}_s(p_i))$ to its s-th nearest neighbor.
3. Find a set J consisting of all $p_j \in R$ for which $d(p_j, \mathbf{nn}_s(p_j))$ is minimum within its set of s nearest neighbors.

The objects in J are the estimated modes of the class distribution in terms of the given dissimilarities. They are used to constitute the modes. The final number of clusters k depends on the choice of s[32], being the choice of this parameter s a drawback of *mode-seeking*.

Concerning the *k*-centers, hierarchical clustering and mode-seeking algorithms, a larger number of clusters is sometimes retrieved, as these methods suffer either from the presence of outliers (objects with large dissimilarities) or have difficulties to accommodate sparse clusters. Moreover, all these adaptations of traditional clustering algorithms suffer from the same problems as the original proposals. Therefore, some alternative strategies might be needed in order to overcome these problems.

4.1.2 Embedding Approach Adaptations

Symmetric dissimilarity representations can be embedded in complete (or approximated) Euclidean or pseudo-Euclidean spaces, where standard partition methods, such as the *k*-means and classifier-clustering can be used. By performing an approximate embedding, some information, possibly reflecting the noise in the data[33], is neglected. It is also possible to re-compute the dissimilarity representation derived from

[30] This value is called the radius of J.

[31] If the splitting continues until each group has one element, this procedure might become a divisive algorithm.

[32] The larger s, the smaller k.

[33] However, relevant information could be discarded.

the approximate embedding, dealing now with a more discriminative dissimilarity representation, which can be further used by neighborhood-based clustering approaches.

4.1.3 Dissimilarity Space Approach Adaptations

Due to the fact that dissimilarity representations in this approach are interpreted in a *dissimilarity* vector space, traditional clustering algorithms designed for vector spaces can be applied. It is often recommended to use a reduced representation $D(T,R)$ of the data set both from an efficiency (time complexity) and a representational (using only informative objects as representatives) point of view.

The representation set R can be selected randomly or by using procedures like k-centers or mode-seeking. Alternatively, principal components could be retrieved from the dissimilarity space (treating it as a usual vector space). The resulting space is referred to as a *PCA-dissimilarity space*. However, if the dissimilarity between objects does not capture the cluster characteristics, the dissimilarity space will not be able to retrieve appropriate clusters. An alternative solution is to consider a flexible non-linear monotonic transformation of the given dissimilarities aiming to emphasize the importance of local neighborhoods and diminish the influence of outliers.

4.2 Alternative Strategies

There are a number of approaches in the literature to cope with the task of *pairwise data clustering* or *proximity based clustering*. However, none of them is perfect and thus, it is necessary to compromise between their pros and cons. Next, some of the most relevant strategies will be discussed.

4.2.1 Optimization-based Clustering

A systematic approach to pairwise clustering by objective functions is based on an axiomatization of invariance properties and robustness for data grouping. As a consequence of this axiomatic approach the discussion is usually restricted to intra-cluster criteria. In [77,78,69,79], an interesting approach to a general proximity-based (neighborhood-based) partitioning (both hierarchical and partitioning) is proposed, where the clustering task is formulated as a combinatorial optimization problem. In these works, an objective function is specified, incorporating a suitably weighted average of the within-cluster and between-cluster dissimilarities.

In [77], the pairwise clustering task is modeled as a combinatorial optimization problem depending on *boolean* assignments $M_{iv} \in \{0,1\}$ of sample i to cluster v (this is an *NP-Hard*[80] combinatorial optimization problem). The cost function for pairwise clustering is defined where only pairs of data points assigned to the same cluster contribute to the total cost and dissimilarities between data which belong to different clusters are not counted.

The discovery of structure in data sets can be achieved by embedding the data in a d-dimensional space where the dissimilarities of pairs of data points are approximated by distances in a Euclidean space, forcing the symmetry of the dissimilarity measure, which is not always appropriate. An approximate solution is formulated in the maximum entropy framework using a variational principle to derive corresponding data partitionings in the Euclidean space. This approximation solves the embedding problem and the grouping of these data into clusters simultaneously and in a self consistent fashion. However, the learning framework is still the *embedding approach* and the analysis performed earlier also apply in this case.

In [78], an extension of the idea in [77] is proposed to perform hierarchical clustering. The algorithm proposed in this work belongs to the class of *deterministic annealing* algorithms, a deterministic variant of the well-known meta-heuristic *simulated annealing*[81]. Deterministic annealing algorithms favor parallel implementations and they are scalable with respect to the quality of the solution and complexity trade off.

In [69], as in [78], the proposal is based on a deterministic annealing approach that simultaneously embeds data in a Euclidean vector space, which can be used for dimensionality reduction and data visualization. Here, the dependencies between the clustering procedure and the Euclidean representation are still reflected. However, a modified cost function is proposed with the advantage of not enforcing the symmetry of the dissimilarity measure. In addition, the characteristics of the original cluster structure are preserved much better than the classical *Multidimensional Scaling* (MDS) techniques[68,82] with subsequent central clustering.

