Prototype Selection for Classification in Standard and Generalized Dissimilarity Spaces

Prototype Selection for Classification in Standard and Generalized Dissimilarity Spaces

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft;
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben;
voorzitter van het College voor Promoties
in het openbaar te verdedigen op 24 September 2015 om 15.00 uur

door

Yenisel PLASENCIA CALAÑA

Computer Science Licentiate van
Universiteit "Universidad de La Habana"
geboren te Havana, Cuba

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. M.J.T. Reinders
Copromotor: Dr.ir. R.P.W. Duin

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. M.J.T. Reinders,	Technische Universiteit Delft, promotor
Dr.ir. R.P.W. Duin,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr. E.B. García Reyes,	Advanced Technologies Application Center, Cuba
Prof.dr.ir. B.P.F. Lelieveldt,	Leids Universitair Medisch Centrum
Prof.dr.ir. P.P. Jonker,	Technische Universiteit Delft
Prof.dr.ir. B.J.A. Kröse,	Universiteit van Amsterdam
Dr. M. Bicego,	University of Verona, Italy

This work was partly supported by the FET programme within EU FP7, under the SIMBAD project (contract 213250).

Prototype Selection for Classification in Standard and Generalized Dissimilarity Spaces

Thesis

presented for the degree of doctor
at Delft University of Technology
under the authority of the Vice-Chancellor,
prof.ir. K.C.A.M. Luyben,
to be defended in public in the presence of a committee
appointed by the Board for Doctorates
on September 24 2015 at 15.00 hours

by

Yenisel PLASENCIA CALAÑA

Computer Science Licentiate from
Havana University
born in Havana, Cuba

This thesis is approved by the supervisor: Prof.dr.ir. M.J.T. Reinders

Adjunct supervisor: Dr.ir. R.P.W. Duin

Composition of the Doctoral Examination Commitee:

Vice-Chancellor,	chairman
Prof.dr.ir. M.J.T. Reinders,	Delft University of Technology, supervisor
Dr.ir. R.P.W. Duin,	Delft University of Technology, adjunct supervisor
Prof.dr. E.B. García Reyes,	Advanced Technologies Application Center, Cuba
Prof.dr.ir. B.P.F. Lelieveldt,	Leiden University Medical Center
Prof.dr.ir. P.P. Jonker,	Delft University of Technology
Prof.dr.ir. B.J.A. Kröse,	University of Amsterdam
Dr. M. Bicego,	University of Verona, Italy

This work was partly supported by the FET programme within EU FP7, under the SIMBAD project (contract 213250).

Contents

Chapter 1

Introduction

We, as humans, interpret the world around us through the objects and phenomena that we perceive as well as from information sources other than our own perception (e.g. a teacher, Internet). We are able to analyze and organize this information, realizing both explicitly and implicitly that there are different types of relations among entities. Such relations are usually based on commonalities of entities which make them to belong to similar categories. By discarding noise or small differences, defining and analyzing what is relevant and organizing this information, we build up our knowledge of the world.

The gained knowledge is applied afterwards on a regular basis for problem solving. This is true even for the most common and apparently trivial procedures, such as picking up a pen to write something. To perform this task, we need to know what a pen is before we can even find it. Whit this knowledge we can inspect our environment and decide which objects belong to the category pen. If we did not build up this knowledge before, we will not be able to find a pen, even if it is in front of our eyes.

In order to decide if objects belong to the same category (or class), we must first learn the common patterns that characterize the class by individual characteristics or similarities. To achieve this, a set of examples is usually needed. We sense the objects and learn the characteristics that are relevant to the class. Alternatively, one can determine the similarity between objects to find out if they are sufficiently similar to belong to the same class. And probably our brains process other criteria which cannot be easily formalized.

As computers become more and more powerful, there is growing interest in creating automatic methods that are able to do the same: learn patterns relevant to categories and later assign these categories to some unseen entities. The field of pattern recognition [1] is concerned with the question of how to create these automatic methods that learn the relevant knowledge for the characterization of categories from labeled objects, and that based on this, can decide the membership of an unseen object to one of the categories.

1.1 Representations for pattern recognition

A fundamental question for pattern recognition systems is what defines what is "relevant" for class memberships of objects in a particular scenario. Knowledge on this relevancy is generally referred to as prior knowledge. An important consequence of this knowledge is that it guides the way on how we need to represent the objects such that it is best suited to be exploited for use in automated pattern recognition systems. The main approaches for representation in automated pattern recognition include: the feature or vector space representation [1], the structural representation [2], and the (dis)similarity representation [3].

1.1.1 Vector space representation

The vector space representation encodes measurements on objects in vectors of the same fixed length. Based on, for example, statistical techniques, the vector space can be analysed for representation of objects of particular categories [4]. Generally, the analyses in these vector spaces assume metric dissimilarities [3], and most often even Euclidean [3] ones. However, a Euclidean metric, despite being mathematically sound, may not be sufficiently robust or discriminative for many real world problems [5, 6].

1.1.2 Structural representation

Suppose that we want to represent objects by their different parts and the relations among those parts. Further suppose that the number of parts may vary from one object to the other. In such a situation, the vector space representation is clearly unable to represent these objects. For such problems, the structural representation is best suited, which represents objects in the form of graphs or sometimes by strings [7]. A major limitation for the structural representation is that, opposite to vectorial representation, there is a limited amount of analyses techniques which are able to tackle classification (see [2] for an overview).

1.1.3 Dissimilarity representation

In the dissimilarity representation, introduced by Duin and Pekalska [3], pairwise resemblance between objects is measured and used for representation. The approach is in agreement with what humans perceive for creating the knowledge about different classes, so proximity information may be more suitable to define class memberships than features of the individual objects. The resemblance is usually provided by experts in the form of a square proximity matrix where entries are the results of all the pairwise comparisons made for the objects. One interesting aspect of this approach is that proximities can be computed directly on the original objects as well as on top of vectorial or structural representations. Figure 1.1 presents a graphical representation of the three main approaches.

The vector space representation seems to be a less robust modeling since it cannot easily codify objects with variable parts, but much more classification techniques are available. The structural representation is potentially a more robust modeling but less classification techniques are available for this representation. The dissimilarity representation seems to be more in agreement with how humans create knowledge about class differences [3]. Also there are still many classification techniques available for this representation [8]. Within this thesis we therefore focus on the dissimilarity representations.

1.2 Dissimilarity representations

As experts gained more experience on how to represent their data better for automatic analysis, they started to incorporate invariances and expert knowledge in the dissimilarities they compute. Due to this, more complex dissimilarities arose, which may even be non-Euclidean and non-metric. The framework developed for dissimilarity representations addresses this type of proximity measures. Initially the only classifier suited to handle the dissimilarity representation was the k Nearest Neighbour Classifier (k-NN). This classifier however suffer from sensitivity to noise and outliers which is even more harmful for small sample sizes. Besides, they require high storage capacity, and they are in general very slow for classifying a new object. Another limitation is that they do not use the information contained in the dissimilarities to remote

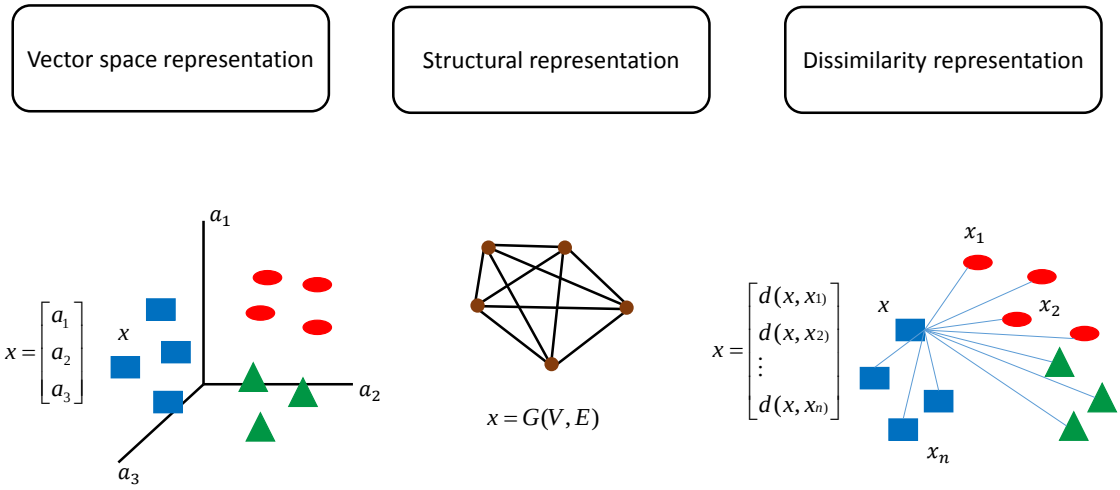


Figure 1.1: Schematic view of the three main representations for pattern recognition

objects which may be helpful.

Dissimilarity spaces (DSs) were proposed to overcome the above limitations of k-NN classifiers [3]. They offer a promising alternative for classification of dissimilarity data since they are able to handle dissimilarities which are non-Euclidean and even non-metric, and they take into account all the information present in a given matrix of dissimilarities between objects.

We assume in this thesis the following definition of a distance function and a dissimilarity function. A metric or distance function on a set X is the function $d : X \times X \to \mathbb{R}^+$ where, $\forall x, y, z \in X$, it satisfies:

$1 : d(x, x) = 0$
$2 : d(x, y) = 0 \leftrightarrow x = y$
$3 : d(x, y) = d(y, x)$
$4 : d(x, z) \leq d(x, y) + d(y, z)$

A dissimilarity function on a set X is the function $d : X \times X \to \mathbb{R}^+$ where, $\forall x, y \in X$, it fulfills:

$1 : d(x, x) = 0$
$2 : d(x, x) \leq d(x, y)$

It can be seen that the definition of dissimilarity function assumed throughout this thesis is very general and has weak assumptions. For pattern recognition we believe that it is more important to create dissimilarity measures containing expert knowledge about the problem which allows learning and class discrimination instead of imposing metric properties that might not contribute to discrimination. Note that a dissimilarity can be transformed into a similarity by straightforward operations, for example, a dissimilarity d can be transformed into a similarity s by applying $s = -d$.

If the final goal after computing dissimilarities is learning from a set of objects, we do further assume that: small values of dissimilarity represent high resemblance between the objects being

compared, and, the more different the objects are, the larger their dissimilarity. This is the monotonicity assumption.

Throughout this thesis we assume that an expert gives us a dissimilarity measure or matrix for a dataset. There are procedures that automatically create dissimilarities [9], by learning a "good" measure to discriminate objects. Albeit these procedures usually learn metric dissimilarities.

When Pekalska et al. [3] proposed the dissimilarity representation approach they claimed that the nearness information is more important for discriminating between the classes than the composition and features of each object independently. In addition, this approach has the potential of unifying the statistical and the structural approach because, for example, dissimilarities can be computed from a structural representation. Next, using the computed dissimilarities, classification can be performed with any of the available classifiers for feature spaces. Therefore, dissimilarity representations bridge the gap between structural and vectorial representations [7, 10]. In [11] we can find theory, methods, experimental results and open questions on the dissimilarity representation.

According to [3], there are three different approaches for classification using a dissimilarity representation:

1. k-NN rule applied to the dissimilarity matrix [12, 13]
2. Classifiers constructed in an embedded space [14, 15]
3. Classifiers constructed in a dissimilarity space [3, 11]

The first approach refers to the well known k-NN classifiers that can be considered as the first dissimilarity-based classifiers. However, they operate on the original dissimilarities directly while the other approaches map the data first to another representation space. In the case of the second approach, the authors proposed to embed the dissimilarities into a Pseudo-Euclidean space aiming to maintain the all vs. all dissimilarities as good as possible. In the third approach, the data is mapped to a space that is built by a set of items called representation set or prototypes set. This dissimilarity space is not aiming at preserving the dissimilarities. Instead, the goal is to exploit the dissimilarities with the set of prototypes to build a good representation for classification. However, there are some studies showing that the DS preserves the partial order of dissimilarities [16], therefore despite the original dissimilarities are not exactly preserved as in the embedded space approach, their order relations are. Figure 1.2 presents these classification approaches.

One advantage of the dissimilarity space over the two others is that one does not need to compute the full dissimilarity matrix: only the dissimilarities with respect to a set of prototypes must be computed. In addition, when using an embedded space, the dimension of the Pseudo-Euclidean space is generally defined by the amount of training data to be used by the classifier which in classification problems without previous knowledge must be large. Besides, the mapping is computationally expensive ($O(n^3)$), and the projection of incoming test data is still an open problem. Contrary, the dimension of the DS is defined by the number of prototypes selected which is usually smaller than the training set cardinality. In addition, there is the advantage of a trade-off between classification accuracy in the DS and computational efficiency. The mapping to the DS is computationally inexpensive except for very expensive dissimilarity measures, and the mapping on unseen data is well defined. Consequently, in this study, we will focus on the third approach which is briefly presented in the next section.

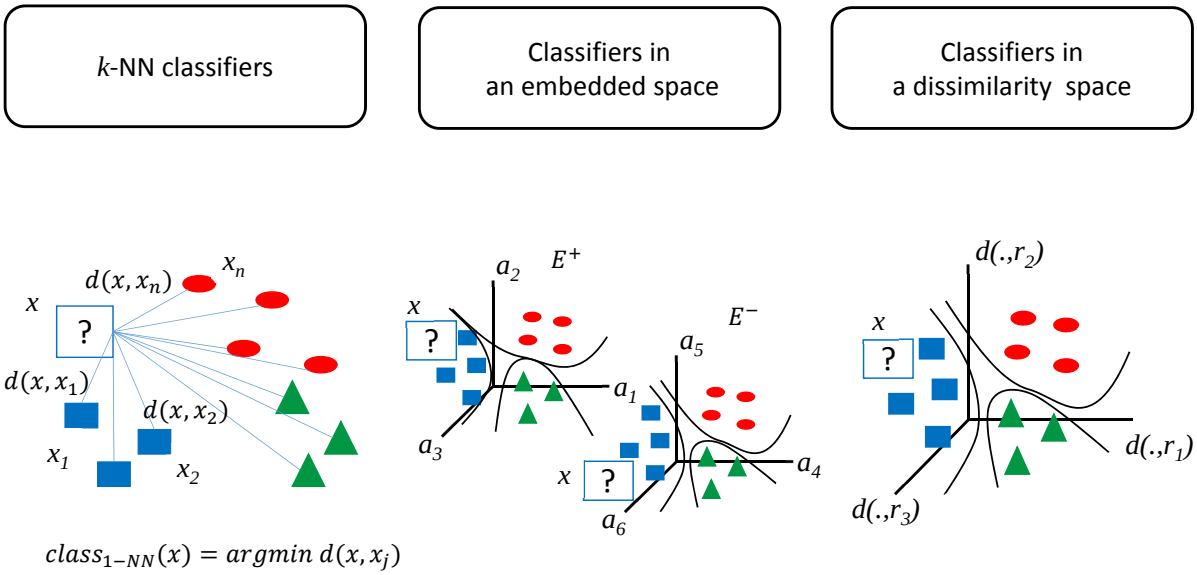


Figure 1.2: Comparison of the three main classification approaches for a dissimilarity representation, from left to right: k-NN classifiers which assigns the class of the object(s) with minimum distance to the input test object x, classifiers in an embedded Pseudo-Euclidean space with the related positive (E^+) and negative (E^-) Euclidean parts (see [14, 15] for more information), and classifiers in a dissimilarity space

1.3 Dissimilarity space and prototype selection

The dissimilarity space was proposed by Pekalska et al. [3]. It was postulated as a Euclidean vector space allowing the use of several statistical classifiers created for such spaces. Let X be a space of objects which might not be vectorial, and let Z be a space of prototypes which may coincide with X or may be composed by models of objects in X. Usually, in practice we only have a finite sample $T = \{x_1, x_2, ..., x_n\}$, such that $T \in X$. To generate the DS we need a representation set $R = \{r_1, r_2, ..., r_k\}$, $R \in Z$, which is a collection of prototypes. Let $d : X \times Z \rightarrow \mathbb{R}^+$ be a suitable dissimilarity measure that allows one to compute some type of resemblance between objects in T and prototypes in R, which extends to a $n \times k$ dissimilarity matrix $D(T, R)$. In case $Z = X$, $d(\cdot, \cdot)$ is provided by an expert to measure pairwise dissimilarities between objects, but if Z contains models derived from the objects in X, $d(\cdot, \cdot)$ is a function of the expert's defined dissimilarities between the object and the other objects that created the prototype model. Usually, the items belonging to R are chosen adequately based on some criterion. Different criteria can be thought of depending on the data distribution and nature of the problem at hand. It is often convenient to select the set R out of the given finite set of objects T as a starting point. Once R is determined, the dissimilarities of objects in T to objects in R are computed for obtaining the representation of T in the DS. The dissimilarity space is created by the data dependent mapping $\phi_R^d : X \rightarrow \mathbb{R}^k$ where an object x is represented

by the vector of dissimilarities between x and R:

$$\phi_R^d(x) = [d(x, r_1)\ d(x, r_2)\ \ldots\ d(x, r_k)]. \tag{1.1}$$

Each coordinate of a mapped object in the DS corresponds to its dissimilarity with some prototype and the dimension of the space is determined by the amount of prototypes used. The question arises how to select the "best" set of prototypes for a given problem.

This thesis is concerned with the selection of the prototypes for the creation of dissimilarity spaces which provide good trade-offs between classification accuracy and computational efficiency. Ideally, we aim at achieving the best possible classification accuracy for the problem in a selected dimensionality which may be defined by the user or by some intrinsic dimension estimation method. Our interest is on small representation sets which avoid the theoretical and practical problems of large dimensionalities [4]. In addition, sometimes a small representation set can even lead to better classification results than a larger one, e.g. when we discard noisy objects for prototypes or because the curse of dimensionality is avoided [17]. The minimization of the number of dissimilarity computations is of great importance, especially for reducing cost of online computations of dissimilarities for incoming test data and for measures that are expensive to compute.