An approach that covers the case of sparse proximity matrices, and is extended to nested partitionings for hierarchical data clustering is proposed in [79]. As the aforementioned works, it is based on a rigorous mathematical framework for solving the associated optimization problem and this solution is reached by means of heuristic procedures. The objective functions proposed so far for data grouping focus on intra cluster criteria. However there are situations, where the exclusive focus on compactness fails to capture essential properties of the data and these approaches are not well suited for the detection of elongated structures in the object domain[83].

Another idea, proposed in [83], discusses a path-based pairwise clustering, which emphasizes the within-cluster connectivity by the use of graph methods. Intuitively, two objects are considered as similar if there exists a within-cluster path between them without any edge of a large dissimilarity. In fact, the effective dissimilarity between objects is defined as the largest edge cost on the minimal intra-cluster path connecting both objects. As a result, a new dissimilarity emphasizing connectedness rather than compactness is developed that is further used for grouping using a cost function. In addition, a new multiscale optimization scheme for the new clustering approach has been developed. Nonetheless, there is a large design freedom in this new multi-scale optimization scheme, given the fact that new concepts like the one for dissimilarity matrix of a *coarser level*[83] and a *mapping*[83] for objects between levels must be defined.

In spite of the fact that there is a well grounded theory supporting the optimization based proposals, such models could be very complex. They are not guaranteed to find the global minimum, although the solutions found by approximation algorithms[80] are often acceptable.

4.2.2 Evidential Clustering

Another proximity-based algorithm, called *evidential clustering* (EVCLUS) is proposed in [84] and relies on the evidence theory[34]. In this approach, a *basic belief assignment* (bba) is assigned to each object over a given set of groups, in such a way that the degree of conflict between two bbas reflects the dissimilarity between the corresponding objects. The set of bbas forms a *credal partition*, which contains a lot of information, so that it may be useful to summarize it in the form of a fuzzy or crisp partition. The *credal partition* may also be viewed as a rich and general model of partitioning, from which fuzzy and hard partitions can be computed as by-products.

This approach allows for an additional flexibility, resulting both in greater expressive power and in improved robustness with respect to atypical data. It also allows us to combine clustering results obtained from several dissimilarity matrices provided by different experts or measurement devices. In conjunction with *dissimilarity increments*[35], it could be used for clustering ensemble tasks in the context of dissimilarity representations as proposed in [85].

[34] The theory of belief functions.

[35] Concept that will be discussed later in this work.

It is known that EVCLUS performs well[84], provided that a suitable trade off parameter λ[36] is chosen. However, if the parameter λ deviates from the optimal value, very bad results are found. Also, EVCLUS is initialization-sensitive and is characterized by a high computational burden. The number of parameters to be optimized is linear in the number of objects but exponential in the number of clusters. Hence, computational problems may quickly arise when the number of clusters increases.

4.2.3 Spectral Clustering

A promising alternative that has been applied in a number of fields is to use *spectral methods* for clustering. In these methods, the *top* eigenvectors of a matrix derived from the distance measure (which could also be a dissimilarity measure) between points are used. A couple of works are briefly discussed next.

In [86], an algorithm based on Laplacian Eigenmaps is proposed, whose locality preserving characteristics make it relatively insensitive to outliers and noise. A by-product of this is that the algorithm implicitly emphasizes the natural clusters in the data. In [87], the concept of the eigenvector is seen as the solution of a relaxation of an *NP-Hard* discrete graph partitioning problem[88]. The *second* eigenvector is used to give a guaranteed approximation of the optimal cut in a graph[89,88]. The analysis is extended to clustering by building a weighted graph in which the nodes correspond to data points and edges are related to the distance between the points.

Despite many empirical successes of *spectral clustering* methods[37], there are several unresolved issues. There are a wide variety of algorithms that use the eigenvectors in slightly different ways and there has been a disagreement on exactly which eigenvectors to use and how to derive clusters from them[90]. One must not forget that solving an eigendecomposion problem could be computationally expensive and some numerical problems might arise. Moreover, there is no proof that many of these algorithms will actually compute a *reasonable* clustering.

4.2.4 Clustering by Means of Dissimilarity Increments

As seen in the clustering methods discussed before, patterns (or elements) are usually assigned to a cluster according to some proximity measure. The choice of such measure can be difficult, since no prior information about cluster shapes or structure is available. Most of the clustering techniques found in the literature use pairwise dissimilarities between patterns to perform that assignment, however, a *high-order* dissimilarity measure has been proposed[91]: the *dissimilarity increments*. The use of this measure gives different information than the use of pairwise dissimilarities. Since a cluster is a set of patterns sharing some characteristics, the dissimilarity increments between neighboring patterns should not occur with abrupt changes and the dissimilarity increments between well separated clusters will have higher values[92].