The representation set is composed by items that are used in a combined way to build the

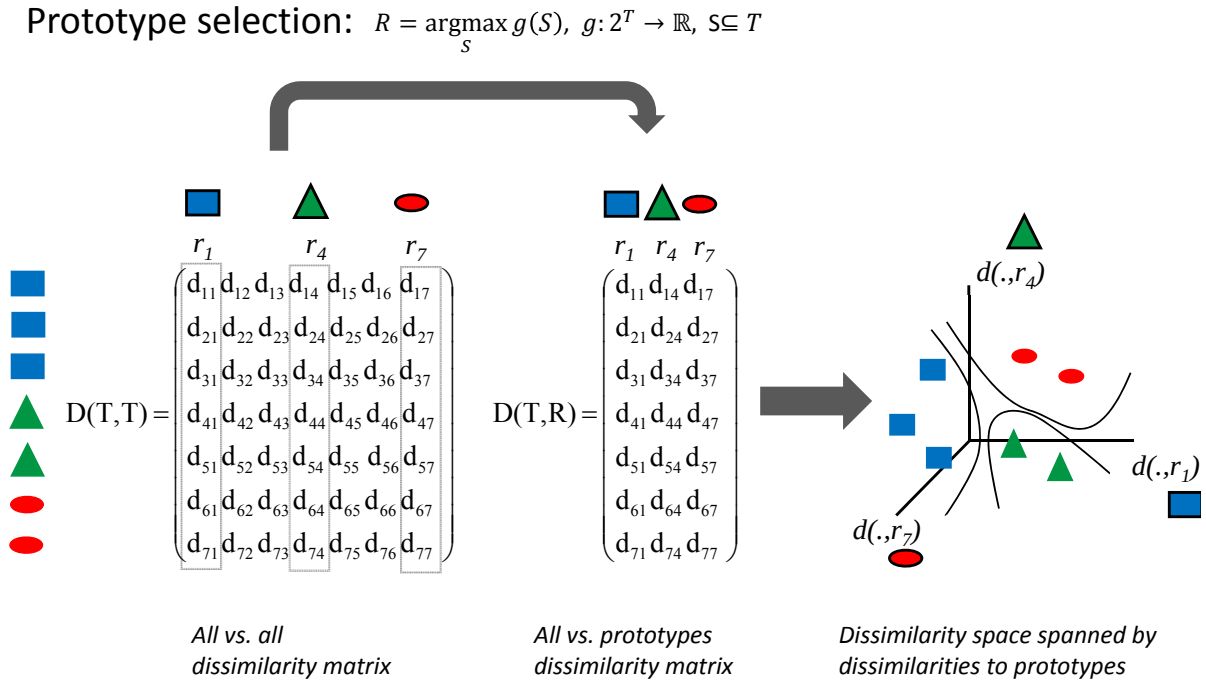


Figure 1.3: Prototype selection considering the set of candidate prototypes as the training set: The selected set of prototypes R is the one that maximizes a function $g : 2^C \to \mathbb{R}$ expressing the representativeness of a set of prototypes.

dissimilarity space. These items are objects or models from the objects present in a dataset. In the DS, dissimilarity vectors computed as shown in equation 1.1 are projected, and this is performed by some specific dissimilarity measure. A more formal definition of a prototype

selection method is as follows. We assume that we have a set $T = \{x_1, x_2, ..., x_n\}$, where x_j may be original objects such as images, raw measurements, vectors, strings, graphs or any other intermediate representation which might not be explicitly given. Instead, we are given the dissimilarities between the objects, i.e. an $n \times n$ dissimilarity matrix $D(T, T)$. The set of candidate prototypes $C = \{c_1, c_2, ..., c_m\}$ is constructed by the following function $c_k = h(\hat{T})$ with $\hat{T} \subseteq T$, where h is a function that provides some type of combination of the objects, e.g. a linear one. Note that this formulation allows the case $c_k = x_j$. The general formulation implies that the function may even return very abstract prototypes, e.g. clusters or any other model as long as we can compute a distance or dissimilarity measure with the models. For some function $g : 2^C \to \mathbb{R}$ expressing the representativeness of a set of prototypes, a prototype selection method finds a subset $R = \arg\max_S g(S)$, where $S \subseteq C$. Therefore, the representation set is a subset of prototypes that represents the dataset well in terms of g, where the interest is not in individual prototypes, but on the representation set as a whole. A graphical representation is shown in Fig. 1.3. Ideally, the more representative the set is, the better classification results in the DS may be obtained. Each selection method has its own definition of representativeness which is given by the selection criterion used (for more information about different criteria studied to find representative prototypes see Chapter 2).

The selection of prototypes seems similar to the selection of features for feature spaces (see Fig. 1.4). However, the interpretation of features is different from the interpretation of prototypes since features might be very different and unique while dissimilarities are homogeneous since they relate the same type of objects. Therefore, adequate methods for selecting features are not necessarily adequate to select prototypes. A set of prototypes as well as vectors of dissimilarities are homogeneous in the sense that they represent values of the same dissimilarity measure. Whereas, a set of features may be very different since features may correspond to different measurements, even not numerical e.g. categorical. Consequently, the comparison of features is ill-defined. In contrast, the comparison of prototypes is well defined and it can be performed in a natural manner by an expert-defined dissimilarity measure or by any distance measure.

Some characteristics are specific for prototypes and cannot be easily established for features. For example close prototypes represent redundant information. Homogeneously distributed prototypes are likely to be good for representing multi-class problems since they should cover all the classes. The most central prototypes in a dataset may not be good for representing multi-class problems since objects from different classes in the same radius from the center of the distribution will have the same representation. Analogies of these observations cannot be found for the case of features.

Despite the fact that feature selection is different from prototype selection, The problems related to high dimensional representation also hold for dissimilarity vectors as they hold for feature vectors:

- High computational costs of classification and storage
- Problems related to the "curse of dimensionality" and small sample sizes
- High-dimensional representations are likely to contain noise since intrinsic dimensionality of the data is usually small, leading to overfitting

These issues encourage the use of a reduced set of prototypes, which in turn promote the study of new prototype selection methods which are able to find the best mapping to a dissimilarity space leading to reduced vectors. A prototype set selection method has two main components: the criterion to be optimized and the search method. The search space for the evaluation function is exponential and finding the optimal solution for the selection problem is intractable. Special interest must be paid to the design of the selection strategy in order to be able to find a

sufficiently good solution without having to analyze all possible solutions. We resort to heuristics for fixing the cardinality of the potential set of prototypes and for exploring solutions in the search space. The criteria to be optimized are either unsupervised or supervised. In the case of an unsupervised criterion we only make use of the dissimilarity information taking advantage of some underlying assumptions related to the objects distribution in the spatial sense. In this category we can find clustering-based methods. In the case of supervised methods, class label information is exploited usually in the form of minimization of a classification error. Other possibilities exist, such as maximizing some interclass distance and minimizing some intraclass distance.

Although we do not propose new approaches in the final classification step, we always use

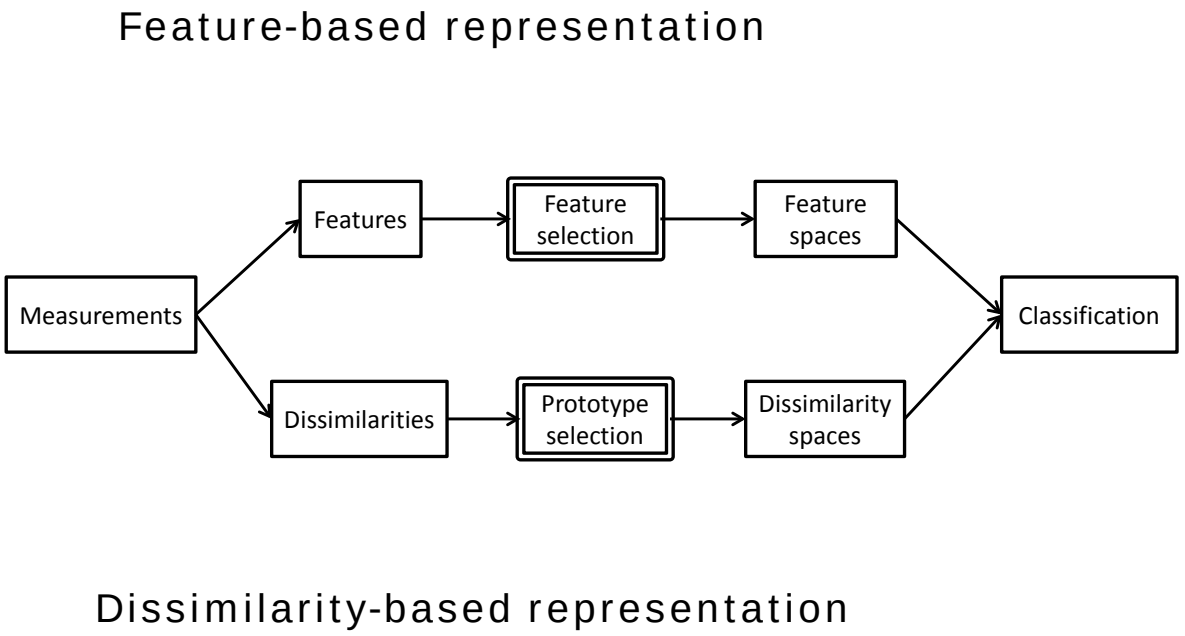


Figure 1.4: Diagram of a classification system for both the feature-based and dissimilarity based representations emphasizing the selection of features or prototypes

the classification results as a reference to validate the methods proposed in this thesis. Thereby, we provide a formal definition of a general classification function for a two-class problem (multi-class problems will be recasted in two-class problems [18]). For a classification problem, there is a labeled training set $T_L = \{(x_1, y_1), ..., (x_n, y_n)\}$ where $x \in X$ represents the data and $y \in Y$ the corresponding class labels. A classification function is the function $f : X \rightarrow Y$ that for new or unseen object $\hat{x} \in X$, $f(\hat{x})$ assigns a class label $\hat{y}$ to the object.

Classifiers operating in the DS suffer from similar issues as the standard classifiers operating in a feature space: there might be a poor generalization especially if the dissimilarity measure is not discriminative enough. Note that by selecting prototypes we do avoid problems with the curse of dimensionality and small sample sizes. Further, by taking into account dissimilarities with all the prototypes we compensate for locally-sensitive measures and obtain an improved

representation. Moreover, the DS space provides some kind of nonlinear mapping that makes the data better linearly separable. Consequently, the linear and quadratic classifiers have shown good performance in previous studies of DS classification [19]. These classifiers, together with the support vector machine and the 1-NN, are used throughout this thesis to show the effectiveness of the proposed prototype selection methods.

All our experimentation protocols follow a similar methodology. From a given dataset, we usually randomly partition it into training, validation and test set for a number of times and the final reported results are the average over the results on the random partitions. The training set is mapped into the DS and used for training the classifiers. The validation set is used for selecting the prototypes and for computing the selection criteria, while the test set is used for computing the classification errors. When our dataset is very small, we use the same set for training and validation. Note that our validation set has a different purpose from validation sets used to optimize classifiers. In our case, the validation set is the set of candidates to select the representation set.

1.4 Outline of the thesis

The content of this thesis is divided into five main topics: **Chapter 2** reviews the "Related work" to show the reader the previous efforts devoted to select prototypes for classification in the DS. **Chapter 3** is concerned with "Prototype selection by genetic algorithms". **Chapter 4** studies the topic "Creation and selection of prototype models", while **Chapter 5** deals with the "Selection of prototypes in extended dissimilarity spaces". **Chapter 6** provides some concluding remarks and open issues.

In Chapter 2 we summarize and discuss some of the main strategies that had been proposed for the selection of a representation set composed by objects or models in order to generate dissimilarity spaces for general domains as well as for specific ones such as graph and string domains. An analysis is performed on what was not yet addressed by these procedures, pointing towards directions of research which guided the work developed in the subsequent chapters. The best procedures in terms of accuracy, efficiency and ability to cope with non-metric dissimilarity measures are highlighted.

In **Chapter 3**, we study the suitability of genetic algorithms (GAs) for selecting prototypes by supervised and unsupervised criteria. It is shown that GAs are able to cope with local optima better than other suboptimal procedures such as the forward selection for the problem of prototype selection. We pay special interest to the problem of selecting prototypes out of very large datasets. We propose two new methods for scalable prototype selection by using fast and scalable unsupervised or supervised criteria, and exploiting the suitability of genetic algorithms to find a good compromise between accuracy of the solution and speed of convergence. In addition, new versions of the methods are proposed and analyzed for the case that the datasets do not fit into memory and the dissimilarities required by the selectors are computed on demand avoiding the computation of the full dissimilarity matrix. Besides, we propose a methodology for adapting the principal component analysis for the linear intrinsic dimension estimation for moderate and large datasets. Parts of this chapter were published in [20, 21].

Chapter 4 is devoted to study how to create and select a representation set composed by models to generate generalized dissimilarity spaces (GDS). By this, we refer to a DS generated by models as prototypes. We first study the selection of linear models of the objects called feature lines and propose a new method for the selection of feature lines that copes with interpolation and extrapolation problems likely to occur in this representation. Besides, the method showed to be more generally applicable to different datasets than the previously proposed methods based on the length of line segments. In the second part of this chapter, we propose the use of new models based on clusters as prototypes and we study what statistics are more proper

for measuring the overall distance between objects and the clusters. The minimum, maximum, and average statistics are considered. We also propose a new method based on the *Nyström* approximation to compute the subspace distance from objects to the subspace spanned by the objects inside the prototype cluster. The suitability of the four proposals is studied for different data distributions. In addition, standard supervised selection methods that optimize classification in the DS are used for the selection of the best representation set based on clusters. The two main parts of the chapter were published in [22, 23].

Asymmetric dissimilarity measures have the peculiarity of returning two different values

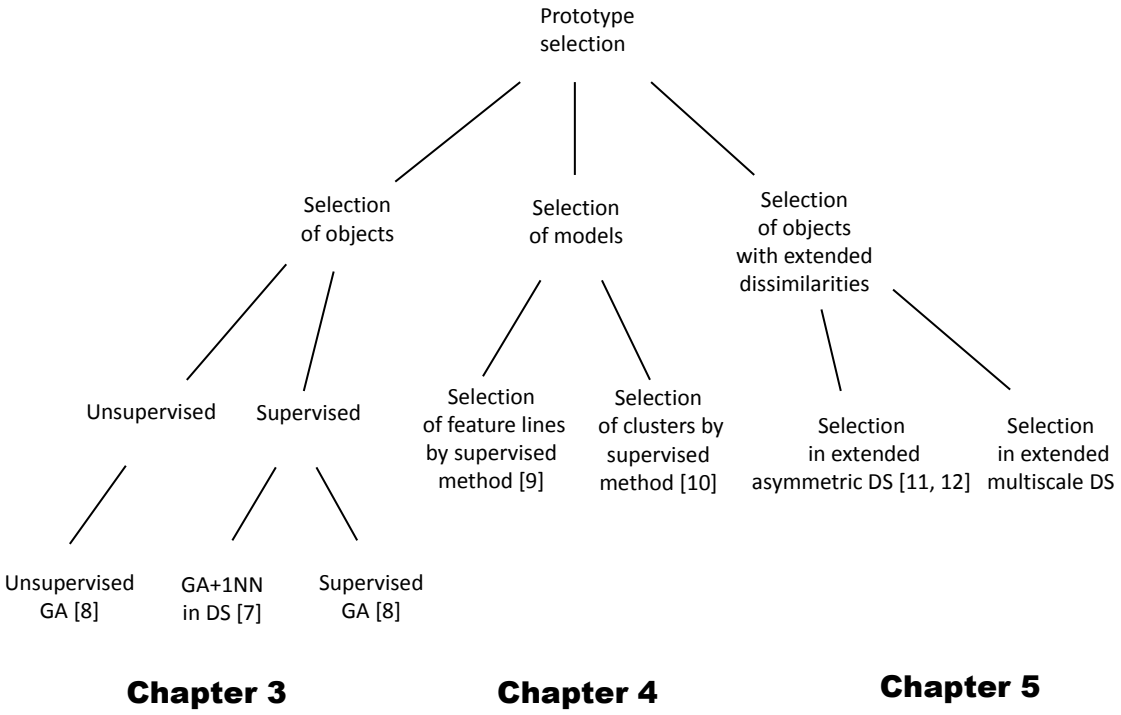


Figure 1.5: General taxonomy of prototype selection methods including the methods proposed in this thesis at the bottom of the hierarchy with the corresponding chapters where they appear

if the computation is performed in the two possible directions, from the objects to the prototypes and from the prototypes to the objects. In **Chapter 5**, we consider that we may loose important information if we impose symmetrization before submitting the data to automatic classification. We study how to actually make use of the two directed asymmetric dissimilarities which lead us to propose the extended asymmetric dissimilarity space as a mean to fully use them without imposing symmetrization. By extended space, we refer to a space that is a cartesian product of different spaces, which in this case coincide with two DS, one for each directed dissimilarity. This is a type of generalized DS, since the prototypes are extended models of the objects by considering multiple dissimilarities. We also study the selection of prototypes in this proposed space. Instead of prototype selection with a symmetrized dissimilarity, in this approach, the prototypes together with the best directions to compute the dissimilarities are selected to conform the final DS. The obtained results confirm that the approach is in many cases superior to any of the symmetrization procedures.

The second topic of **Chapter 5** is the selection of prototypes for multiscale dissimilarity data. When it is possible for some problem to obtain multiscale dissimilarities that may be non-metric the question arises how to select the prototypes from an extended multiscale dissimilarity space in a way that the best information provided by the prototypes in the different scales is preserved. We assume that prototypes which are good in one scale are not necessarily good in other scales, thereby we select the best prototypes with their best related scales. In the final part of the chapter we study the suitability of standard supervised methods (since these methods inherently use our assumption) to select prototypes in the extended multiscale dissimilarity space compared to other methods for combining non-metric dissimilarities for the creation of the DS. The proposal is also compared to the selection of prototypes including all the dissimilarities in all the scales and with classification in individual scales. Some parts of this chapter were published in [24, 25].

General concluding remarks and open issues that may be addressed in future works are presented in **Chapter 6**.
For a better understanding, Fig. 1.5 shows the relations among the proposed methods in a general taxonomy with the corresponding chapters where they appear.

1.5 Main contributions

This thesis is concerned with the selection of prototypes for the classification of data in the dissimilarity space. Our main research question is: Can we create better prototypes and/or selection procedures if we take the nature and characteristics of the dissimilarity data into account in the process? Our general hypothesis is that if we take into account the nature and spatial distribution of the dissimilarity data we can obtain better selection procedures, better prototypes and better DSs in the sense of compromising between classification accuracy and efficiency. From our study we confirmed this hypothesis, especially for the selection of small numbers of prototypes. By making use of the dissimilarity nature of the data, better and faster procedures are obtained. In addition, by creating suitable models, which reflect the spatial distribution of objects, the methods outperform previous general approaches based on objects as prototypes. We found that taking into account the nature of asymmetric and multiscale dissimilarities in the definition of the DS is more successful than ignoring their specificities.

Chapter 2

Related work

Abstract
A common way to represent patterns for recognition systems is by feature vectors lying in some space. If this representation is based only on the predefined object features, it is independent of the other objects. In contrast, a dissimilarity representation of objects takes into account the relations between them by some measure of resemblance (e.g. dissimilarity). The nearest neighbour (1-NN) is a dissimilarity-based classifier that has shown to be very competitive for several pattern recognition problems. Classification results on dissimilarity spaces spanned by dissimilarities to prototypes can reach or improve the 1-NN results in terms of accuracy and computational efficiency. This is possible if a small set of prototypes is selected with similar discriminative power than the complete set of initial prototypes. How to obtain this set has been studied by researchers in the area of dissimilarity representations and graph representations by means of prototype selection methods. In this chapter we present an overview and a discussion of different approaches proposed in the literature on this topic.

2.1 Introduction

Different prototype selection methods have been developed with the aim to find a small representation set that is still capable of generating a dissimilarity space where classifiers can discriminate between the classes as well as or even better than with all the initial objects, since by improving the radio #trainingobjects/dimensionality we avoid overfitting and the curse of dimensionality. Moreover, we gain in computational complexity, both, for representation as well as for classification. For dissimilarity representations the adaptation of prototype selection techniques available for the vector space representation or feature-based approach has been investigated showing good results [17]. Also new techniques have been investigated [11].

In general in the k-NN literature two basic types of algorithms can be identified: prototype generation and prototype selection [3, 11]. The first group focuses on merging the initial prototypes in a way that optimizes the performance of k-NN. Examples of these algorithms are Kmeans [1] and learning vector quantization (LVQ) [26]. The second group focuses on reducing the original set. Condensing methods identify a small set so that the overall performance in this set is similar to the performance in the original set. Editing methods remove noisy samples leaving smooth decision boundaries [27, 28]. Generally, condensing methods are applied after the editing methods. The editing and condensing methods have the disadvantage of usually working in Euclidean spaces.

The main difference of the application of prototype selection methods for k-NN and for dissimilarity space classifiers is that in the first case, the techniques are applied for choosing the final training objects; and in the second case for determining the prototypes to construct the dissimilarity space, since still all the initial objects will be used for training. In [17] the authors compared prototype selection methods for constructing dissimilarity spaces showing good results when used with linear and quadratic classifiers. In [11] various techniques were compared such as Kcentres, Modeseek, feature selection, linear programming, editing-condensing methods, and a mixture of Kcentres with linear programming. These techniques showed good performance especially for small sets of prototypes, where random selection performed worse. Other prototype selection methods have been proposed in the graph and string domain [29,30]. The methods tackle the question of how to select a small representation set for constructing the dissimilarity space.

It is also of our interest to detect the methods that can be applicable to generalized dissimilarity representations, where instead of selecting objects for prototypes, the methods will select models of the objects. We can find studies on generalized dissimilarity representations using feature lines and feature planes in [31].

We briefly summarizes some of the methods presented in the literature on prototype selection, which are applicable to dissimilarity data computed from the initial objects directly, from vectorial representations and from graphs or strings, where prototypes can be objects or models. Also, methods that create or generate new prototypes instead of selecting them from a set of objects or models that already exist have been investigated, but these methods are out of the scope of this thesis since they rely in an underlying feature space. In our case, we only assume that the dissimilarity data is given in the form of a dissimilarity matrix. Examples of those methods are: means of clusters [32], LVQ and mixture of gaussians [17]. The following section presents an overview of the procedures used in the literature [11, 17, 29–31].

2.2 Procedures

Random. This method selects k prototypes randomly from the training set [11]. This is a type of sampling which makes sense for dissimilarity data since it takes into account statistical distributions, e.g. uniform distribution. It can work well for large prototype sets since neighbouring objects present similar representational capabilities (they are similar prototypes candidates). However, the random selection may be less successful if a small prototype set is needed. Besides, as a disadvantage, it is possible to find redundant prototypes with this method.
KCentres. This unsupervised method is based on a clustering procedure for dissimilarity data [11]. It selects k prototypes in a way that they are evenly distributed with respect to the dissimilarity information. The procedure randomly selects a first set of prototypes and assigns the other objects in the dataset to their nearest prototype to create the clusters. By an expectation maximization (EM) type of algorithm, the initial prototype for each cluster is replaced by the center of the cluster, i.e., the prototype that minimizes the maximum distance to the objects in its cluster. For a training set $T = \{x_1, x_2, ..., x_n\}$ divided into k disjoint subsets P_j or clusters containing $n_j, j = 1...k$ objects each, the next criterion is minimized for each cluster centre r_j:

$$\jmath = \max_{x_i \in P_j} d(x_i, r_j). \tag{2.1}$$

As another more robust option, the average distance to all the objects in the cluster can be minimized as well:

$$\jmath = \sum_{x_i \in P_j} d(x_i, r_j). \tag{2.2}$$

The cluster center is selected as prototype after a predefined number of iterations. One disadvantage is that the result is sensitive to the initialization. Another disadvantage is that densely populated regions are more represented than sparsely populated ones.
Modeseek. The method is also based on a clustering procedure [11] for feature spaces. It finds the modes of the density estimate using a nearest neighbour technique. For each object x_j, the method finds the dissimilarity to its sth neighbour. The selected prototypes are the ones with minimum dissimilarity to their sth neighbour:

$$\min_{x_j \in T} : \jmath = d(x_{n_j}, x_j), \tag{2.3}$$

where x_{n_j} is the n-th neighbour of x_j. For given dissimilarity matrices it is a fast procedure.
FeatSel. It is a supervised greedy forward selection [33] optimized for dissimilarity data [11]. It uses as criterion the Leave-One-Out (LOO) 1-NN error based on the selected prototypes. It can be understood as a forward prototype selection for editing. This proposed criterion in [11] is very fast since it operates directly in the dissimilarities between training and representation

objects:

$$\min : \jmath = \sum_{x_i \in T-R} CE(x_i),$$

$$CE(x_i) = \begin{cases} 1, & \lambda_T(x_i) \neq \lambda_R(r_k) \\ 0, & \lambda_T(x_i) = \lambda_R(r_k) \end{cases}, r_k = \underset{r_j \in R}{\operatorname{argmin}}\, d(x_i, r_j) \tag{2.4}$$

in which $\lambda_T(x)$ is the class label of $x \in T$, $\lambda_R(r)$ is the class label of $r \in R$ and $\jmath$ is the 1-NN classification error of the training set T classified by the set of prototypes R. If $R \subset T$, representation objects are excluded in T as a possible nearest neighbor of x if $x = r$ (LOO approach). The prototypes are considered in a DS, but the classification error for each prototype set is computed based on the LOO 1-NN error on the original dissimilarity matrix and not in the DS. In cases where the classification error is the same for different representation sets, ties are solved by selecting R for which the sum of dissimilarities between T and R is minimum.

LinProg. This method [11, 34] solves a linear optimization problem in order to train a sparse separating hyperplane $w^T D(x, R) + w_0$ in a dissimilarity space $D(T, R)$. The w_j are expressed by non negative variables α_j and β_j as $w_j = \alpha_j - \beta_j$. In the optimization problem it is also introduced a nonnegative slack variable ξ_i that accounts for classification errors as well as a regularization parameter γ. Let $x_i \in T$ be training objects with class labels $y_i \in 1, -1$, the minimization problem is formulated as follows:

$$\min : \quad \jmath = \sum_{i=1}^{n} (\alpha_i + \beta_i) + \gamma \sum_{i=1}^{n} \xi_i \tag{2.5}$$

$$\text{subject to} \qquad y_i f(D(x_i, R)) \geq 1 - \xi_i, i = 1, ..., n \tag{2.6}$$

$$\alpha_i, \beta_i, \xi_i \geq 0. \tag{2.7}$$

The final prototypes are the objects that have w_j weights different from 0. The authors state that this procedure may be beneficial for two-class problems from a computational point of view, but for multiclass problems not so much since they may results in a large set of prototypes.

EdiCon. This is the classical Editing Condensing algorithm [27] applied to the original dissimilarity matrix [11] and not to a dissimilarity space. The Editing method removes those objects that are erroneously classified by the 1-NN, so the overlapping of classes is decreased. Then, Condensing is applied. Condensing removes objects taking care that the performance of the 1-NN classifier on the new set is similar to the performance using all the training objects. As it can be seen, the algorithm returns at least one prototype per class, since the 1-NN needs to have some pattern of each class to measure the resemblance of a new incoming test object.

Center prototype selector. This procedure was proposed in [30] for representing strings by edit distances and in [29] for representing graphs as it is also the case of the Border and Spanning prototype selectors that will be explained below. The center prototype selector starts from the set median string, and gradually adds the remaining median objects. All the selected prototypes are in the center of the dataset, therefore they are redundant on their contribution for representing the other objects. On the other hand, outliers are avoided, which is a desirable property in order to discard noise. Starting from the empty set $R = \{\}$, in the j-*th* iteration, the method adds the prototype that fulfills the next condition:

$$\min_{x_j \in T-R} : \jmath = \sum_{\substack{i=1 \\ x_i \in T-R}}^{n} d(x_i, x_j). \tag{2.8}$$

It has the disadvantage that, as it selects the objects in the center of the training set, it does not represent well the distribution of training objects.

Border prototype selector. As its name suggests, this method is based on selecting those

objects belonging to the border of the data distribution [30]. It is important to emphasize that whether one object belongs to the center or border of the dissimilarity data is determined by the dissimilarity measure used and the problem at hand. Especially for dissimilarities computed on top of complicated structures such as graphs, a center or border object may be tricky to define. In this method, the problem of the similar contribution of the center prototypes selected by the previous method is avoided. On the other hand, outliers are likely to be selected, since they usually are in the borders of the data distribution.

Spanning prototype selector. Starting from the set median string, the next prototype to be added to the representation set R is the object with largest minimum distance to the current set of prototypes [30], formally, the one fulfilling:

$$\underset{x_i \in T-R}{\operatorname{argmax}} \{ \min_{r_j \in R} d(x_i, r_j) \}. \tag{2.9}$$

In this way, objects that yield similar contribution to the representation set are not likely to be selected, since the next prototype is always the farthest one to the already selected set. Also, the prototypes have a tendency to be uniformly distributed. The authors also point that outliers are likely to be selected since they have a large distance to other objects. Actually, this method is a variant of the farthest first transversal (FFT) [16] which was originally proposed for a K-centers clustering initialization [35]. The difference between the two is that the FFT starts from a random object and not from the median object.

Length-based selection of feature lines. Feature lines are linear models of the objects constructed in a class wise manner [31, 36]. The method is based on sorting all the lines by length. Then it is possible to start the selection by the one that has the smallest length, and gradually add the next always by smallest size. Another possibility is to select the set of lines that has the largest length. Selecting the middle length feature lines was also considered in order to describe slightly curved manifolds.

MaxNCN. This is a condensing method based on the concept of Nearest Centroid Neighbours (NCN) [17]. For an object p, the nearest objects are the ones that are closer to the object. In addition, for the NCN, it is also considered the symmetrical distribution of the nearest objects around p. The idea is based on the assumption that the geometry of the distribution of objects may be more informative than the distances between them. The method proceeds as follows. The first NCN is the nearest neighbour of p. The next NCN to be added is the one that with the previously added NCN, determines the closest centroid to p. Objects found like this, tend to surround p. The process continues while the class of the next maxNCN is the same as the class of p. As prototypes of one class should be located in a neighbouring area, they can be replaced by a single prototype without a major loss in their representation potential. The first prototype for a class is the object with a larger number of NCN. The algorithm removes the NCNs of this prototype and the prototype itself, and updates the number of neighbours of the remaining class objects as being part of an already processed group or neighbourhood. Then, again the object with larger number of NCN is selected, until it is not possible anymore to select a new prototype.

Reconsistent. The MaxNCN procedure has the disadvantage of discarding objects that are close to the decision boundary. The Reconsistent method tries to tackle this problem. After the MaxNCN procedure is applied, all the objects from the training set are classified by the 1-NN with respect to the prototype set obtained by the MaxNCN. The set of misclassified objects is condensed taking as reference the prototype set. Then, the condensed set is added to the prototype set returned by MaxNCN in order to conform the final representation set. In spite of the fact that the method does not take class separability into account by using the minimization of a classification error, somewhat when it stops once one object of a different class is reached, this can be seen a a way of considering the class separability.

2.3 Comparison

Finally, we will compare some aspects that characterize a prototype selection method:

1. The applicability of the method to generalized dissimilarity representations.
2. If the method requires as parameter the number of prototypes to return (controllable by the user) or if it is able to find the appropriate number automatically.
3. Dependance of the selected representation set on the initialization of the method.
4. If the method interprets the dissimilarities as such and not as arbitrary numbers, e.g. the fact that small numbers indicate high resemblance and large numbers indicate small resemblance is taken into account.
5. If the class separability is optimized.
6. Ability to work with non-metric data, that is if the dissimilarity measure is not restricted to be metric.
7. Ability to handle asymmetric data, that is if the dissimilarity measure is not restricted to be symmetric.

Table 2.1: Comparison of the prototype selection methods from the literature

methods	Applicable to GDR	#prot set by user	Initialization dependant	Interprets diss.	Supervised	Non-metric diss.	Asym. diss.
Random	yes	yes	yes	no	no	yes	yes
Kcentres	no	yes	yes	yes	no	yes	no
Modeseek	no	no	no	yes	no	yes	no
Featsel	yes	yes	no	yes	yes	yes	no
Linprog	yes	no	no	yes	yes	yes	no
Edicon	no	no	no	yes	yes	yes	no
Center	no	yes	no	yes	no	yes	no
Border	no	yes	no	yes	no	yes	no
Spanning	no	yes	no	yes	no	yes	no
Line Length	yes[1]	yes	no	yes	no	no	no
MaxNCN	no	yes	no	yes	no	no	no
Reconsistent	no	yes	no	yes	no	no	no

We can see from this table that only a small number of procedures are applicable to generalized dissimilarity representations, therefore the creation of new methods for this type of representation is a topic that requires future work. In many of the procedures the user has control over the number of prototypes. This may be good for real world applications under time budgets, and when we already know the intrinsic dimensionality of the data. The majority of the procedures are able to deal with non-metric (but symmetric) data. However, only the random selection, due to its simplicity, is able to cope with asymmetric data. This topic requires further study.

2.4 Concluding remarks

In this section we described the prototype selection methods for dissimilarity space classification that existed before this thesis was started. Some suggestions can be made to potential users.

For example, if the expert comes with a classification problem in terms of dissimilarity data, and he knows in advance that the dissimilarity matrix does not fulfill the metric requirements, then the MaxNCN cannot be applied. This method and the Reconsistent were created for data with an underlying metric vector space, so they are not flexible enough to allow the use of non-metric data. The fact that most of the other methods can be applicable to non-metric data, allows the use of dissimilarity data created by experts exploiting knowledge on their specific domains; we claim that this is usually more robust and informative than using one general metric such as the Euclidean.

Prototype selection methods that allow the user to control the number of prototypes to obtain are more adequate for developing systems with execution time constrains; almost all the presented methods fall into this category. Only the EdiCon and Modeseek methods return automatically the best number of prototypes.

For the case when the methods can be executed in a class-wise way, one advantage is that uniformity of the prototypes distribution can be gained when the classes are evenly spread and balanced. If classes are unbalanced, finding a fixed number of prototypes per class will not represent the true distribution. Instead, the number of prototypes should be proportional to the number of elements of each class. Also, exploiting class labels might be beneficial since more information of the problem is being used. One interesting question is whether it is really necessary to have prototypes from all the classes.

Generalized dissimilarity representations allow more flexibility than dissimilarity representations based on just object distances. The possible applicability of the already published methods for dissimilarity representations to generalized ones is a promising research direction.

When class separability can be taken into account in the selection criteria as in the FeatSel case, it may help to obtain good classification performances with a small representation set. This was showed in [11], where the FeatSel method outperformed the others in some datasets.

Random, RandomC and Kcentres have the disadvantage of being dependent on the initialization. Kcentres and ModeSeek are good candidates for unsupervised prototype selection, but they cannot be easily applicable to generalized dissimilarity representations. FeatSel and EdiCon are the only methods that optimize class separability on the training set and due to this they are applicable for the selection of generalized prototypes since a classification error can always be computed for any set of prototypes. The Center Prototype Selector returns prototypes with redundant contribution and the Border and Spanning Prototype Selector have the disadvantage of potentially returning outliers. MaxNCN and Reconsistent need an underlying feature representation available in order to compute the centroids, also they do not allow the use of non-metric dissimilarities.

Results in [11] demonstrate that, generally, the 1-NN classification performances can be reached or outperformed by using prototype selection methods with a small representation set, implying a lower computational cost. There is no superior prototype selection method, it seems that this depends on the problem at hand, dissimilarity measure used, and data distribution. Also, systematic procedures outperformed random ones for small sets of prototypes while for larger sets the random selection performs well.

In [11] the authors claim that prototype selection for multi-class problems pose a superior challenge than two-class ones and require further research.

In general, the authors detected that the best performing procedures from [11] were the Kcentres and the supervised procedures Edicon, Linprog and Featsel. However, the Linprog and Edicon find the number of prototypes automatically, which can be very large for the Linprog case. Therefore we consider only the Kcentres and Featsel since we want to keep control over the number of prototypes for comparison purposes. In addition, from the approaches in [29,30], the best performing procedures were the Spanning and K-Medians (as they named a Kcentres) prototype selectors. Since all these procedures are able to deal with non-metric data, they will

be used for comparison purposes throughout this thesis when applicable for the problems under consideration.

Chapter 3

Prototype selection by genetic algorithms

3.1 Prototype selection for dissimilarity representation by a genetic algorithm

This section has been published as "Prototype Selection for Dissimilarity Representation by a Genetic Algorithm", by Yenisel Plasencia-Calaña, Edel Garcia-Reyes, Mauricio Orozco-Alzate, Robert P. W.Duin, in *Proceedings of the IEEE 20th International Conference on Pattern Recognition, 2010.*

Abstract

Dissimilarities can be a powerful way to represent objects like strings, graphs and images for which it is difficult to find good features. The resulting dissimilarity space may be used to train any classifier appropriate for feature spaces. There is, however, a strong need for dimension reduction. Straightforward procedures for prototype selection as well as feature selection have been used for this in the past. Complicated sets of objects may need more advanced procedures to overcome local minima. In this paper it is shown that genetic algorithms, used previously for feature selection, may be used for building good dissimilarity spaces as well, especially when small sets of prototypes are needed for computational reasons.

3.1.1 Introduction

Data preparation and representation are usually required steps before an automatic analysis on the data is performed. From the pattern recognition point of view it is possible to have supervised (generalization) as well as unsupervised analyses on the basis of some representation of the patterns. Widely extended is the use of vector spaces to represent patterns, where statistical classifiers can be used and are theoretically justified by the metric or Euclidean properties assumed beforehand. More complex representation have been also studied, e.g. structural representations [37], where patterns are encoded in graphs, strings or grammars. Another way to encode the patterns is by dissimilarities derived from pairwise comparisons between objects. It is possible to compute dissimilarities between vectors or structures [29, 30], and also from original data without intermediate representations (e.g. dissimilarities derived from a matching process of images directly).

If we are dealing with dissimilarities instead of distances, where an exact metric embedding is not possible, the alternatives for the classification of this data are [3]: classification by the k Nearest Neighbour (k-NN) rule, classification in dissimilarity spaces and classification after embedding the data in pseudo-Euclidean spaces. The computation of dissimilarities can be computationally demanding as in case of comparisons of images and graphs, then a reduction of the number of dissimilarities to be measured is of interest. One way to do this is by prototype selection. In dissimilarity spaces prototype selection aims to find small sets of objects in order to decrease the dimension of the dissimilarity vectors but for the generalization step all the training objects are included. In the case of the k-NN rule and the pseudo-Euclidean spaces the prototype selection will determine the number of training samples.

For dissimilarity representations some prototype selection techniques have been investigated showing good results [11], specially when used with linear and quadratic classifiers. In [11] various techniques were compared such as Kcentres, mode seeking, forward feature selection, linear programming, editing-condensing, and a mixture of Kcentres with linear programming. It showed that forward selection is one of the best, especially for small sets of prototypes. Other prototype selection methods have been proposed in the graph and string domain [29, 30]. The methods tackle the question of how to select a small representation set for constructing the dissimilarity space. In this paper it is of our interest to study a selective scheme and not a creative scheme since one goal is to work with a given dissimilarity matrix that could have been computed directly from the initial data, and an intermediate space for creating artificial prototypes may not exist.

Genetic algorithms (GAs) have been used for feature selection [38] as well as for prototype selection [39]. In these cases, each specimen is represented by a binary chromosome whose genes are associated either to features or prototypes. However, in almost all studies where GAs have been used for prototype selection, the aim is reducing computational demands of the nearest neighbor classifier as conceived for feature spaces but not for choosing appropriate prototypes in order to build a dissimilarity space.