In [93][38], the proposed procedure is a partitional procedure that intrinsically identifies the number of clusters without necessity of *a priori* knowledge of design parameters. Although it provides a dendrogram type graph describing the structure of data, an *isolation criterion* is proposed to determine when a cluster is *correctly defined*. Nonetheless, the structure of the resulting dendrogram is conditioned by the cluster isolation criterion[67] and thus, it should be interpreted differently than traditional dendrograms. For instance, a *cut* in the dendrogram does not correspond to a data partition. Also, the isolation criterion depends on the correct choice of certain parameters and if this choice is not adequate, some issues might arise like merging very different clusters, spurious clusters could be detected and one might face the phenomenon of *premature isolation*. A premature isolation of a cluster means that it is prematurely decided

[36] The purpose of λ is twofold: to control both the hardness of the resulting partition and the number of free parameters of the method.

[37] Algorithms that cluster points using eigenvectors of matrices derived from the data.

[38] This is the first work, to our knowledge, that used the concept of *dissimilarity increments*.

that this cluster can not be merged anymore, phenomenon that usually arises when clusters have a few (usually 10 or less) samples.

A major difficulty with hierarchical methods concerns its complexity, both in terms of time and space, limiting its range of applicability. The hierarchical agglomerative framework combined with dissimilarity increments proposed in [93] is extended in [94] to be able to handle large data sets by proposing the integration of sampling techniques into the clustering process. The high dimensional data set is mapped into a reasonable small number of prototypes, easily handled by the hierarchical clustering method. The whole data set is partitioned into a large number of small and compact clusters, representing each cluster by its centroid; centroids are then clustered using the hierarchical clustering technique. The final data partition is obtained by joining all the patterns represented by the prototypes gathered in the same cluster. However, in this procedure small natural clusters could be merged and thus, each formed cluster should go through a more detailed analysis in order to detect finer grained clusters. In spite of the gain in efficiency, this proposal suffers from the rest of the limitations of the previous one. Moreover, this new proposal depends on the initialization procedure and minor degradations of clustering results could be observed.

The approach in [67] somehow mitigates the limitation of previous works of having a weak estimate characterizing cluster statistics in the presence of few samples. The problem of unreliable estimates of dissimilarity increments is addressed by proposing a more adequate dynamic threshold that compensates the effect of under-estimation of *gaps* statistics in the early stages of the clustering algorithm. However, this strategy is still sensitive to noise[91] and may lead to drastic changes in pattern associations, which could be reflected in distinct topology dendrograms.

In another work, [95], a new algorithm is proposed based on Gaussian Mixture Decomposition[96] (GMD) as a starting point for the hierarchical framework discussed before. A new statistical model for the *dissimilarity increments distribution* for 2-dimensional data (2-DID)[39] is also proposed, which was assumed by the authors to follow an exponential distribution[40]. Another contribution of this work is a strategy to choose the parameter that guides the merge of clusters relying on the Graph-based Dissimilarity Increments Distribution index[97]. However, there are some drawbacks of this proposal. The initialization procedure based on GMD presents some problems, it is unable to detect clusters of arbitrary shapes and has the tendency to capture more than one cluster into one gaussian component if they are located near each other, even if they are separable[92].

In order to overcome the difficulties caused by the gaussian mixture decomposition, a hierarchical algorithm combined with the concept of dissimilarity increments, named SL-DID, is proposed in [92]. The proposed algorithm, based on the single linkage method, is parameter-free and can identify classes as the union of clusters following the dissimilarity increments distribution. The key difference with single linkage is that using single linkage the most similar pair of clusters at each iteration is always merged, while in SL-DID some tests are made using the dissimilarity increments distribution, whose results determine whether that pair of clusters is merged or not. Due to this merge strategy, SL-DID can adequately identify well separated clusters with arbitrary shape and densities, and it offers a deeper insight into the structure of touching clusters[92]. However, we believe that the merging criterion is based on specific empirical knowledge derived from experiments and a more profound analysis should be carried out.

In [98], a statistical model for dissimilarity increments for d-dimensional data is derived[41] assuming that the objects follow a multivariate gaussian distribution (d-DID)[42]. By means of experiments, empirical evidence show that the derived distributions 2-DID and d-DID are a better approximation to the empirical

[39] All this under mild approximations.

[40] This assumption was made by means of visual inspection in previous works.

[41] Also under mild approximations.

[42] This type of distribution imposed on the data is not always true.

distribution than the exponential distribution considered in [91]. Even though d-DID can be well approximated by 2-DID[43], it is very computationally expensive and numerically unstable for large values of d. Still an unresolved issue of the method proposed, is that it still relies on an initial partition, which was generated by GMD or by k-means in this case.