Fitness functions employed in the above-mentioned studies made possible to emphasize either on the maximization of the classification accuracy or either on penalizing the cardinality of the representation set. Kuncheva and Jain [40] compare the GA-based selection against the sequential forward feature selection (SFS) method as well as against two classic rules of condensing [13]. In general, these studies conclude that the use of GAs for feature/prototype selection is suggested in spite of its non-optimality; nonetheless, others such as [33] point out that simpler feature selection methods should be preferred because GA-based results might not be always so good as expected.

In this paper, we will show that GA-based selection can overcome local minima thus outperform the sequential forward selection of prototypes for dissimilarity space classification. Both procedures are compared on the basis of the same selection criterion, the optimization of class separability by the minimization of the leave one out (LOO) 1-NN error in the training set.

3.1.2 Prototype selection by a genetic algorithm for dissimilarity spaces

The dissimilarity space was proposed by Pekalska et al. [3]. It was postulated as a Euclidean vector space. For its construction a representation set $R = \{r_1, r_2, ..., r_k\}$ is needed, where the objects belonging to this set are chosen adequately based on some criterion. Let X be the training set, R may or may not overlap with X. Once we have R, the dissimilarities of the objects in X to the objects in R are computed. Any object is now represented by a vector of dissimilarities d_x to the objects in R as shown in (3.1).

$$d_x = [d(x, r_1)\ d(x, r_2)\ ...\ d(x, r_k)]. \tag{3.1}$$

The dissimilarity space is defined by the set R. Each set of coordinates of a point in that space corresponds to the dissimilarities to the prototypes and the dimension of the space is determined by the number of prototypes selected. A good selection of R allows one to decrease the computational complexity at the cost of slightly increased error rates. Finding good criteria and algorithms for prototype selection is still an open issue. One of the best methods in [11], forward selection, used the minimization of the LOO 1-NN error as criterion. The computation of this criterion is fast as it relies on the given distances. No other computation than minimum operations are needed.

GAs are inspired in natural systems where species evolve by selection, reproduction, and mutation, ensuring better organisms as well as diversity. They are classified as global search heuristics. One drawback of the methods is that some parameters should be set by the user. Another drawback is that solutions are sensitive to initialization, and a satisfactory solution may or may not have been reached when the method stops.

In the GA vocabulary, a gene is a part of a candidate solution, a chromosome is a string of genes that represents a candidate solution, and a population is the available set of chromosomes to be explored. An important component of a GA is the fitness function that evaluates the candidate chromosomes, the GA minimizes or maximizes the value returned by this function and the best candidate is selected in each evolution cycle according to its fitness. The simplest way to encode the candidate solutions is by strings over the alphabet $\sum = \{0, 1\}$. For example a set of strings of size two over this alphabet is $\sum^2 = \{00, 01, 10, 11\}$.

The GA for prototype selection has as input parameters the dissimilarity dataset and the desired number of prototypes k. Solutions are encoded by binary strings. The first chromosome is chosen as the best, and we iterate in an evolution cycle while the fitness value of the new best chromosome improves in a new iteration. The solutions or chromosomes will evolve by the reproduction, mutation and selection operations.

The heuristic search strategy of the GA tries to avoid the local minima by introducing the random mutations. The forward search from the R initial candidates for prototypes, evaluates first each individual prototype and the one that leads to the best classification performance is

kept, then the second prototype that combined with the first leads to the best classification performance is added and so on, this way of finding the solutions can cause the convergence to local minima.

3.1.3 Experiments

In order to evaluate our hypothesis on the benefits of the GA search strategy over the forward search for the problem of prototype selection for dissimilarity space classification we conducted some experiments on different dissimilarity datasets. The methods are compared in terms of classification errors of the Linear Discriminant Analysis (LDA) in the dissimilarity space while varying the number of prototypes. We compared the GA and the forward selection of prototypes using as selection criterion the LOO 1-NN error. The parameters for the GA were: 50 chromosomes for the initial population, 0.5 of mutation and reproduction probability, and 3 generations without change as stopping criterion. As a baseline we also computed the results for the random selection and for the 1-NN classifier applied directly on the total dissimilarity matrix and not in the dissimilarity space. Learning curves are used to illustrate the changing rate of learning for the LDA classifier while varying the number of prototypes.

Four different dissimilarity datasets were used for the experiments, the CatCortex, Chickenpieces-20-60, CoilYork and CoilDelftDiff datasets [41]. All of them are multiclass problems. The datasets were split randomly into training and test sets taking approximately 50 percent of the objects in each set. A comparison of different properties of the datasets can be found in Table 5.3.

The smallest dataset is the CatCortex [42], the 65x65 dissimilarity matrix is measured in an ordinal scale and describes the connection strengths between 65 cortical areas of a cat from four regions (classes): auditory (10 samples), frontolimbic (19), somatosensory (18) and visual (18).

The dissimilarity dataset Chickenpieces-20-60 is computed from the chickenpieces image dataset [43]. From these images the edges are extracted and approximated by segments of length 20, and a string representation of the angles between the segments is derived. The dissimilarity matrix is composed by edit distances between these strings. The cost function between the angles is defined as the difference in case of substitution and as 60 in case of insertion or deletion.

The CoilYork dataset is composed by dissimilarities between graphs derived from four objects of the COIL database, the graphs are the Delaunay triangulations derived from corner points of the images [44]. The dissimilarity matrix is constructed by graph matching, using the algorithm of [45].

The CoilDelftDiff dissimilarity dataset is also computed from a set of graphs derived from four objects of the COIL database. The graphs are the Delaunay triangulations derived from corner points of the images [44]. Graphs are compared in the eigenspace with a dimensionality determined by the smallest graph in every pairwise comparison by the JoEig approach [46].

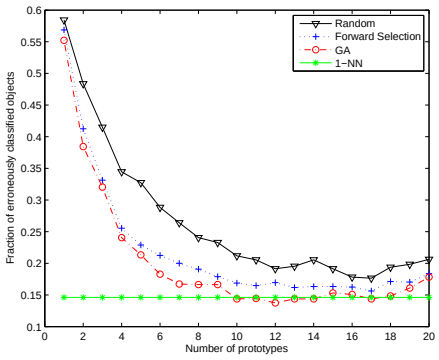
Average classification errors over 40 repetitions of the 1-NN in the dissimilarity matrix and

Table 3.1: Properties of the datasets

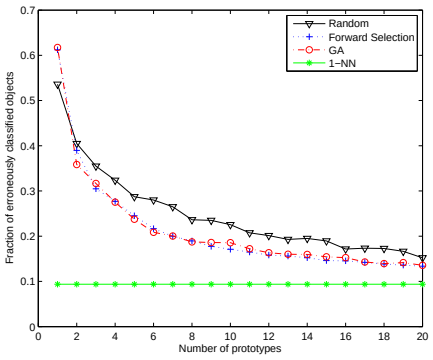
Datasets	# classes	# objects per class	symmetric	metric
CatCortex	4	10,19,18,18	yes	no
ChickenPieces-20-60	5	117,76,96,61,96	no	no
CoilYork	4	4x72	no	no
CoilDelftDiff	4	4x72	yes	no

LDA in the dissimilarity space for the random, forward and GA selection as a function of the number of prototypes are presented in Fig. 3.1.

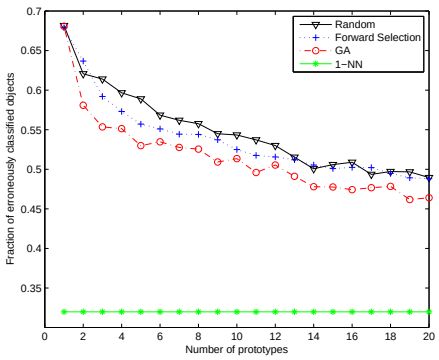
It was observed that in a number of cases for small sets of prototypes the GA search can find better solutions than the forward search as it can be seen in the CoilYork, CatCortex and CoilDelftDiff datasets. But in other cases as in the ChickenPieces-20-60 dataset the GA cannot improve the solution found by the forward selection. In the CatCortex and CoilDelftDiff with prototype selection it is possible to reach or improve the 1-NN classification results at a lower computational cost since the 1-NN needs to measure all the dissimilarities to the training objects, and with prototype selection only the dissimilarities to the prototypes are measured. In the other datasets the 1-NN has a smaller error but at the cost of a superior computational demand.



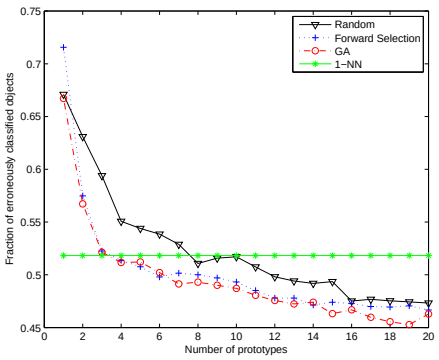
(a) CatCortex dataset



(b) Chickenpieces-20-60 dataset



(c) CoilYork dataset



(d) CoilDelftDiff dataset

Figure 3.1: Classification results of LDA in the dissimilarity space and 1-NN on the original dissimilarities as reference

3.1.4 Discussion and conclusion

The problem of prototype selection for building dissimilarity spaces is related to that of feature selection for feature spaces. There are, however, a few significant differences. In contrast to feature sets, the set of candidate prototypes (the training set of objects) constitutes a homogeneous field: neighboring object have similar properties as prototypes. There is not really a scaling issue because all dissimilarities are in the same range. As given dissimilarities may result from clustered sets of objects and are often non-Euclidean, the search for good sets of prototypes may be hampered by local minima. On the positive side it should be mentioned that the given

dissimilarities may be used for a fast computation of separability criteria. This makes the use of GAs desirable as well as feasible.

In this paper we have shown by a set of experiments on given dissimilarity matrices that GAs may be used successfully to construct good dissimilarity spaces. There is not always a need to do this. Forward search procedures as well as even random selection may also do well. For some problems, of which we have given clear examples, GAs perform better, especially when small sets of prototypes (e.g. one to twenty) are needed and also when training sets are small and complicated. It is of importance when fast classifiers have to be built in dissimilarity spaces maximizing the classification accuracy. For larger training and prototype sets, GAs might need a careful selection of their parameter values.

Acknowledgment

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250) as well as the project "Cálculo científico para caracterización e identificación en problemas dinámicos" (code Hermes-10722) granted by Universidad Nacional de Colombia.

3.2 Scalable prototype selection by genetic algorithms

Parts of this section has been published as "Towards Scalable Prototype Selection by Genetic Algorithms with Fast Criteria", by Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Heydi Me'ndez Vazquez, Edel Garcia-Reyes and Robert P. W.Duin, in *proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, S+SSPR 2014, LNCS. A version of the chapter will be submitted to the Pattern Recognition journal.*

Abstract

Classification in the dissimilarity space has become a very active research area since it provides a possibility to learn from data given in the form of pairwise non-metric dissimilarities, which otherwise would be difficult to cope with. The selection of prototypes is a key step for the further creation of the space. However, despite previous efforts to find good prototypes, how to select the best representation set remains an open issue. In this paper we study how to select the set of prototypes out of very large datasets. We propose two methods based on genetic algorithms which optimize unsupervised and supervised scalable criteria for this purpose. The unsupervised criterion is based on the Minimum Spanning Tree of the graph created by the prototypes as nodes and the dissimilarities as edges. The supervised criterion is based on counting matching labels of objects and their closest prototypes. The suitability of these type of algorithms is analyzed for the specific case of dissimilarity representations. The experimental results showed that the methods select good prototypes taking advantage of the large datasets, and they do so at low runtimes.

Introduction

The vector space representation is a common option to represent the data for learning tasks since many statistical techniques are applicable for this kind of representation. However, there is an increasing number of real-world problems which are not vectorial. Instead, the data are given in terms of pairwise dissimilarities which may be non-Euclidean and even non-metric. In [3] several approaches were presented to learn from dissimilarity data, where the dissimilarity space (DS) has several advantages over the other approaches. In the DS approach, the dissimilarities of the training objects to the representation set are interpreted as coordinates of a vector space.

A careful selection of the prototypes is needed to build a "good" DS based on a small set of prototypes. However, a random selection was found to perform well for large numbers of prototypes [11]. Since this is the fastest method, the selection of good prototypes by more dedicated methods is only of interest for small numbers of prototypes. A good method must be able to find a minimal set without a significant decrease in the accuracy of classifiers in the DS.

Several methods have been proposed [11, 47] to find small representation sets by supervised or unsupervised strategies. Supervised methods have the advantage of maintaining a high accuracy, however this is achieved at high computational costs; besides, they might suffer from overfitting. Unsupervised methods have the advantages of being fast and avoiding overfitting. However, as a disadvantage, they are not always good in maintaining class separability since class labels are not taken into account.

In this paper we focus on the selection of prototypes for classification in the DS. Our more specific target is to learn from large datasets, thereby we must design scalable procedures since it is known that the full search for the best prototypes set is intractable. This has been overlooked so far and only some studies such as the one in [16] cope with the problem. The analysis of large datasets is of interest because, nowadays, vast amount of data arise due to dropping costs for capturing, transmitting, processing and storing. There are also modalities that have millions of classes such as biometrics and others that present hundreds of thousands samples such as brain tractography data in [16].

We propose to use genetic algorithms (GAs) for scalable prototype selection with a fast clustering replacing the random initialization of the standard GAs. We discuss whether the proposed procedures are good for this purpose and on which situation they are better than other procedures. In addition, we propose a modification to the principal component analysis (PCA) method to compute the intrinsic dimension of large datasets.

The selection of prototypes seems similar to the selection of features for feature spaces.

However, the interpretation of features is different from the interpretation of prototypes since features might be very different and unique while dissimilarities relate similar objects. Therefore, adequate methods for selecting features are not necessarily adequate to select prototypes. A set of prototypes as well as vectors of dissimilarities are homogeneous in the sense that they represent values of the same dissimilarity measure; however a set of features may be very different since features may correspond to different type of measurements, even not numerical e.g. categorical. Thereby, the comparison of features is ill-defined. In contrast, the comparison of objects is well defined and it can be performed in a natural manner by an expert-defined dissimilarity measure or by an appropriate distance measure. In large datasets, objects have similar neighbors that may cause (after replacement) just slightly better or worse results. These are convenient properties that encourage the use of GAs.

It is usually assumed in the literature that linear-time algorithms are acceptable for scaling up to large datasets [48]. We also adopt this assumption. The parameter that dominates the complexity in our problem is the number of samples, since we assume that the other parameter, the number of selected prototypes, will always be small. We propose new prototype selection methods which may receive an already computed full dissimilarity matrix that fits into memory, and also address the case when the full dissimilarity matrix do not fit into memory, and the approaches have to compute or load the dissimilarities on demand. Some strategies to cope with scalability [48] include parallelism, techniques to work with data that do not fit into memory, stochastic methods, etc. In this study we focus on the time complexity. The space complexity is quadratic (the size of the full dissimilarity matrix) for the case when the full matrix fits into memory. In the case that dissimilarities are computed on demand, the space complexity is smaller but at the cost of increased time complexity as a function of the complexity for dissimilarity computation.

In [20], a GA was proposed for prototype selection but it is not able to cope with scalability issues. Here, we develop two versions of a scalable GA that optimize two different criteria for prototype selection on data where dissimilarities are either given or must be computed on demand. The remaining part of the paper is organized as follows: Section 3.2.1 presents the dissimilarity representation and prototype selection, Section 5.1.2 presents the two proposed methods and their complexity analysis for the case when dissimilarities are given and Section 3.2.5 presents their modifications and new analysis for the case when dissimilarities must be computed on demand. Section 3.2.7 presents our proposal for estimating the linear intrinsic dimension for large datasets. Experimental results are reported and discussed in Section 4.1.6 and conclusions are drawn in Section 3.2.10.

3.2.1 Dissimilarity space and prototype selection

The dissimilarity space was proposed by Pekalska and Duin [3]. It was postulated as a Euclidean space, which allows the use of all the classifiers that assume such space. Let X be the space of objects which may not be a vector space, but an input space of row measurements. Let $R = \{r_1, r_2, ..., r_k\}$ be the set of prototypes such that $R \in X$, and let $d : X \times X \rightarrow \mathbb{R}^+$ be a suitable dissimilarity measure for the problem. The prototypes may be chosen based on some criterion or even at random; however, the goal is that they have good representation capabilities specially when pursuing small representation sets. For a finite training set $T = \{x_1, x_2, ..., x_l\}$ such that $T \in X$, the dissimilarity space is created by the data dependent mapping $\phi_R^d : X \rightarrow \mathbb{R}^k$ where:

$$\phi_R^d(x_i) = [d(x_i, r_1)\ d(x_i, r_2)\ ...\ d(x_i, r_k)]. \tag{3.2}$$

Many methods have been proposed for the selection of prototypes in small-sized datasets. The study in [11] presents the Kcentres cluster-based method, an editing and condensing method, as well as supervised methods such as the forward selection (FS) to select prototypes.

However, only the Kcentres is able to scale to large datasets. The methods proposed in [47, 49] to select prototypes for strings and graph problems maintain in general a linear computational complexity. However, only two of the methods have good performances: the Kcentres and a version of the farthest first transversal (FFT) which they call spanning prototype selector. Thereby, we include them in our comparisons. We assume we have a validation set V which is used to select the prototypes out of it. The validation set is also used to compute those selection criteria that involve dissimilarity computations between the prototypes and objects in the dataset. In very large datasets we can afford large validation sets, but at the same time this poses the challenge to create methods which are able to deal with such large sets. It is worth noting that Kcentres has a complexity of $O(nk + k * (n/k)^2)$ and FFT has a complexity of $O(nk)$, where $k = |R|$ and $n = |V|$. For the sake of simplicity we say that they both behave as linear or almost linear time algorithms. The method proposed in [16] presents a variant of FFT that uses a sample of the dataset. However, the sampling decreases performance of the original version. Thereby, we use the version computed on the full dataset for our experiments.

3.2.2 Proposed methods

We propose two different variants of GA which are designed in order to be fast for prototype selection. The two methods receive as parameter the desired number of prototypes. Finding an appropriate number of prototypes for each particular problem is of interest since it allows one to save computing time and execute prototype selection methods only for a particular cardinality of the set. A good practice is to find the intrinsic dimensionality and select the number of prototypes accordingly. In our setup, we may or may not have a square dissimilarity matrix D computed among all training samples. The prototypes will be selected from the set of samples. Dissimilarities must be measured with these prototypes in case they are not already computed. Note that our goal is not to generate new prototypes as combinations of the original ones, but to select ones that already exist.

The GA is a search method based on heuristics that mimic the natural evolution mechanisms, by evolving individuals (chromosomes or solutions) created after each generation by the best fitted ones. This property of GAs makes them much better scalable than using a full search. In our problem, each individual is a set of prototypes of fixed cardinality k codified in a $k - vector$ containing in each position the index of the potential prototype. For example, the $5 - vector$ $(65, 30, 7, 19, 87)$ codifies an individual representing a set of 5 potential prototypes which can be accessed in some data structure by the indexes 65, 30 and so on. The GA usually starts the search in an initial population of randomly generated individuals.

Before executing the GA, we propose to perform a nearest prototype clustering assignment to randomly chosen centers in the set of candidates to prototypes to find k clusters. The number of clusters equals the desired number of prototypes. The candidates are clustered in order to guide the GA search in such a way that it has a faster convergence. The clustering runtime is $O(nk)$, where $n = |V|$ and $k = |R|$. The GA is slightly modified since its initial population is now generated by more restricted chromosomes. Each prototype represented in a position or gene of a chromosome is randomly sampled from a different cluster. In each generation, the best solution according to the fitness function is found and reproduced with each member of the population with a preset probability for the genes using uniform reproduction or crossover. Elitist selection is performed since the best fitted individual is retained for the next generation without undergoing mutation. In addition, only the best fitted individual is selected as parent of the next population of individuals. The rest of the population undergoes gene mutation with a preset probability which is usually small. We keep the constrain that the new index codified in a gene must belong only to the specific cluster linked to the gene. The pseudo-code for the case where dissimilarities are already given is presented in Algorithm 1.

In order to be fast and achieve full scalability, these methods should be able to handle: (1)

Algorithm 1: Scalable Genetic Algorithm

Input: D: dissimilarity matrix among samples and candidates to prototypes; k: desired number of prototypes, S: number of individuals in the population, rp: reproduction probability, mp: mutation probability, $iter$: number of generations

Output: bestindividual: set of prototypes indexes

```
// perform a nearest prototype clustering assignment to randomly chosen
   centers in the space of candidates to prototypes to find k clusters
1  cluslabs ← NN cluster asignment(D, k);
// randomly generate the population ensuring that, in the j-th position
   of the individual, only objects belonging to the j-th cluster are
   allowed
2  P ← GenerateInitialPopulation(cluslabs, D, k, S);
3  bestindividual ← P[1];
// find the best solution from the population and assign it to
   bestindividual
4  foreach currentindividual in P do
       // Note that the next line is where the proposed selection criteria
          must be used as the Fitness function
5      if Fitness(currentindividual, D) > Fitness(bestindividual, D) then
6          bestindividual ← currentindividual;
7      end
8  end
9  while number of generations < iter do
       // Evolution cycle
10     foreach currentindividual in P do
           // Reproduction, replace a gene of currentindividual with
              probability rp by a gene of the best solution
11         Reproduce(bestindividual, currentindividual, rp);
           // Mutation, change a gene of currentindividual with probability
              mp
12         Mutate(currentindividual, mp);
13     end
       // find the best solution from the population and assign it to
          bestindividual
14 end
```

large sets of candidates for prototypes, (2) large numbers of individuals in the search space of the GA and (3) large number of samples ($|V|$) to be used (if the selection criterion requires it) to compute the fitness function. In our proposal the GA handles well (1) large sets of candidates for prototypes since we discarded the standard binary codification of individuals that demands vectors of length equal to n where $n \gg k$. Instead, we resorted to vectors of length equal to the number of prototypes k since we codify only the indexes of the prototypes to be evaluated. Scalability in the number of individuals to analyze in the search space (2) is also achieved since the stopping condition is a small predefined number of GA generations that does not depend on the number of individuals in the search space. A small number of generations is sufficient for GA's convergence thanks to its guided sampling since not all the possible combinations of prototypes are explored but only the best ones which arise after each generation. In addition, the initial clustering helps to avoid redundant prototypes in the same individual. Scalability in

the number of samples to be used in the computation of the fitness function (3) will be explained in the next subsections.

3.2.3 Minimum spanning tree-based unsupervised criterion

In the fitness function computation, the set of k prototypes being evaluated as well as the n validation samples are usually involved. The proper number of prototypes k depends on the intrinsic dimension of the data which is usually small, thereby, $n \gg k$. For large datasets, this implies that the dominant term for the fitness computation is the total number of samples n. To achieve scalability in the fitness function, it must be able to scale to a large n. This highly depends on how the criterion to be optimized in the fitness function by the GA is conceived. Our first proposal for GA criterion is based on the minimum spanning tree (MST) of a set of prototypes. Prototypes are interpreted as nodes in a graph and dissimilarity values between prototypes correspond to edge weights. The sum of edge weights (usually named tree weight) is used as criterion to be maximized, thereby increasing the diversity of the prototypes and improving the coverage over the DS.

The MST weight is related to the Rényi entropy of the set of prototypes [50]. This relation is monotonically increasing: the entropy of the set of prototypes increases as the MST weight increases. The entropy is also a type of diversity measure which confirms our intuition that higher MST weights are related to higher diversity of the prototypes.

As we used the Prim's algorithm to find the MST and the graph is complete, the computation of this criterion from an already constructed graph has a runtime of $O(k^2 \log(k))$. Therefore, it is independent on the large number of samples n and, as a consequence, highly scalable for very large problems. The pseudo-code is presented in Algorithm 2. The total runtime of the proposed GA with this criterion is as follows. For computing the initial clustering of the prototypes the runtime is $O(nk)$, for the fitness function $O(k^2 \log(k))$, and $O(k)$ for mutation and reproduction since each position of the vector representing an individual has to be analyzed. The dominant term in the whole procedure is $O(nk)$, as we assume that the desired number of prototypes is small and fixed, therefore the total runtime is $O(nk)$. However, if the initial clustering step is not performed, the complexity is only $O(k^2 \log(k))$, which in case of requiring sub-linear (in n) methods is more appropriate.

3.2.4 Supervised criterion based on counting matching labels

Our second proposed criterion is a linear-time supervised criterion that is different from previous expensive supervised ones [51] since it does not compute a classification error in the DS or an intra-class distance, which are usually quadratic. Our method, instead, considers each candidate for prototype as a representative of a cluster and every object in V is assigned to the cluster represented by its nearest prototype. The proposed criterion counts the number of assigned objects whose class labels match their representative class label. The best solution is the one that maximizes this value. This has the smallest runtime for a supervised method that analyzes all the samples with relation to the prototypes ($O(nk)$). The pseudo-code is presented in Algorithm 3.

The runtime of the complete supervised GA is as follows. For computing the clustering the runtime is $O(nk)$, for the fitness function $O(nk)$, and for mutation and reproduction $O(k)$. The total runtime is $O(nk)$. However, in practice, this is higher than the unsupervised procedure since it is multiplied by a higher constant due to the cost for comparing the class labels. In general, these times are better than or comparable to other linear or almost linear (in n) algorithms compared in our experiments such as the Kcentres and the FFT.

Algorithm 2: Unsupervised Fitness function by MST

Input: w: vector of prototypes indexes; D: dissimilarity matrix
Output: fitnessvalue: fitness value
// Interpret the prototypes indexed in w as nodes and dissimilarities among them stored in D as edges weights of a complete graph G = (v,e)
// compute minimum spanning tree by Prim's algorithm
1 $v' \leftarrow v[1]$;
2 $k \leftarrow |v|$;
3 $e' \leftarrow \emptyset$;
4 **while** $|e'| < k - 1$ **do**
  // select an edge of minimum weight which connects one node in v' with a node which is not in v'
  // add the new edge to e', add the new node to v'
5 **end**
6 $MST \leftarrow (v', e')$;
// sum all the weights of edges e' in MST
7 $fitnessvalue \leftarrow$ SumWeights(MST);

Algorithm 3: Supervised Fitness function based on counting matching labels

Input: w: vector of prototypes indexes; D: dissimilarity matrix
Output: fitnessvalue: fitness value
// interpret the prototypes r_j indexed in w as centers of clusters
1 $fitnessvalue \leftarrow 0$;
2 **foreach** $x \in V$ **do**
  // find the nearest prototype of x
3 $r' \leftarrow$ argmin($D[x, r_j]$);
4 **if** getclasslabel(x)=getclasslabel(r') **then**
5 $fitnessvalue \leftarrow fitnessvalue + 1$;
6 **end**
7 **end**

3.2.5 Proposed GAs when dissimilarities must be computed on demand

In this section we assume that datasets are so large that the presented approaches that load the full dissimilarities into memory are not feasible and dissimilarities have not been pre-computed and stored. The dissimilarities must be computed on demand using a proper dissimilarity measure for the problem. In addition, the data is given by raw measurements or some other intermediate representation, e.g. images/time signals/graphs and we assume that finding a dissimilarity between any two objects needs q computations. The proposed GAs are modified accordingly allowing scalability for such large datasets where the storing of the full dissimilarity matrix into memory is not possible. In this section we will describe the main components of the proposed GAs that change in this new situation as well as their computational complexity analysis. We assume that the complexity of the dissimilarity measure is linear in the number of measurements. This is true, for example, for distances such as those of the Minkowski family which contains the widely used Euclidean one.

First, in the clustering, the nearest prototype clustering assignment to random centers must include the computation of dissimilarities among the randomly initialized prototypes and all the samples, but this is performed only once so the total cost is $O(nkq)$, being q the number

of measurements. The two important links of a general purpose GA with an specific problem are the encoding of solutions or individuals and the fitness function. The encoding of solutions is affected by the need to compute dissimilarities on demand since we access now to the object instead of its already pre-computed dissimilarities with other objects. In the next subsection the modifications needed for the proposed fitness functions are explained.

3.2.6 Unsupervised and supervised fitness function modifications

In the case of the unsupervised fitness, the all-against-all pairwise dissimilarities among the candidate prototypes indexed in an individual must be computed. The second step is to compute the fitness value which includes the MST construction. The total fitness function, including first the computation of the square pairwise dissimilarity matrix for the prototypes and second the MST computation, takes $O(k^2q+k^2\log(k))$, which boils down to $O(k^2(q+\log(k)))$. In case the clustering step is used, the total runtime of the unsupervised GA is $O(nkq+k^2(q+\log(k)))$. If the clustering step is not used, the GA method takes $O(k^2(q+\log(k)))$.

The supervised fitness function must also be reformulated when the dissimilarities between the samples are not given. In this case, the dissimilarities between all the objects and the prototypes must be computed before the criterion. The total complexity of the fitness function, including the computation of dissimilarities between objects and prototypes, is now $O(nkq)$. The total runtime remains the same whether the clustering step is used or not. The computational complexities for both cases, when the dissimilarities are given as in [21], or when they must be computed on demand are summarized in Table 3.2.

Table 3.2: Computational complexities when dissimilarities are given and when dissimilarities are computed on demand

Type of problem	Fitness function	GA with clust.	GA no clust.
Given diss. unsup.	$O(k^2\log(k))$	$O(nk+k^2\log(k))$	$O(k^2\log(k))$
Given diss. sup.	$O(nk)$	$O(nk)$	$O(nk)$
On demand diss. unsup.	$O(k^2(q+\log(k)))$	$O(nkq+k^2(q+\log(k)))$	$O(k^2(q+\log(k)))$
On demand diss. sup.	$O(nkq)$	$O(nkq)$	$O(nkq)$

3.2.7 Intrinsic dimension estimation for large datasets

One drawback of the proposed methods is that they rely on a given number of prototypes and thereby, on the dimension k of the DS. In practice, the proper k to use must be estimated from the intrinsic dimension (ID) of the data. We study how to find the ID for large datasets. The ID of a dataset is the smallest number of variables needed to describe the data properly. In the pattern recognition context, this is usually referred to as the dimension of the subspace or manifold where the objects lie. The term degrees of freedom is sometimes used as well. If the ID is unknown, the prototype selection methods must be executed for several values of k to find the one who leads to the best compromise between accuracy and efficiency. We avoid this by finding the ID k by an ID estimation method.

The standard linear ID estimator is the Principal Component Analysis (PCA) approach. The general idea is that, after centering the data by substracting the mean, PCA estimates the covariance matrix of the data and finds its eigenvalues and eigenvectors. Given the training set $T=\{x_1,x_2,...,x_l\}$, and the data mean $\phi=\frac{1}{n}\sum_{i=1}^{l}x_i$, the covariance matrix C is given by:

$$C=\frac{1}{n-1}\sum_{i=1}^{l}(x_i-\phi)(x_i-\phi)^T=AA^T \tag{3.3}$$

Next, the eigenproblem $Cv = \lambda v$ is solved and the eigenvalues are sorted in descendent manner and scaled in a way that their summation is 1. The cumulative sum is computed until some predefined threshold is reached, usually 0.98 or 0.95. The ID k corresponds with the rank where the threshold was reached. The interpretation for such selection in terms of the PCA is that the maximum variance of the data can be explained with k dimensions and the other dimensions are representing noise. However, the eigendecomposition is $O(l^3)$ for covariance matrices, thereby its standard version is not adequate for very large datasets.

Here, we adapted the PCA for ID estimation on large datasets by resorting to data sampling. To support the proposal, we study the influence of different sampling cardinalities on the PCA results. In addition, PCA has a "trick" [52] that makes it suitable for large datasets if the number of prototypes is much smaller than the number of samples. This comes from the fact that eigenvectors of the covariance matrix AA^T are equal to the eigenvectors of the smaller matrix A^TA multiplied by A up to scaling. If this holds, the PCA complexity is $O(q^3)$, being q the number of prototypes. As the complexity depends on the minimum between the number of objects and the number of prototypes, we decide to select a sample of the large dataset and use it both as the set of objects and as the set of prototypes. However, in order to detect the intrinsic dimension of the dataset, the cardinality of the sampled set Z must satisfy: $|Z| \gg$ID. We can apply the mentioned trick to our dissimilarity representation. The standard PCA can operate in a DS created by the all-against-all dissimilarity matrix among the randomly sampled set of objects of moderate cardinality (e.g. 1000).

One drawback of applying PCA is that it assumes that the data is linearly distributed and for some datasets it might not be the case. Therefore, in case of nonlinearly distributed data the actual ID would be smaller than the one returned by PCA.

3.2.8 Datasets and experimental setup

Four different datasets of small to medium size were used for the experiments considering that the full dissimilarity matrix fits into memory. A brief description of them is provided below:
Zongker data. It is computed by deformable template matching by Jain and Zongker [53]. The dissimilarity measure is the result of an iterative optimization of the non-linear deformation of a grid on the images of digits.
Pendigits dissimilarity data. It was created by Bunke and Spillman [49] for the data described in [54] by Alimoglu and Alpaydin. The digits are written by 44 different writers. An edit distance was computed for string representations. Note that both problems, Zongker data and pendigits data, are studied for recognition of handwritten digits.
XM2VTS face data. This dataset was created using face images from the XM2VTS database [55] which contains images of the same subject under different lighting conditions. It contains 3540 images corresponding to 295 subjects. All face images were cropped to 120x140 pixels and Local Binary Patterns (LBP) histograms were extracted. The chi square distance was computed on the histograms.
Diabetes dataset. The data comes from the National Institute of Diabetes and Digestive and Kidney Diseases (USA). It is available at UCI Machine Learning Repository [56]. One class is healthy individuals and the other is individuals with a higher risk of diabetes. The Euclidean distance was used on the original features.

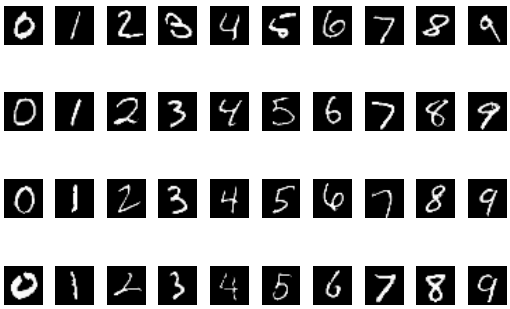
In addition, the following four datasets of large size were used for the experiments to test our versions of the proposed methods for the case when the full dissimilarity matrix does not fit into memory.
MNIST data. The dataset [57] was collected from subsets of NIST having a balanced number of digits written by high-school students and Census Bureau employees. The original black and white images from NIST were scaled to fit in a 20x20 pixel box while preserving their aspect ratio. The digits were centered in a 28x28 image by computing the center of mass of the digit

image pixels, and translating the digit image to fit the center in that of the 28x28 image. Our dissimilarities were created using Euclidean distances on the 28×28 images.
Street View House Numbers data. The dataset is used for the purpose of digit recognition in the wild [58]. We divided it into SVHN1 and SVHN2 depending on which of the available sets (the smaller with 73257 or the larger with 531131 images) is used as validation set. Examples of digits from MNIST and this dataset are shown in Fig. 3.2. For SVHN1 and SVHN2 we had to process the 32×32 digit images first since they are very noisy. The digit histograms were equalized and the resulting image intensities were scaled by a Gaussian-shaped function emphasizing the middle of the images which actually contain the digits and gradually giving less weight to farther pixels which contain noise. Next, the images were blurred using a 5×5 Gaussian kernel with σ parameter equal to 0.5. SVHN1 is the smaller dataset with 73257 images containing the most difficult samples. SVHN2 is larger, with 531131 extra data containing easier digits. These sets are later partitioned into two sets, one for prototype selection and one for training. The standard test set of 26032 images is used for testing.
YouTube Faces dataset. The original version of YouTube faces database [59] is composed by face videos and it was designed to investigate unconstrained face recognition. It contains a total of 3,425 videos from YouTube of 1,595 subjects. The shortest clip contains 48 frames, the longest clip contains 6,070 frames, while the average length of a video clip is 181.3 frames. In the database, there are 1045 subjects with at least one video having more than 100 frames. We used these subjects for our experiment, and only one video was selected for each person. The videos were split into frames to construct the datasets. LBP descriptors were computed for the normalized faces contained in the frames and Euclidean distances are used as dissimilarity measure, since they behave well on these descriptors. Besides, these Euclidean distances are in agreement with our computational complexity analysis since we assume the use of a linear time dissimilarity measure. The characteristics of the datasets as well as the cardinality of the training sets used are summarized in Table 5.3.

Table 3.3: Characteristics of the datasets used in this study, the last column ($|V|$) refers to the validation set cardinality used for the experiments

Datasets	# Classes	# Obj	Metric	$\|V\|$
Zongker	10	200×10	no	1000
Diabetes	2	500/268	yes	384
Pendigits	10	10992	no	5000
XM2VTS	295	12×295	yes	1770
MNIST	10	70000	yes	60000
SVHN1	10	99289	yes	73257
SVHN2	10	630420	yes	531131
YouTube	1045	308963	yes	247170

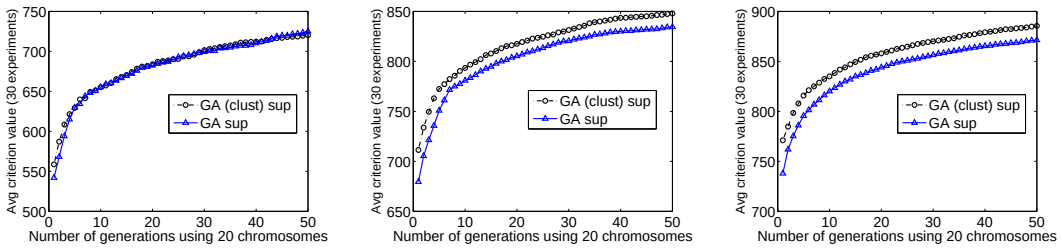
The small and medium-sized datasets were divided into validation, training and test set 30 times. The validation is used to select the prototypes out of it and to compute the selection criterion. In the case of MNIST, the standard training and test set division was used, except that the training set was randomly divided 10 times into one set for validation with 98% of the total objects (without considering the test set), and the other 2% was used to train the classifier. This was done since our purpose is to show that the selection methods scale well to large datasets and still find good prototypes. Small training sets were used since they give us the opportunity to analyze better if the selection with the proposed methods is successful since, in some cases, a large training set may compensate for an inadequate prototype selection. Similarly, in the case of SVHN1 we randomly divided 10 times the difficult set of 73257 images to use 98% for selecting the prototypes and 2% to train the classifiers. We also carried out an



(a) MNIST

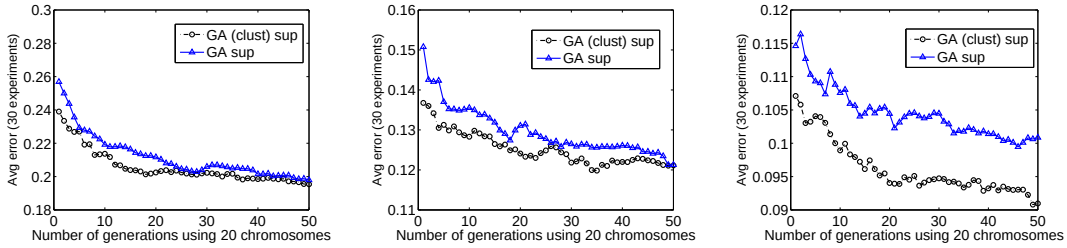
(b) SVHN1

Figure 3.2: Examples of grayscale images from the digits datasets used in the experiments



(a) 10 prototypes (b) 20 prototypes (c) 30 prototypes

Figure 3.3: Average criterion values for different numbers of prototypes and different number of generations of the GA for Zongker data



(a) 10 prototypes (b) 20 prototypes (c) 30 prototypes

Figure 3.4: Average errors for different numbers of prototypes and different number of generations of the GA for Zongker data

experiment on SVHN2 where we selected the prototypes out of the extra set of SVHN containing 531131 images, we train with a subset of 53113 images randomly selected and test with the standard test set, to show the scalability for half a million images.