Recently in [99], a family of agglomerative hierarchical methods, called Hierarchical Clustering based on Dissimilarity Increments Distribution (HCDID) was proposed. This family of agglomerative hierarchical methods integrate the concept of dissimilarity increments in traditional linkage algorithms. These algorithms are able to find the number of clusters and experimental results showed that any algorithm from the proposed family outperforms the corresponding traditional one[99]. In spite of the advantages offered by these algorithms over the traditional ones, the algorithms from the HCDID family have some parameters that need to be set, which is still an open issue.

4.3 Clustering Ensemble on the Dissimilarity Representation

In a wide variety of fields, several measurements are often performed over the same set of objects with the goal of achieving a more comprehensive and robust understanding of the problem at hand. Real world objects could be represented in a variety of ways and several learning techniques could be applied. With the evolution of fields like spectral data acquisition each object is usually analyzed by means of several techniques (several spectral data are acquired) offering complementary information. If the dissimilarity representation is the representation method of choice, we might end up with several dissimilarity matrices (one for each kind of technique) describing the same data set. However, all this information might not be fully exploited if no proper way can be found to *combine* the results obtained for each technique.

In the context of unsupervised classification, it is commonly desired to fuse the different clustering results obtained for the same data set. This process of combining different clustering results is usually known as clustering ensemble and this idea emerged as an alternative approach for improving the results of individual clustering algorithms. Every clustering ensemble method consists of two main steps: the *generation step* (the creation of a set of partitions of the objects) and a *fusion step* (computation of a new partition integrating the partitions obtained in the generation step)[2]. A peculiarity of clustering ensemble methods is that they should take into account the peculiarities of the problem at hand. Moreover, it can not be stated that clustering results obtained by a clustering ensemble method are *better* than the ones that were originally *combined*, it can only be ensured that the new clustering is a consensus of all the previous ones.

In our particular scenario with several dissimilarity representations, the combination of unsupervised classification results could be addressed by traditional clustering ensemble techniques. The key difference in this approach is the application of clustering algorithms designed specifically to deal with dissimilarity information. In the same way, traditional *generation mechanisms* and *consensus functions* could be used once the clustering results are provided. However, as traditional clustering ensemble techniques usually require specific application knowledge, we believe that some relevant information somehow encoded in the dissimilarity representations might be neglected in the ensemble process.

To our knowledge, practically no works have been reported concerning clustering ensemble methods in the context of the dissimilarity representation. However, due to the close relation existing between the combination of dissimilarity measures[100] (or their transformed versions) and the area of combining classifiers[36], a feasible strategy could be to find a *suitable combination* of such dissimilarity representations. Several criteria could be applied here, from stacking the dissimilarity matrices, to find a *proper*

[43] Which is in fact done.

combination of such matrices by means of techniques like a weighted combination, information theoretic based criteria, and more. It is important to mention that this strategy of combining the dissimilarity information has the advantage of taking into account specific domain knowledge of the problem at hand in the ensemble process.

4.4 Multi-way Clustering

As stated before in this work, clustering is a well-studied problem. However, most clustering algorithms are based on the assumption that the data set is represented as a two-way array. This problem arises in many scenarios in the context of multi-way data and usually the goal is to obtain meaningful clusters in one of the modes based on all the chosen variables. To our knowledge, not too much attention has been paid to this subject, although there are some works concerning the clustering task in multi-way data. The relevant works regarding this subject in the literature are discussed next.

Some schemes have been proposed for cluster analysis in three-way data sets. The underlying principle in these techniques is to cluster all modes of the data array simultaneously. It is important to stress that most of these proposals have been designed and tested in specific domains and it is not always straightforward (or even possible in some cases) to extend them to any higher-order array.

In [101], an algorithm named TriCluster combines subspace clustering with graph-based approaches to capture coherent clusters in three-way arrays[44]. Fixing one of the modes (preferably the *smallest* one), the algorithm constructs a *range multigraph*[45] and then it searches for *constrained maximal cliques* (maximal cliques under some constraints) in this multigraph to yield a set of *biclusters*[102] for that particular slice. Then, another graph is built using the biclusters (as vertices) obtained for each slice and the final set of *triclusters* is obtained by finding cliques in this last graph. Depending on different parameter values, TriCluster can find arbitrarily positioned and overlapping clusters. However, biclustering is known to be a *NP-Hard* problem[102], and thus many proposed algorithms use heuristic methods or probabilistic approximations, which could affect the accuracy of the final clustering results.

Multi-way distributional clustering (MDC)[103] relies on subspace clustering and introduces an extension of two-way clustering to multi-way arrays. The proposed algorithm focuses on document clustering and interleaves top-down clustering of some variables and bottom-up clustering of the other variables, with a local optimization correction routine.

The clustering method proposed in [104] makes use of an exploratory visualization approach on three-way arrays. The cluster of samples in the object mode is achieved by clustering the scores (sample loadings) obtained from a multi-way decomposition model, for example, a Tucker 3 or a PARAFAC model. However, hierarchical clustering techniques are applied based on standard similarity measures over the scores[46], which are not always the best choice. This approach is particularly appropriate in situations where these models convey a meaningful description of the data, therefore the *quality* of the model should be assessed first. Moreover, it generalizes straightforwardly to higher-order data and also to two-way data. In spite of all that, this proposal relies on the approach of learning through the scores computed by a multi-way decomposition model (multi-way decomposition + two-way learning), which means it still has its same limitations.

As previously discussed, there are many procedures for unsupervised classification in the case of two-way data, but there is a limited number of techniques for clustering multi-way profile data. However, if the dissimilarity representations for multi-way data proposed in [13] are employed, a new alternative

[44] It was designed particularly for (and tested in) microarray data sets.

[45] A compact representation of all similar value ranges between any two sample columns.

[46] For example, Euclidean, Mahalanobis, and others.

methodology for clustering multi-way objects can be derived. Once the multi-way array is represented by means of dissimilarity information, traditional clustering algorithms for such proximity representation could be applied. This new approach has all the advantages of the dissimilarity representation and we believe it is worth analyzing.

5 Conclusions

In this work, relevant aspects regarding the task of clustering spectral data have been carefully analyzed. Due to the many theoretical and practical reasons addressed here, the dissimilarity representation emerged as an alternative, and very suitable strategy to represent spectral data (few high dimensional data in general) for further supervised and unsupervised learning tasks.

As already described, in the context of the dissimilarity representation some attention has been paid in order to find a *proper* representation of spectral data by means of proximity data. Dissimilarity measures proposed in the literature have been mainly studied in supervised classification tasks, although they can also be used in unsupervised classification environments. However, we believe that some progress can still be made in this sense. In the process of designing a *good* dissimilarity measure there is still an open question: what makes two spectra similar/dissimilar? According to our research, there could be some room for improvement in topics like dealing with abrupt shape changes in the spectra, variable amounts of missing data, spectral misalignment, and more.

In our review of key aspects of the dissimilarity representation two issues were identified for unsupervised environments: the *proper* selection of the learning framework and the possibility of exploring alternative representations of dissimilarity information. In the case of alternative representations of structural nature like graphs and more specifically trees, they have not been completely exploited for learning tasks under the dissimilarity representation. Besides the fact that this particular representation does not *impose* any geometry on the data, relevant information can still be extracted, not to mention the improvement on the spatial complexity of the representation.

Concerning the status of clustering methods developed for proximity representations, it could be observed that even though there are several methods proposed in the literature, not too many classical clustering procedures have been extended to work with objects represented by pairwise characterizations. In our opinion, clustering methods defined for classical vector representations could be extended to work with this kind of data. Moreover, the concept of *high-order dissimilarities* could be further explored since it could offer a *better* characterization of the space were the samples lie in. This could be advantageous given the fact that pairwise data clustering algorithms do not fully exploit the fact that they are dealing with dissimilarity information. We also recommend to do some research concerning the application of these (and also new) pairwise clustering algorithms in related tasks such as *cluster-based outlier detection* and *cluster-based prototype selection* for the dissimilarity representation.

Another open issue detected in our investigation is the subject of clustering ensemble in the context of the dissimilarity representation. In our opinion, this area has not been explored and we think good results could be obtained from a *proper combination* of different dissimilarity information. We also propose to analyze with greater depth the relation between this subject and the combination of classifiers.

Regarding the clustering of multi-way spectral data, we reached the conclusion that this topic is practically unexplored. In view of its importance, we believe that this topic should not be neglected. In view of the very recent and promising findings in the area of multi-way classification by means of the dissimilarity representation[13], we propose to begin with the application of these results in the context of clustering.