In the case of the YouTube dataset, we randomly divided the data into 80% for validation from which the 0.05% was used for training, and 20% for testing. The globally best performing classifier per dataset between the linear discriminant classifier (LDC), which is the Bayes classifier assuming normal densities with identical covariance matrices, and the 1-NN classifier was used to report the classification errors for the different prototype selection methods compared in the medium-sized dataset, the compared methods are:

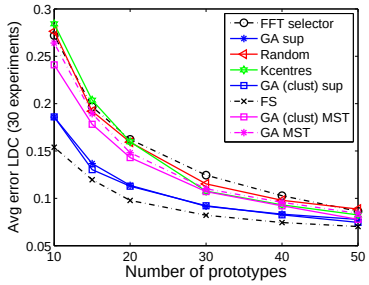
- Random selection
- Forward selection [11] optimizing the supervised criterion
- FFT [16]
- Kcentres [11]
- GA in the space of clustered prototypes with the proposed unsupervised fitness function based on MST (GA (clust) MST)
- GA with the proposed unsupervised fitness function based on MST without clustering the prototypes (GA MST)
- GA in the space of clustered prototypes with the proposed supervised fitness function (GA (clust) sup)
- GA with the proposed supervised fitness function (GA sup)

In the case of the large datasets the classifiers tested were the 1-NN and the quadratic discriminant classifier (QDC) since we can afford to estimate different covariances for large datasets. In addition, the QDC outperforms the LDC for all the procedures on these datasets. The FS was not considered for the experiments on the large datasets due to its lack of scalability.

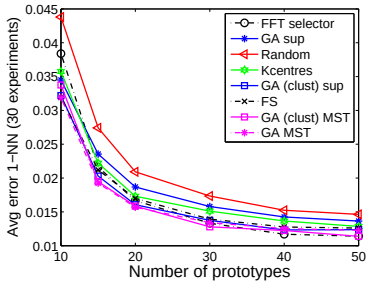
The parameters used for the GA are: 20 individuals for the initial population, 0.5 for probability of reproduction per gene, and 0.02 for probability of mutation per gene. The stopping condition is 20 generations reached for the GA with initial clustering in the space of prototypes, and 25 for the GA without the clustering for the medium-sized datasets. These numbers are different in order to show that the clustering allows a faster convergence of the GA even when some advantage is given to the version without the clustering. For the large-sized datasets, both versions use 20 generations as stopping criterion and the probability of mutation is increased

to 0.1 to allow a higher diversification of the solutions and therefore better exploration of the large search space.

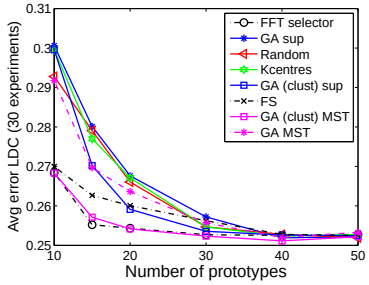
3.2.9 Results and discussion



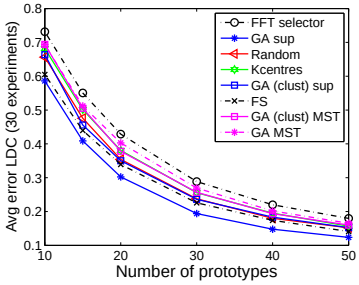
(a) Zongker LDC



(b) pendigits 1-NN

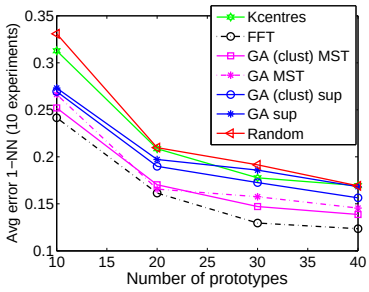


(c) Diabetes LDC

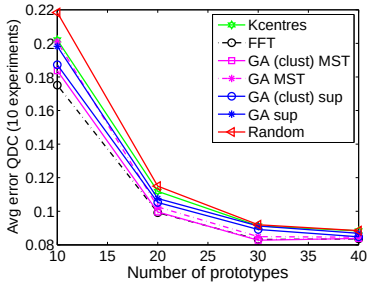


(d) XM2VTS LDC

Figure 3.5: Average errors for different numbers of prototypes for the best performing classifier in each dataset



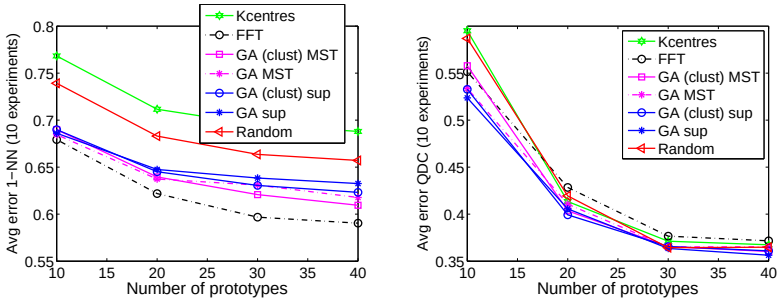
(a) MNIST 1-NN



(b) MNIST QDC

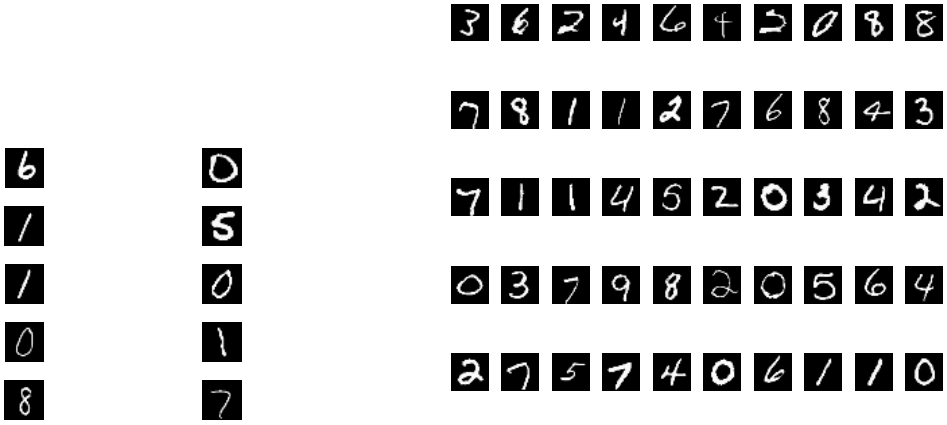
Figure 3.6: Average errors for different numbers of prototypes in the MNIST data

Figure 3.3 presents the criterion values convergence for different numbers of prototypes. Figure 3.4 presents the test errors related to the criterion values in Fig. 3.3 for the different GA generations. We observe that 20 generations provide a good compromise for acceptable classification error at acceptable runtime. We used this number of generations (20) in all the



(a) SVHN 1-NN (b) SVHN QDC

Figure 3.7: Average errors for different numbers of prototypes in the SVHN data



(a) Two prototypes (b) Ten prototypes

Figure 3.8: The prototypes selected by different methods on the MNIST dataset from top to bottom: GA (clust) MST, GA MST, GA (clust) sup, GA sup, random selection

experiments of the GA with initial clustering.
Figure 3.5 presents the average errors over 30 experiments for 10, 15, 20, 30, 40 and 50 prototypes for datasets where dissimilarities are given. In the case of the MNIST and SVHN datasets the dissimilarities must be computed on demand, Figs. 3.6 and 3.7 show the results averaged over 10 experiments for 10, 20 30 and 40 prototypes.

It can be seen in Fig. 3.5(a) that, for the Zongker dataset, the best results are obtained with the FS and the supervised criterion, also the proposed GA with this criterion outperforms the other unsupervised methods in accuracy with a comparable efficiency. The FS outperformed the GA in this dataset because it does not present a significant class overlap. However, when there is a significant overlap among the classes as in XM2VTS (see Fig. 3.5(d)), the GA outperforms the FS since it takes into account the contribution of the prototypes as a whole. One problem of the FS that causes its lower performance is that for a new object to be added to the set, its contribution is only analyzed with the already selected objects. Thereby, we loose the possibility of finding better solutions that involve the new object if they exist.

From results on Diabetes dataset in Fig. 3.5(c), it can be seen that the proposed GA with the

Table 3.4: Classification errors using a training set of 12358 objects for Youtube when selecting 10, 20 and 30 prototypes out of 247170 objects set. Best results are in bold.

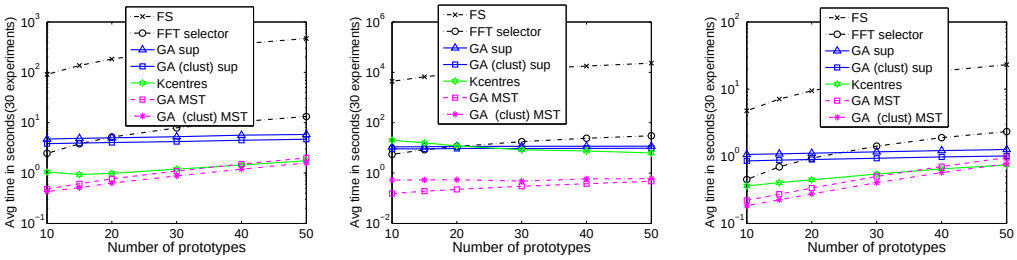
Method \ # Prot.	10	20	30
GA(clust)MST+QDC	**0.3431**	0.1831	0.1685
GA MST+QDC	0.3617	0.1936	0.1623
GA(clust)sup+QDC	0.3926	0.1957	0.1586
GA sup+QDC	0.3494	0.1986	0.1619
random+QDC	0.3551	0.1988	0.1634
FFT+QDC	0.3963	0.2006	0.1614
GA(clust)MST+1-NN	0.3694	**0.1423**	0.1121
GA MST+1-NN	0.4022	0.1655	**0.1064**
GA(clust)sup+1-NN	0.4503	0.1668	0.1121
GA sup+1-NN	0.4032	0.1900	0.1219
random+1-NN	0.4117	0.1927	0.1227
FFT+1-NN	0.4275	0.1555	0.1108

Table 3.5: Classification errors using a training set of 53113 objects and the standard test set of 26032 for SVHN2 when selecting 10, 20 and 30 prototypes out of the set of half a million objects. Best results are in bold.

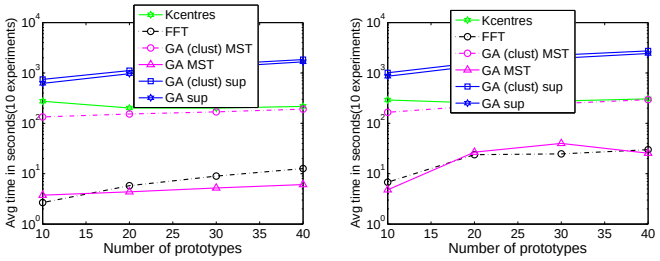
Method \ # Prot.	10	20	30
GA(clust)MST+QDC	0.4678	0.3175	0.2404
GA MST+QDC	0.5043	0.3247	0.2470
GA(clust)sup+QDC	0.4823	0.3090	0.2435
GA sup+QDC	0.4661	**0.3082**	**0.2402**
random+QDC	0.5197	0.3241	0.2480
FFT+QDC	0.4801	0.3136	0.2414
GA(clust)MST+1-NN	0.5308	0.3925	0.3582
GA MST+1-NN	0.5539	0.4015	0.3592
GA(clust)sup+1-NN	**0.4466**	0.3475	0.3224
GA sup+1-NN	0.4554	0.3742	0.3275
random+1-NN	0.5831	0.4209	0.4018
FFT+1-NN	0.5638	0.4141	0.3730

unsupervised criterion outperforms the other methods except for the FFT which has a similar performance. From results for Pendigits in Fig. 3.5(b) we find again that the GA with the unsupervised criterion is among the best performing methods both in speed and accuracy (see Fig. 3.9(b)). Regarding our second question whether cluster analysis may be helpful, we see that results with initial clustering are usually equal to or better than those without clustering the prototypes before executing the GA, except for the XM2VTS complicated dataset that presents a significant class overlap.

In the XM2VTS we find that the best method is the proposed supervised GA but without the clustering. A clear explanation of why this happens was derived from the data exploration by multidimensional scaling (MDS) in Fig. 3.10(d) and the knowledge of the dataset characteristics. This is a dataset with strong illumination changes: frontal, right, and left illuminations of the faces. The three different illuminations create three large clusters of objects, so they are determined by noise instead of by the identities. Due to this, the unsupervised clustering before

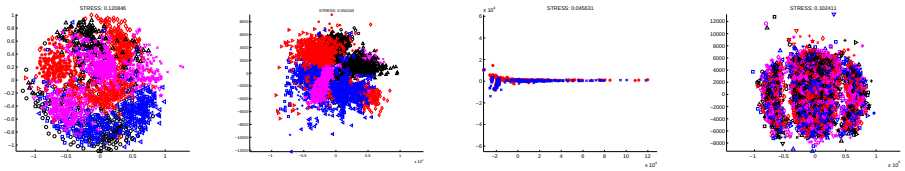


(a) Zongker (b) Pendigits (c) Diabetes

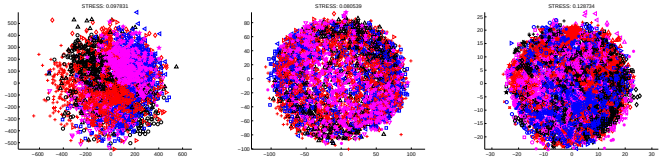


(d) MNIST (e) SVHN

Figure 3.9: Average times in seconds plotted in log scale



(a) Zongker (b) Pendigits (c) Diabetes (d) XM2VTS



(e) MNIST (f) SVHN (g) YouTube

Figure 3.10: MDS mappings

the GA does not find meaningful clusters for this complicated problem. However, our supervised criterion handles the class overlap better. Thereby, the supervised clustering assignment that we perform to evaluate our supervised criterion is not linked to the initial unsupervised clustering of the space of prototypes. These results show that the proposed GAs are good for prototype selection. In the next set of experiments, we will show that they are able to select prototypes out of very large datasets, without having to resort to the straightforward solution of randomly sampling the datasets.

In the experiments on the large-scale datasets we can see from Figs. 3.6 and 3.7 that the

1-NN benefits more from a systematic selection of the prototypes than the QDC. This can be seen from the improvements found when using the proposed systematic methods for selection compared to the random selection. By performing a t-test we detected that the improvements were statistically significant for the 1-NN in the MNIST and SVHN datasets. Our methods are almost always the best performing ones for the QDC classifier. However, the FFT is better than our procedures when using the 1-NN.

Figures 3.8(a) and 3.8(b) show the digits returned as prototypes for MNIST when the methods select the best two and ten. It can be seen that the GAs returns digits that differ from each other, manifesting the suitability of the used criteria to allow diversity in shapes of the set of prototypes. It is clear that the random selection is not able to achieve this diversity. The supervised GA returns perfectly handwritten digits, which can be expected since those digits are the ones that represent their classes better. Therefore, depending on the problem, the selected prototypes carry a meaning that can be used in other analysis different from classification. From the results on Youtube data in Table 3.4, it is interesting to see that with only 30 prototypes we obtain remarkably good results for a 1045-class problem. This means that we do not need objects from all the classes to generate a good DS. We use all the classes for training but only 10, 20 or 30 objects are used, in total, to create the space where the training set is mapped. The GA maximizing the MST outperformed the other methods in all cases.
Table 3.5 shows the classification errors when selecting the prototypes out of the set of half a million images. It can be seen that the supervised selection by GA provides the best results. Note that it uses the complete set of half a million objects to compute the selection criterion in each GA iteration, in contrast to the unsupervised version that only needs the dissimilarities among the prototypes being evaluated to compute the criterion. Thereby, it is useful to use as many objects as possible to compute the selection criterion for this difficult dataset. Since the computation of the supervised criterion is linear in the number of objects, the computation cost is affordable.

The computation times for some of the datasets reported in Fig. 3.9 show that the GA with unsupervised criterion is the fastest method in all the datasets, except for 10 prototypes in MNIST and SVHN1. The supervised proposal is comparable to other unsupervised methods. This analysis, together with the computational complexity analysis, indicates that the proposed GAs are able to scale well to large datasets when the final goal is the selection of a small set of prototypes. However, the fitness function must also be designed scalable. In addition, GAs are embarrassingly parallelizable, which highlight even more the benefits of GAs for scalable prototype selection. This means that they can be easily decomposed into parallel subtasks. This can be exploited to speed-up the process further.

We analyzed relations between performance of methods and data distribution by inspecting the MDS plots. We found that the supervised method handles well datasets with homogeneous distributions or with class overlap as the one in Fig. 3.10(a). In contrast, the MST-based unsupervised criterion copes better with non-homogeneous distributions where we can find, inside the same class, densely populated regions as well as sparse ones as in Fig. 3.10(b). In addition, the MST-based GA handles well easier problems and elongated classes as in the Diabetes dataset in Fig. 3.10(c). We found that among the competitors the FFT provided reasonably good results especially for easier problems and when using the 1-NN classifier. The FFT might be preferred in case of clear separable classes and the GA, especially the supervised version, for more difficult problems with overlapping classes. This can be expected since the GA refines the set of prototypes en each iteration while the FFT adds one prototype in each iteration and the method stops when the desired cardinality of the set is reached without any refinement of this set.

We study the ID estimation as a way to overcome the computation of prototype sets of different cardinalities which incur in higher computational costs. Our proposal is aimed to

validate the standard PCA as a tool for estimating the ID of large datasets by analyzing the variance on smaller subsets of the whole data. The datasets were sampled uniformly at random and we computed IDs for sample sizes of 200 to 60000, assuming that the important information is contained in a 95% of the data variance. Table 3.6 presents the IDs found by PCA for the different sample sizes, it can be seen that the results are quite stable when more than 1000 objects are used to compute the PCA. This supports our proposal of PCA for ID estimation for large datasets by resorting to a randomly sampled small subset of the data to execute the method. For the tested datasets of digit images with low ID, a moderate sample size (e.g. 1000 or 5000) is sufficient for ID estimation. However, for a large ID, a larger sample size is needed due to problems with the curse of dimensionality.
It is worth noting at the end of the discussions that the final solutions found by the proposed GAs are not affected by the fact that dissimilarities among objects and prototypes are computed on demand. The results are exactly the same as if they were obtained with the full dissimilarity matrix loaded in memory.

Table 3.6: Intrinsic dimension estimation for different sample sizes on SVHN1 and MNIST

# objects / dataset	200	500	1000	5000	10000	15000	20000	40000	60000
SVHN1 0.95	18	20	20	21	21	21	21	21	21
MNIST 0.95	37	46	49	52	52	52	52	52	52

3.2.10 Conclusions

The selection of prototypes is a crucial step for classification in the dissimilarity space. In this paper we proposed two different prototype selection methods by genetic algorithms and two scalable supervised and unsupervised criteria which are used for the fitness functions. Our work focuses on achieving low computational costs by exploiting the suitability of genetic algorithms to find good trade-offs between time complexity and accuracy of the solution and maintaining low asymptotic complexities in the fitness function. Experimental results showed the validity of the proposals for selecting good prototypes and the runtime analysis showed that the methods are able to scale to large datasets. Other general approaches to cope with scalability include parallelism, stochastic methods, among others.

The proposed unsupervised method is the fastest one since the evaluation of its criterion does not depend on the size of the dataset but on the number of prototypes. Besides, the linear time supervised criterion is also very fast compared to other supervised ones, which are generally quadratic and thereby do not scale to large datasets. Its computational complexity is comparable to previous unsupervised methods from the literature.

After comparing the unsupervised and supervised methods for the selection, a question arises: is the label of a prototype really relevant? Another object with the same dissimilarities to other objects but a different label will likely generate the same result. However, the use of labels, in general, allow us to emphasize that we search for different objects, improving the coverage over the space of objects. In addition, in our procedure, we also ask that each of these prototypes must represent its class as good as possible. These requirements make the procedure especially good for datasets with overlap among the classes. However, in other cases, unsupervised procedures may do equally well. Especially, if we are aiming at many more, or much less prototypes than classes, their labels will not help us.

We found that it is profitable in some cases to involve as many objects as possible in the computation of the selection criterion as we do in our supervised proposal. This holds especially for difficult datasets since we find improvements over the other methods when using this strategy. In addition, the computational burden when including even millions of objects in the criterion

computation is affordable, as the supervised criterion is linear in the number of objects. For not very complicated problems, the unsupervised selection that only uses the distances among the prototypes in the criterion computation is sufficiently good, and no further improvements are found by involving the large datasets in the criterion computation.

It is clear that the GA samples its search space in a clever way using the heuristics related to crossover and mutation, circumventing the infeasibility of a full search. In our case, we submit all objects in the dataset for clustering before its search is started, influencing the total amount of time needed to find the solution. However, in some cases it may be preferred to speed-up this further especially for very large datasets. A smaller candidate set may be created by some smart sampling of the dataset on which the GA may operate. The proposed initial clustering does a similar thing, but on the search space level instead of on the data level.

To complement our procedures, we studied the suitability of the principal component analysis for intrinsic dimension estimation of large datasets by sampling the data. We concluded that this technique is convenient for this purpose since its results are stable over different sampling cardinalities. As a requirement, these cardinalities should be larger than the actual intrinsic dimension of the data.

Chapter 4

Prototype models creation and selection

4.1 Selecting feature lines in generalized dissimilarity representations

This section has been published as "Selecting feature lines in generalized dissimilarity representations for pattern recognition", by Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Edel Garcia-Reyes and Robert P. W.Duin, in *Digital Signal Processing, Elsevier, 2013.*

Abstract

Recently, generalized dissimilarity representations have shown their potential for small sample size problems. In generalizations by feature lines, instead of dissimilarities with objects, we have dissimilarities with feature lines. One drawback of such generalization is the high amount of generated lines that increases computational costs and may provide redundant information. To overcome this, the selection of lines based on the length of the line segments has been considered in previous works, showing good results for correlated data. In this paper, we propose a new supervised criterion for the selection of feature lines. Experimental results show that the proposed criterion obtains competitive or better results than those obtained by previous criteria, especially for data with high intrinsic dimension, spherical data and data with outliers. As our proposal provides better results for small representation sets, it allows one to obtain a good trade-off between classification accuracy and computational efficiency.

4.1.1 Introduction

Dissimilarity representations (DRs) for pattern recognition [3] arose as an alternative to feature-based and structural representations. They can be built directly from raw data as well as on top of these two representations [17, 37]. Intuitively, we can realize that a class is constituted by a set of similar objects so (dis)similarities play an important role in the process of classification [3]. Sometimes, the data is given in terms of dissimilarities since it can be difficult, expensive, inconvenient or even impossible to extract good features to characterize the data. Also, for some problems, experts can define robust dissimilarities that incorporate expert knowledge and invariances.
Recently, there have been promising studies on the topic of classification in dissimilarity spaces [17]. A dissimilarity space is a Euclidean vector space defined by a set of prototypes. To embed a new object into the vector space, it is needed to compute the dissimilarities of the object with the set of prototypes. If a training set is available, classifiers can be trained in the dissimilarity space. One drawback is that, usually, these dissimilarities are obtained from matching processes that are computational expensive. Prototype selection is a way to approach this problem by reducing the number of prototypes to which an incoming object is going to be compared in order to be classified afterwards.

More recently, Orozco-Alzate et al. [31,36] introduced and developed the topic of generalized dissimilarity representations (GDRs) by feature lines and feature planes. These approaches are specially useful for small sample size problems or data under representational limitations. In these approaches, objects are represented by their dissimilarities with lines or planes. GDRs by feature lines give one the opportunity to exploit the geometry of the cloud of points distributed in the original feature space and thereby use more information. This geometry can even be captured when the original feature representation is not available and only the distances between objects are used.

For classification in generalized dissimilarity spaces, the feature lines are used as prototypes and not as the training set. The original objects constitute the training set, but the representation of these objects is constructed using the distances to the feature lines. One drawback is the high amount of lines that can be created since this increases the number of dissimilarity computations for posterior representation of training and test objects. Selecting the most representative feature lines is a way to overcome this drawback. Up to now, the feature lines selection methods considered have been the random selection and selection by the length of the feature line segments. New selection methods using discrimination ability of the feature lines may be more suitable to reduce the high computational costs while maximizing the accuracies of classifiers constructed in generalized dissimilarity spaces spanned by distances to the selected feature lines.

One issue that affects classification accuracy in generalized dissimilarity spaces spanned by feature lines is the interpolation inaccuracy [31]. The interpolation and the extrapolation inaccuracies were detected when a rectified version of the nearest feature line (NFL) classifier was proposed [60]. The interpolation inaccuracy appears when lines are constructed in a class that is divided into clusters and when between the clusters there are instances of a different class as it can be seen in Fig. 4.1; where a point p belonging to the class "triangle" is closer to a feature line that links the two clusters of the class "square." The extrapolation inaccuracy appears when the extrapolating part of a line trespasses the territory of a different class as it is shown in Fig. 4.2; where a point p that belongs to the class "triangle" is closer to the extrapolating part of a feature line of the class "square."

In [31], the authors demonstrate the applicability of the generalized approach to correlated

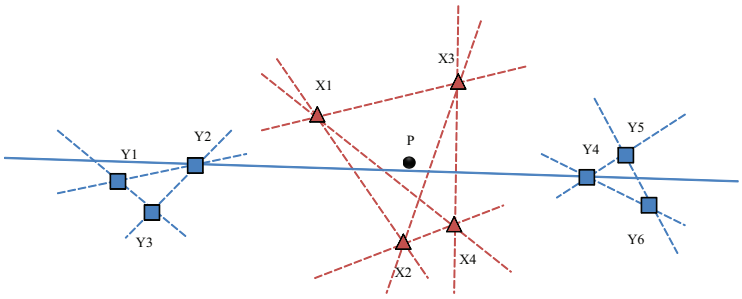


Figure 4.1: Interpolation inaccuracy

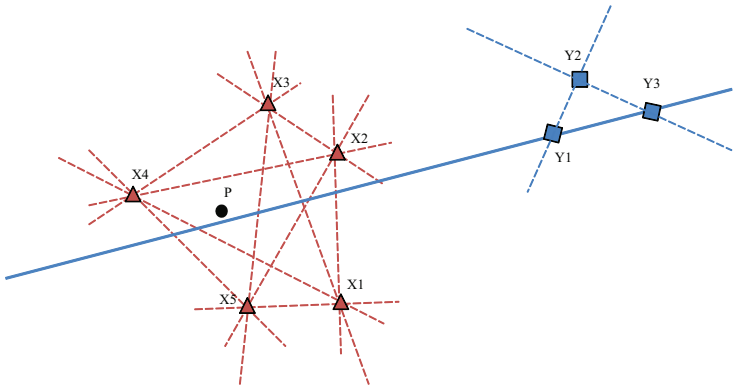


Figure 4.2: Extrapolation inaccuracy

data sets because the procedure exploits the linear geometric relations in the data; in addition, they show that selecting few of the largest lines is sufficient to represent all the objects accurately. Also, the study in [61] points out that the selection of the middle length feature lines is suitable for manifold structures. A representation set composed by feature lines should avoid the inaccuracies related to the computation of distances between objects and feature lines in order to have a good representational power. In this paper, we propose a new criterion for selecting feature lines that tries to avoid these inaccuracies. It is based on selecting the lines that minimize the NFL [62, 63] classification error in the training set. As this classifier is affected by these inaccuracies, by minimizing its error we are selecting the lines that are likely to avoid these problems. This allows taking advantage of the power of generalizations by feature lines not only for correlated data sets but also for data with different distributions such as spherical ones.

The paper is divided as follows. Subsection 4.1.2 introduces the dissimilarity space and the generalized dissimilarity space by feature lines. Subsection 4.1.5 presents the proposed criterion.

Subsection 4.1.6 presents the data and experimental setup, while subsection 4.1.7 presents the results and discussion. Conclusions are drawn in Subsection 4.1.8.

4.1.2 Dissimilarity representations

In this section we introduce some concepts such as dissimilarity spaces and generalized dissimilarity spaces.

4.1.3 Dissimilarity space

The dissimilarity space (DS) was proposed by Pekalska and Duin [3]. In a pattern recognition problem, objects are usually represented by feature vectors. These vectors can be embedded in feature vector spaces that are Euclidean or metric. Alternatively, objects can be represented by a vector of dissimilarities with other objects, and these vectors can be mapped to a DS. This space was postulated as a Euclidean one, providing the possibility of using well-known statistical classifiers such as linear and quadratic ones. Given a training set T and a representation set $R = \{r_1, r_2, ..., r_k\}$, that is usually a subset of T, any object x in T can be represented by a vector of dissimilarities with the objects in R:

$$d_x = [d(x, r_1)\ d(x, r_2)\ ...\ d(x, r_k)], \tag{4.1}$$

where $d(x, r_1), ..., d(x, r_k)$ are dissimilarities from the object x to the prototypes.

Classifiers can be trained in the DS using T. For the classification of a new test object, the representation is obtained in the same way as for training objects (see Eq. 4.1). The dimension of the DS as well as the length of the dissimilarity vectors are determined by the cardinality of the representation set $|R|$: the amount of prototypes selected. Prototype selection allows users to decide their preferred trade-off between classification accuracy and computational efficiency. There are methods that even return the "best" number of prototypes automatically such as the LinProg and EdiCon procedures proposed in [11].

4.1.4 Generalized dissimilarity space by feature lines

The generalized dissimilarity space (GDS) was proposed to cope with small sample size problems since the data is enhanced by the creation of feature lines. Analogous to the DS, the GDS by feature lines is a space where objects are represented by their dissimilarities with the feature lines [31]. In this approach, the feature lines are the prototypes. Feature lines are computed between objects of the same class. The dimension of the GDS and the length of the dissimilarity vectors are determined by the amount of feature lines used. We denote the representation set of feature lines as $R_L = \{L_1, L_2, ..., L_k\}$, where R stands for a representation set and L stands for feature lines. Let us suppose we have a training set of C classes with N objects per class, then the total number of feature lines that can be generated is: $n = N * (N-1)/2 * C$. The representation of any object x is:

$$d_x = [d(x, L_1)\ d(x, L_2)\ ...\ d(x, L_k)], \tag{4.2}$$

where $d(x, L_1), ..., d(x, L_k)$ are dissimilarities from the object x to those feature lines.

The usual assumption when using DR, that is also made in this paper, is that the set of objects is given only in terms of pairwise dissimilarities. This means that we do not have a feature space where the lines between two feature vectors can be easily generated from the position of an object to the position of another object. Instead, the feature lines must be computed in terms of the available set of dissimilarities that contain the information about the nearness between the objects. This can be done following the methodology in [31]. Given the dissimilarity matrix $D(T, R)$ and in order to build the generalized dissimilarity matrix $D(T, R_L)$,

it is needed to compute the height h of a scalene triangle as shown in Fig. 4.3, where x_i^c and x_j^c are two arbitrary objects of the same class, x_k is a new object and d_{ik}, d_{jk}, and d_{ij} are the dissimilarities between the three objects. In this approach, the authors decided to restrict themselves to metric dissimilarities to avoid complex numbers when solving the equations to find h. Let us define:

$$s = (d_{ik} + d_{jk} + d_{ij})/2, \tag{4.3}$$

then, the area of the triangle can be computed by:

$$A = \sqrt{s(s - d_{ik})(s - d_{jk})(s - d_{ij})}. \tag{4.4}$$

As the area can also be computed by:

$$A = \frac{d_{ij} \times h}{2}, \tag{4.5}$$

Eqs. (4.6) and (4.5) can be solved for h that coincides with the distance of x_k with the feature line:

$$h = \frac{2 \times \sqrt{s(s - d_{ik})(s - d_{jk})(s - d_{ij})}}{d_{ij}}. \tag{4.6}$$

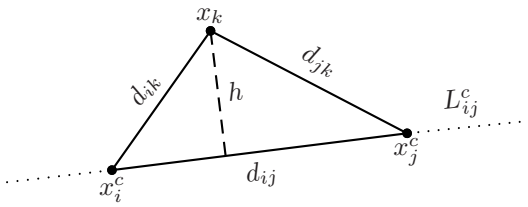


Figure 4.3: Scalene triangle where height is computed to find dissimilarity with a line for data without an associated feature representation available

4.1.5 Proposed criterion

The motivation in our proposal for the selection of feature lines is to minimize the interpolation problems that limit the applicability of GDRs by feature lines. Our main objective is to select the best small set of feature lines trying to affect as least as possible the classification accuracy in GDS while the computational and storage costs are decreased. The set of feature lines selected will be used to generate the GDS, since training and test objects will be represented by their dissimilarities with the feature lines. We will try to achieve this by a new criterion to perform the selection. This will contribute to diminish computational costs of classification by computing less distances with feature lines of an incoming object, and by using less dimensions to train a classifier in a GDS. Selection criteria for feature lines have been based only on geometric properties such as the length of segments between the points defining the feature lines [61].

For the classification of objects using feature lines, the first classifier proposed was the NFL [62, 63]. When a new object x is submitted to classification for the NFL classifier, it is found the feature line $L_{\hat{i}\hat{j}}^{\hat{c}}$ whose distance to the new object is minimum, then, the classification rule assigns to the object x the class $\hat{c}$ to which this nearest feature line belongs. The NFL distance is defined therefore as:

$$d(x, L_{\hat{i}\hat{j}}^{\hat{c}}) = \min_{L_{ij}^c \in R_L, 1 \leq c \leq C, 1 \leq i,j \leq N_{i \neq j}} d(x, L_{ij}^c), \tag{4.7}$$

in which C is the number of classes and N is the number of objects per class.

We propose a new criterion for feature line selection for GDS creation based on the minimization of the NFL error in the training set. The NFL classifier is very sensitive to the interpolation inaccuracies, so the set of prototype feature lines that ensure a smaller error of the NFL are likely to carry less interpolation inaccuracies when they are used for constructing the generalized dissimilarity space. This criterion should be more robust than the length based criteria that do not try to find discriminative feature lines or to minimize the interpolation problems present in a GDS created by feature lines. We use the forward selection (FS) to optimize the criterion. Having a training set composed by feature lines, the FS starts from an empty set and sequentially adds the line that, together with the already selected lines, ensures the best classification result for the NFL on the training set. The proposed criterion is formulated in Eq. 4.8:

$$\min_{R_L} : \jmath(T, R_L) = \sum_{x \in T} CE(x),$$

$$CE = \begin{cases} 1, & \lambda_T(x) \neq \lambda_{R_L}(L) \\ 0, & \lambda_T(x) = \lambda_{R_L}(L) \end{cases}, L = \operatorname{argmin} d(x, L_{i\hat{j}}^{\hat{c}}), L_{i\hat{j}}^{\hat{c}} \in R_L \tag{4.8}$$

in which $\lambda_T(x)$ is the class label of $x \in T$ and $\lambda_{R_L}(L)$ is the class label of the L that satisfies $\min d(x, L_{i\hat{j}}^{\hat{c}})$. CE is the classification error; it is 1 if the label of the object and its nearest feature line do not match and 0 otherwise. $\jmath$ is thereby the NFL classification error of the training set T classified by the set of feature lines R_L.

4.1.6 Datasets and experimental setup

We conducted experiments on eight different data sets: the Difficult data set generated by gendatd in PRTools [64], the Oral [65] and the Laryngeal [66] data sets. From the UCI Machine Learning Repository [56] we used the Liver, the Heart, the Diabetes and the Sonar data sets. In the Heart data set, objects with missing values were neglected, and only the first thirteen attributes were used since they are the most discriminative as agreed by experts. A data set of volcanic events from the Galeras volcano in Colombia was also used.

Difficult normally distributed data set. This is an artificial two-class problem created by the gendatd command in PRTools [64] where the classes have different class variances and class overlap.

Oral data set. The Oral data set consists of autofluorescence spectra acquired from healthy and diseased mucosa in the oral cavity. The data was collected at the Department of Oral and Maxillofacial Surgery of the University Hospital of Groningen [65]. The classes represent healthy tissue and diseased tissue.

Laryngeal data set. The Laryngeal data set is from the Bulgarian Academy of Sciences [66]. The goal is the diagnosis of laryngeal pathology, and especially in detecting its early stages. Classes are normal and pathological voices; they are described by 16 parameters in the time, spectral and cepstral domains.

Liver data set. The liver data was created by BUPA Medical Research Ltd. It is available at UCI Machine Learning Repository [56]. It contains information about parameters that are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. Classes are normal or liver disorder. The first features are blood tests that are thought to be sensitive to liver disorders and the last feature is the number of half-pint equivalents of alcoholic beverages drunk per day.

Diabetes data set. It comes from the National Institute of Diabetes and Digestive and

Kidney Diseases (USA). It is available at UCI Machine Learning Repository [56]. One class is constituted by healthy individuals and the other by individuals with a higher risk of diabetes. All patients here are females of at least 21 years old of Pima Indian heritage. Attributes are: number of times pregnant, plasma glucose concentration a 2-hours in an oral glucose tolerance test, diastolic blood pressure, triceps skin fold thickness, 2-hour serum insulin, body mass index, age, diabetes pedigree function and if the diabetes test was positive or not.