References

1. Kittler, J., Roli, F.: Multiple classifier systems. Springer (2000)
2. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. International Journal of Pattern Recognition and Artificial Intelligence **25**(03) (2011) 337–372
3. Kroonenberg, P.M.: Applied multiway data analysis. Volume 702. John Wiley & Sons (2008)
4. Smilde, A., Bro, R., Geladi, P.: Multi-way analysis: applications in the chemical sciences. John Wiley & Sons (2005)
5. Fukunaga, K., Hayes, R.R.: Effects of sample size in classifier design. Pattern Analysis and Machine Intelligence, IEEE Transactions on **11**(8) (1989) 873–885
6. Raudys, S.J., Jain, A.K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis & Machine Intelligence (3) (1991) 252–264
7. Duin, R.P., Roli, F., de Ridder, D.: A note on core research issues for statistical pattern recognition. Pattern recognition letters **23**(4) (2002) 493–499
8. Silverman, B., Ramsay, J.: Functional Data Analysis. Springer (2005)
9. Pekalska, E., Duin, R.P.: The dissimilarity representation for pattern recognition: foundations and applications. Number 64. World Scientific (2005)
10. Porro, D., Duin, R.W., Talavera, I., Hdez, N.: The representation of chemical spectral data for classification. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer (2009) 513–520
11. Varmuza, K., Karlovits, M., Demuth, W.: Spectral similarity versus structural similarity: infrared spectroscopy. Analytica chimica acta **490**(1) (2003) 313–324
12. Gutiérrez-Rodríguez, A.E., Medina-Pérez, M.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., García-Borroto, M.: New dissimilarity measures for ultraviolet spectra identification. In: Advances in Pattern Recognition. Springer (2010) 220–229
13. Porro Munoz, D.: Classification of continuous multi-way data via dissimilarity representation. PhD thesis, TU Delft, Delft University of Technology (2013)
14. Porro, D., Duin, R.P., Talavera, I., Hernández, N.: Alternative representations of spectral data for classification. In: Proc. ASCI. (2009)
15. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Fourth Edition. 4th edn. Academic Press (2008)
16. Gourvénec, S., Stanimirova, I., Saby, C.A., Airiau, C., Massart, D.: Monitoring batch processes with the statis approach. Journal of chemometrics **19**(5-7) (2005) 288–300
17. Leardi, R., Armanino, C., Lanteri, S., Alberotanza, L.: Three-mode principal component analysis of monitoring data from venice lagoon. Journal of Chemometrics **14**(3) (2000) 187–195
18. Acar, E., Çamtepe, S.A., Krishnamoorthy, M.S., Yener, B.: Modeling and multiway analysis of chatroom tensors. In: Intelligence and Security Informatics. Springer (2005) 256–268
19. Estienne, F., Matthijs, N., Massart, D., Ricoux, P., Leibovici, D.: Multi-way modelling of high-dimensionality electroencephalographic data. Chemometrics and Intelligent Laboratory Systems **58**(1) (2001) 59–72
20. Cattell, R.B.: The three basic factor-analytic research designs-their interrelations and derivatives. Psychological bulletin **49**(5) (1952) 499
21. Tucker, L.R.: The extension of factor analysis to three-dimensional matrices. Contributions to mathematical psychology (1964) 109–127
22. Kiers, H.A.: Some procedures for displaying results from three-way methods. Journal of chemometrics **14**(3) (2000) 151–170
23. Harshman, R.A.: Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis. (1970)
24. Cattell, R.B.: Parallel proportional profiles and other principles for determining the choice of factors by rotation. Psychometrika **9**(4) (1944) 267–283
25. Acar, E., Yener, B.: Unsupervised multiway data analysis: A literature survey. Knowledge and Data Engineering, IEEE Transactions on **21**(1) (2009) 6–20
26. Harshman, R.A.: Parafac2: Mathematical and technical notes. UCLA working papers in phonetics **22**(3044) (1972) 122215
27. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3) (1966) 279–311
28. Salvatore, E., Bevilacqua, M., Bro, R., Marini, F., Cocchi, M.: Classification methods of multiway arrays as a basic tool for food pdo authentication. Food Protected Designation of Origin: Methodologies & Applications, Wilson & Wilson's Comprehensive Analytical Chemistry **60** (2013)
29. Andersson, C.A., Bro, R.: The n-way toolbox for matlab. Chemometrics and Intelligent Laboratory Systems **52**(1) (2000) 1–4
30. Arancibia, J.A., Boschetti, C.E., Olivieri, A.C., Escandar, G.M.: Screening of oil samples on the basis of excitation-emission room-temperature phosphorescence data and multiway chemometric techniques. introducing the second-order advantage in a classification study. Analytical chemistry **80**(8) (2008) 2789–2798

31. Durante, C., Bro, R., Cocchi, M.: A classification tool for n-way array based on simca methodology. Chemometrics and Intelligent Laboratory Systems **106**(1) (2011) 73–85
32. Wold, S., SJÖSTRÖM, M.: Simca: a method for analyzing chemical data in terms of similarity and analogy. (1977)
33. Edelman, S., Duvdevani-Bar, S.: Similarity, connectionism, and the problem of representation in vision. Neural computation **9**(4) (1997) 701–720
34. Edelman, S.: Representation is representation of similarities. Behavioral and Brain Sciences **21**(04) (1998) 449–467
35. Edelman, S.: Representation and recognition in vision. Volume 3. MIT press Cambridge, MA (1999)
36. Duin, R.P., Pekalska, E., Paclik, P., Tax, D.: The dissimilarity representation, a basis for domain based pattern recognition. In: Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings, Cambridge, UK. (2004) 43–56
37. Plasencia, Y., Garcıa, E., Duin, R.P., Méndez, H., San Martın, C., Soto, C.: Dissimilarity representations for thermal signature recognition at a distance. In: 15th Annual Conf. of the Advanced School for Computing and Imaging. (2009) 1–7
38. Martínez-Díaz, Y., Méndez-Vázquez, H., Plasencia-Calaña, Y., García-Reyes, E.B.: Dissimilarity representations based on multi-block lbp for face detection. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer (2012) 106–113
39. De Marsico, M., Riccio, D., Mendez Vazquez, H., Plasencia Calana, Y.: Getsel: Gallery entropy for template selection on large datasets. In: Biometrics (IJCB), 2014 IEEE International Joint Conference on, IEEE (2014) 1–8
40. Satta, R., Fumera, G., Roli, F.: Exploiting dissimilarity representations for person re-identification. In: Similarity-Based Pattern Recognition. Springer (2011) 275–289
41. Satta, R., Fumera, G., Roli, F.: Fast person re-identification based on dissimilarity representations. Pattern Recognition Letters **33**(14) (2012) 1838–1848
42. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on **13**(1) (1967) 21–27
43. Goldfarb, L.: A new approach to pattern recognition. Progress in pattern recognition **2** (1985) 241–402
44. Plasencia-Calana, Y., Garcıa-Reyes, E., Duin, R.: Prototype selection methods for dissimilarity space classification. Technical report, Technical report, Advanced Technologies Application Center CENATAV (2010)
45. Paclık, P., Duin, R.P.: Dissimilarity-based classification of spectra: computational issues. Real-Time Imaging **9**(4) (2003) 237–244
46. Pekalska, E., Harol, A., Duin, R.P., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative. In: Structural, Syntactic, and Statistical Pattern Recognition. Springer (2006) 871–880
47. Laub, J., Roth, V., Buhmann, J.M., Müller, K.R.: On the information and representation of non-euclidean pairwise data. Pattern Recognition **39**(10) (2006) 1815–1826
48. Duin, R.P., Pekalska, E.: Non-euclidean dissimilarities: causes and informativeness. In: Structural, Syntactic, and Statistical Pattern Recognition. Springer (2010) 324–333
49. Paclik, P., Duin, R.: Classifying spectral data using relational representation. na (2003)
50. Yuhas, R.H., Goetz, A.F., Boardman, J.W.: Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm. In: Summaries of the third annual JPL airborne geoscience workshop. Volume 1., Pasadena, CA: JPL Publication (1992) 147–149
51. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International journal of computer vision **40**(2) (2000) 99–121
52. Zuo, W., Zhang, D., Wang, K.: An assembled matrix distance metric for 2dpca-based image recognition. Pattern Recognition Letters **27**(3) (2006) 210–216
53. Yang, J., Zhang, D., Frangi, A.F., Yang, J.y.: Two-dimensional pca: a new approach to appearance-based face representation and recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**(1) (2004) 131–137
54. Porro-Muñoz, D., Duin, R.P., Orozco-Alzate, M., Talavera, I., Londoño-Bonilla, J.M.: The dissimilarity representation as a tool for three-way data classification: a 2d measure. In: Structural, Syntactic, and Statistical Pattern Recognition. Springer (2010) 569–578
55. Porro-Muñoz, D., Duin, R.P., Orozco-Alzate, M., Bustamante, I.T.: Continuous multi-way shape measure for dissimilarity representation. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Springer (2012) 430–437
56. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R.: Model-based evaluation of clustering validation measures. Pattern Recognition **40**(3) (2007) 807–824
57. Hartigan, J.A.: Clustering algorithms. John Wiley & Sons, Inc. (1975)
58. Spath, H.: Cluster analysis algorithms for data reduction and classification of objects. Ellis Horwood, Ltd. Chichester, England (1980)
59. Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc. (1988)
60. Kaufman, L.R., Rousseeuw, P.: Pj (1990) finding groups in data: An introduction to cluster analysis. Hoboken NJ John Wiley & Sons Inc

61. Dubes, R.C.: Cluster analysis and related issues. In: Handbook of pattern recognition & computer vision, World Scientific Publishing Co., Inc. (1993) 3–32
62. Mirkin, B.: Mathematical classification and clustering, volume 11 of nonconvex optimization and its applications (1996)
63. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) **31**(3) (1999) 264–323
64. Kolatch, E.: Clustering algorithms for spatial databases: A survey. PDF is available on the Web (2001)
65. Estivill-Castro, V.: Why so many clustering algorithms: a position paper. ACM SIGKDD explorations newsletter **4**(1) (2002) 65–75
66. Berkhin, P.: A survey of clustering data mining techniques. In: Grouping multidimensional data. Springer (2006) 25–71
67. Fred, A.L.: Clustering based on dissimilarity first derivatives. In: PRIS. (2002) 257–266
68. Duda, R.O., Hart, P.E.: Pattern recognition and scene analysis (1973)
69. Hofmann, T., Buhmann, J.M.: Pairwise data clustering by deterministic annealing. Pattern Analysis and Machine Intelligence, IEEE Transactions on **19**(1) (1997) 1–14
70. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3) (2009) 15
71. De Marsico, M., Nappi, M., Riccio, D., Tortora, G.: Entropy-based template analysis in face biometric identification systems. Signal, Image and Video Processing **7**(3) (2013) 493–505
72. Anderberg, M.R.: Cluster analysis for applications. 1973. Academic, New York
73. Everitt, B., Landau, S., Leese, M.: Cluster analysis arnold. A member of the Hodder Headline Group, London (2001)
74. Ypma, A., Ligteringen, R., Frietman, E.E., Duin, R.P.: Recognition of bearing failures using wavelets and neural networks. In: Proceedings of the 2nd UK Symposium on Applications of Time-Freqency and Time-Scale Methods, Citeseer (1997) 69–72
75. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Volume 1., Oakland, CA, USA. (1967) 281–297
76. Cheng, Y.: Mean shift, mode seeking, and clustering. Pattern Analysis and Machine Intelligence, IEEE Transactions on **17**(8) (1995) 790–799
77. Buhmann, J., Hofmann, T.: A maximum entropy approach to pairwise data clustering. In: Pattern Recognition, 1994. Vol. 2-Conference B: Computer Vision &amp; Image Processing., Proceedings of the 12th IAPR International. Conference on. Volume 2., IEEE (1994) 207–212
78. Hofmann, T., Buhmann, J.M.: Hierarchical pairwise data clustering by mean-field annealing. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN 95). Springer, Citeseer (1995) 197–202
79. Puzicha, J., Hofmann, T., Buhmann, J.M.: A theory of proximity based clustering: Structure detection by optimization. Pattern Recognition **33**(4) (2000) 617–634
80. Cormen, T.H.: Introduction to algorithms. MIT press (2009)
81. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., et al.: Optimization by simulated annealing. science **220**(4598) (1983) 671–680
82. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Transactions on computers (5) (1969) 401–409
83. Fischer, B., Zöller, T., Buhmann, J.M.: Path based pairwise data clustering with application to texture segmentation. In: Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer (2001) 235–250
84. Denœux, T., Masson, M.H.: Evclus: evidential clustering of proximity data. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on **34**(1) (2004) 95–109
85. Aidos, H., Fred, A.: Dissimilarity increments distribution in the evidence accumulation clustering framework. In: Pattern Recognition and Image Analysis. Springer (2013) 535–542
86. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: NIPS. Volume 14. (2001) 585–591
87. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems **2** (2002) 849–856
88. Chung, F.R.: Spectral graph theory. Volume 92. American Mathematical Soc. (1997)
89. Spielmat, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. In: Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on, IEEE (1996) 96–105
90. Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Volume 2., IEEE (1999) 975–982
91. Fred, A.L., Leitã, J.: A new cluster isolation criterion based on dissimilarity increments. Pattern Analysis and Machine Intelligence, IEEE Transactions on **25**(8) (2003) 944–958
92. Aidos, H., Fred, A.: Hierarchical clustering with high order dissimilarities. In: Machine Learning and Data Mining in Pattern Recognition. Springer (2011) 280–293
93. Fred, A.L., Leitão, J.: Clustering under a hypothesis of smooth dissimilarity increments. In: Pattern Recognition, 2000. Proceedings. 15th International Conference on. Volume 2., IEEE (2000) 190–194
94. Fred, A.L.: Context-dependent clustering based on dissimilarity increments. In: In Proceedings of the Portuguese Conference on Pattern Recognition. (2002)

95. Aidos, H., Fred, A.: On the distribution of dissimilarity increments. In: Pattern Recognition and Image Analysis. Springer (2011) 192–199
96. Figueiredo, M.A., Jain, A.K.: Unsupervised learning of finite mixture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(3) (2002) 381–396
97. Fred, A.L., Jain, A.K.: Cluster validation using a probabilistic attributed graph. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, IEEE (2008) 1–4
98. Aidos, H., Fred, A.: Statistical modeling of dissimilarity increments for d-dimensional data: Application in partitional clustering. Pattern Recognition **45**(9) (2012) 3061–3071
99. Aidos, H., Fred, A.: A family of hierarchical clustering algorithms based on high-order dissimilarities. In: Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European, IEEE (2014) 1432–1436
100. Pekalska, E., Duin, R.P.: On combining dissimilarity representations. In: Multiple Classifier Systems. Springer (2001) 359–368
101. Zhao, L., Zaki, M.J.: Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ACM (2005) 694–705
102. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Ismb. Volume 8. (2000) 93–103
103. Bekkerman, R., El-Yaniv, R., McCallum, A.: Multi-way distributional clustering via pairwise interactions. In: Proceedings of the 22nd international conference on Machine learning, ACM (2005) 41–48
104. Acar, E., Bro, R., Schmidt, B.: New exploratory clustering tool. Journal of Chemometrics **22**(1) (2008) 91–100

RT_076, octubre 2015

Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona

Diseño de Portada: Di. Alejandro Pérez Abraham

RNPS No. 2142

ISSN 2072-6287

Indicaciones para los Autores:

Seguir la plantilla que aparece en www.cenatav.co.cu

CENATAV
7ma. A No. 21406 e/214 y 216, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200

Impreso en Cuba