Heart data set. Heart disease diagnosis data set refers to the presence of heart disease in the patient. Classes are normal and sick. It is available at UCI Machine Learning Repository [56]. The data was collected by:

- Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
- University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
- University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
- V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D

Features are age, sex, chest pain type, resting blood pressure, serum cholesterol, resting electrocardiographic results, maximum heart rate achieved among others.

Sonar data set. In this data, the objects are sonar signals, bounced off a metal cylinder and a roughly cylindrical rock. Classes are mines (metal cylinder) or rock. The data set was developed by Terry Sejnowski, at the Salk Institute and the University of California at San Diego, and R. Paul Gorman of Allied-Signal Aerospace Technology Center. It contains signals obtained from a variety of different aspect angles, spanning 90 degrees for the cylinder and 180 degrees for the rock. It is available at UCI Machine Learning Repository [56].

Galeras data set. Galeras is an active volcano located in the southwest Colombian Andes near the border of Ecuador. Its elevation is 4.276 m (14,029 feet). The volcano has erupted more than 20 times since it was first visited by European explorers in the 16th century. It is monitored by The Volcanological and Seismological Observatory in Pasto (VSOP), that has a monitoring network composed by 9 stations: 7 short-period stations and 2 broadband stations. One of the short-period stations located in a place called *Anganoy* (ANGV) at a distance of 0.9Km SE from the main active crater, is considered a reference by the VSOP staff. Signals used in this paper were recorded at the ANGV station. The classes of volcanic events used in our experiments are: volcano tectonic (VT) events, long period (LP) events, and tremors (TR). Data is composed by normalized power spectral densities (PSDs) of length 128 estimated via Yule-Walker's method, using an autoregressive model of order 6.

Table 4.1 shows the properties of these data sets in their original format. Subsequently, for applying the feature lines approach, we move from original feature representations to dissimilarity representations by computing pairwise distances between all the objects in the data set. The distances to the feature lines are computed using the objects pairwise distances as it is explained in Subsection 4.1.4. We are restricted to metric distances as it is needed to avoid complex numbers in the computation of distances from objects to the feature lines. For the first data sets, the DR was constructed from the Euclidean distances between the feature vectors but for the Galeras data set it was used the L1 distance. Using the matrices of dissimilarities between the objects, we can construct a dissimilarity representation by feature lines.

For each dissimilarity data set, fifteen objects were randomly selected for training from each class; thirty in total for two-class problems and fortyfive for the Galeras data that is a three-class problem. The remaining objects were used for testing. The total number of objects per class for each data can be found in Table 4.1. The nearest neighbour (1-NN) classifier assigns to a test object the class of the nearest object between those thirty or fortyfive selected; the Euclidean

Table 4.1: Description of the original form of the data sets used in the experiments and training set ($|Train|$) and test set ($|Test|$) sizes used for classification in GDS

Data	# Classes	# Objects per class	Dimension in feature space	$\|Train\|$ in GDS	$\|Test\|$ in GDS
Difficult	2	100/100	2	30	170
Oral	2	581/123	199	30	674
Laryngeal	2	81/132	16	30	183
Heart	2	160/137	13	30	267
Liver	2	145/200	6	30	315
Diabetes	2	500/268	8	30	738
Sonar	2	97/111	60	30	178
Galeras	3	150/294/590	128	45	989

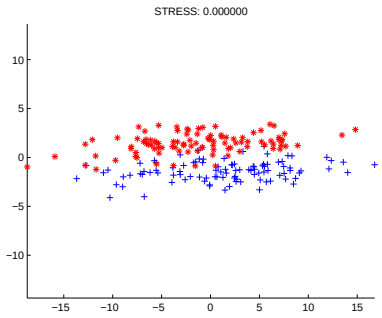
distance was used to compare the pairs of objects in order to find the prototype object with the smallest distance to the test object. In the case of generalized representations for data sets with two classes, 210 feature lines were constructed in total from the thirty objects, $\frac{15*14}{2} = 105$ for each class; in the Galeras data 315 feature lines were constructed. The NFL classifier uses all these feature lines as training set and assigns to a new object the class of the feature line that had the smallest distance to the test object. The 210 and 315 feature lines are used respectively as potential candidates for the construction of the GDS.

For the creation of the GDS, the selection methods search the best 2, 5, 10, 15 and 20 feature lines among the available candidate feature lines according to each criterion. The linear discriminant analysis (LDA) classifier was trained using all the original randomly selected fifteen objects per class in each of the GDSs generated by the different sets of 2, 5, 10, 15, and 20 feature lines for the different selectors. The errors are computed in the test data. In each configuration, training and test objects projected in the GDS were represented by dissimilarity vectors of length equal to the amount of feature lines considered. When two feature lines are considered for example, we obtain a space of two dimensions and the dissimilarity vectors representing training and test objects contain two values. These values encode the dissimilarities of the object with the feature lines that generated the space. We decided to use the classification error as a measure to compare the performance of the proposed feature line selection method with the other selection methods. We compared the LDA classification errors in the GDS using our proposed method, the FS minimizing the NFL error on the training set (LDA+FS+NFL error in GDS), with the following feature lines selection methods as reference: random (LDA+random in GDS), selection of the shortest feature lines (LDA+shortest lines in GDS), and selection of the largest feature lines (LDA+largest lines in GDS).

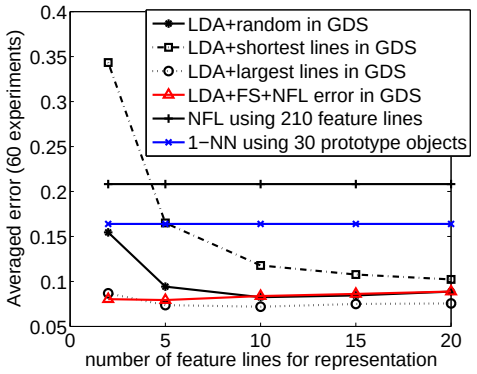
For the cases of selection by largest length, first, the feature line with largest length of one class is selected, subsequently, the line with the largest length from another class and when we have one line from each class we start again from the first class, in a way that it is ensured that all the classes are evenly represented. We also included in the comparison, as reference, the 1-NN classifier in the original data without generalization using the distances to the randomly selected training objects; and the NFL in the matrix of distances to the total set of feature lines. The process of splitting the data into training and testing sets was repeated 60 times. Average errors were computed and the curves of error rates for varying number of feature lines are shown in Figs. 4 to 8. This implies that the lower a curve or a line is, the better the performance of the related method is.

As regularization parameters for the LDA, we used 0.01 for the two parameters respectively as suggested in [3]. It is important to take into account that 210 distance computations are made with the feature lines when classifying test objects with the NFL; when classifying the

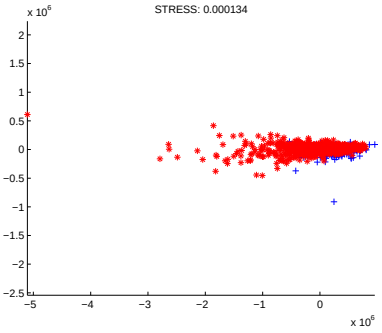
data in the GDS using the LDA and the selection methods, only the dissimilarities with the small sets of 2, 5, 10, 15 and 20 feature lines are measured. Figures 4 to 8 also show data distribution for each problem by a mapping of distances using a multidimensional scaling (MDS) [67].



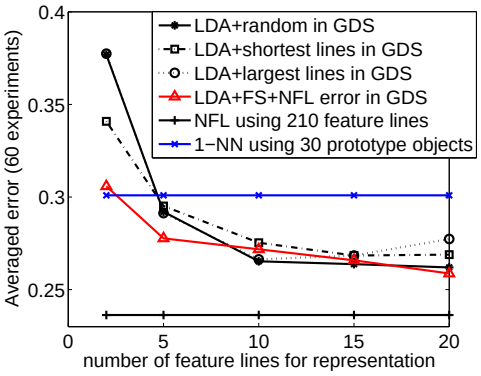
(a) 2D MDS projection of Difficult data set



(b) Classification errors in the Difficult data set



(c) 2D MDS projection of Oral data set

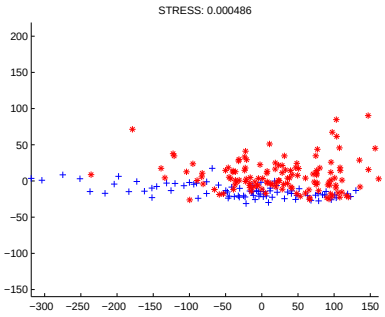


(d) Classification errors in the Oral data set

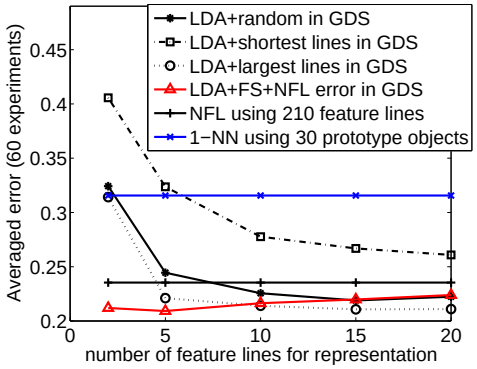
Figure 4.4: Data distributions in (a) and (c), and fraction of classification errors vs number of prototypes for the LDA in dissimilarity spaces generated by feature lines, and for 1-NN and NFL using the distances directly for different data sets ((b) and (d)). The proposed approach is the one in solid red curves with triangles.

Table 4.2: Percentage of cumulative variance retained by the principal components

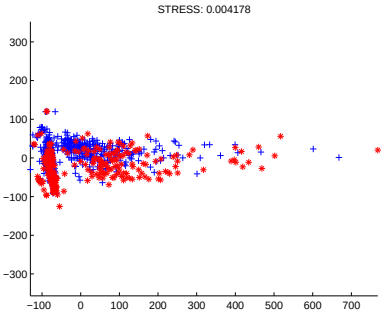
# Comp	Diffic.	Oral	Laryng.	Heart	Liver	Diab.	Sonar	Gal.
1	93	97	93	74	71	88	31	39
2	100	98	97	89	85	95	52	67
3	-	99	98	98	97	97	60	70
4	-	99	99	99	98	98	67	73
5	-	99	99	99	99	99	72	76



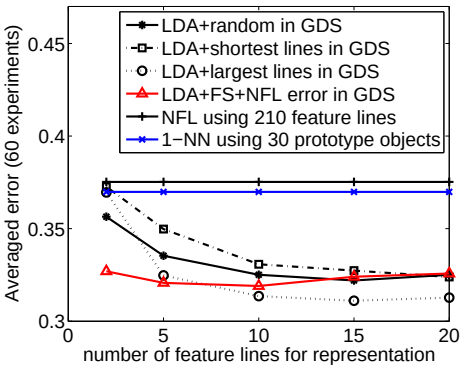
(a) 2D MDS projection of Laryngeal data set



(b) Classification errors in the Laryngeal data set



(c) 2D MDS projection of Diabetes data set



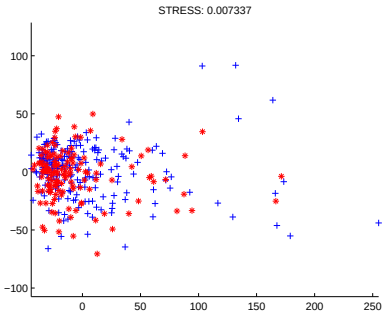
(d) Classification errors in the Diabetes data set

Figure 4.5: Data distributions in (a) and (c), and fraction of classification errors vs number of prototypes for the LDA in dissimilarity spaces generated by feature lines, and for 1-NN and NFL using the distances directly for different data sets ((b) and (d)). The proposed approach is the one in solid red curves with triangles.

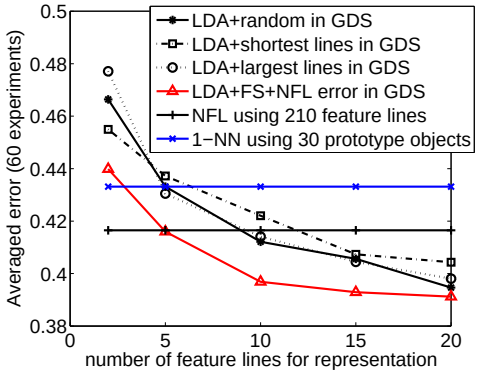
4.1.7 Results and discussion

Table 4.2 shows the cumulative variances retained by the principal components of each data found by Principal Component Analysis (PCA) in the original feature space. It can be seen that the Difficult, Laryngeal and Oral data sets are the ones with highest variance retained in the first component, so they are elongated data sets.

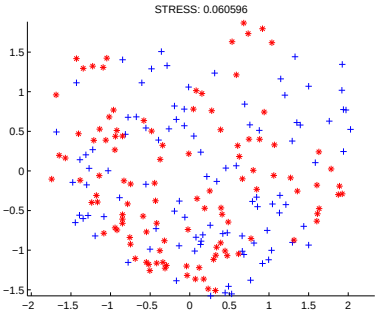
In the results for the Difficult data set in Fig. 4 (b), it can be seen that the proposed selection method behaves similar to the largest lines selector, which is the best performing method. This is to be expected since previous works concluded that the largest lines were very good for representing elongated data. In the results for the Oral and Laryngeal data sets in Figs. 4 (d) and 5 (b), it can be seen that the *FS+NFL error* outperforms all the feature lines selectors for very small numbers of feature lines. For more than five feature lines, the *FS+NFL error* method performs similar to the other selection methods. In the Diabetes data set in Fig. 5 (d), a similar behaviour is observed. In Table 4.2, it can be seen that the principal component of this data retains the 88 percent of the data variance. From this and from the MDS plot in Fig. 6 (c), we



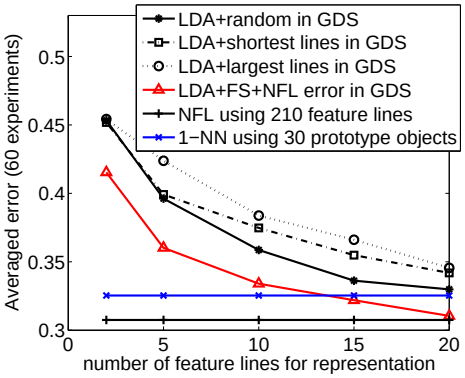
(a) 2D MDS projection of Liver data set



(b) Classification errors in the Liver data set



(c) 2D MDS projection of Sonar data set

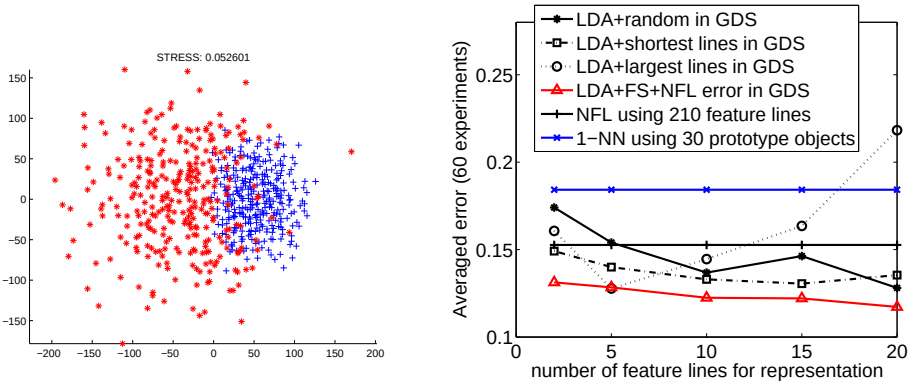


(d) Classification errors in the Sonar data set

Figure 4.6: Data distributions in (a) and (c), and fraction of classification errors vs number of prototypes for the LDA in dissimilarity spaces generated by feature lines, and for 1-NN and NFL using the distances directly for different data sets ((b) and (d)). The proposed approach is the one in solid red curves with triangles.

can see that this data is also elongated. In general, for elongated data, the proposed *FS+NFL error* selector outperforms all the other methods for very small numbers of feature lines such as two. The proposed approach performs similar to the selection of the largest lines for more than five prototypes.

In three of the four elongated data sets, the LDA classifier in the GDS generated by the feature lines obtains lower classification errors than those obtained by the NFL using distances to feature lines directly and the 1-NN using distances to the original training objects. In these data sets, five feature lines provide a good trade-off between classification accuracy and computational cost. In the Difficult and Laryngeal data, increasing the number of feature lines for prototypes when using the proposed criterion only leads to the same or worst results since the intrinsic dimension of the data is very low. This can be deduced by the cumulative variances, thereby adding more dimensions can lead to either the same or noisy results. Other causes of the worst results are the sensitivity of the FS to local optima, and sometimes the chosen parameters for the regularization of the LDA may not be sufficiently good. In the Laryngeal data set, the best overall result was obtained by the proposed selection criterion using five prototypes.



(a) 2D MDS projection of artificial Spherical data set

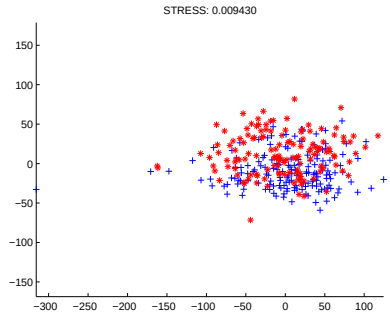
(b) Classification errors in the artificial Spherical data set

Figure 4.7: Data distributions in (a), and fraction of classification errors vs number of prototypes for the LDA in dissimilarity spaces generated by feature lines, and for 1-NN and NFL using the distances directly for artificial Spherical Gaussian data (b). The proposed approach is the one in solid red curves with triangles.

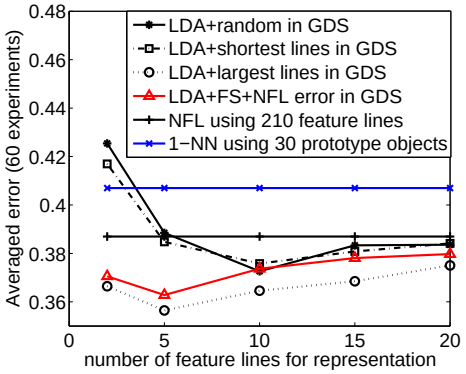
From Table 4.1 we can see that the Oral, Laryngeal and Diabetes data have 199, 16 and 8 features respectively but their intrinsic dimension is much lower as demonstrated by the elongated distribution and the high amount of variance contained in the principal components. The number of features in the feature space does not influence the number of feature lines to select as prototypes. However, for approximately linear data, the number of feature lines to select can be roughly deduced from the number of principal components that retain, for example, the 95 percent of the data variance.

In the Liver and Sonar data sets in Fig. 6 (b) and (d) that have a more spherical distribution, the proposed feature line selector outperforms the other selectors for all the numbers of feature lines. In these cases, we do not observe the same phenomenon as in the elongated data where adding more feature lines sometimes decreased the classification performance. For increasing number of feature lines, the method is always improving and this seems to be happening because of the spherical distribution having the data variances contained in more directions and not only in a few dominating ones. This implies that the intrinsic dimension of the data is larger, which can be confirmed from Table 4.2.

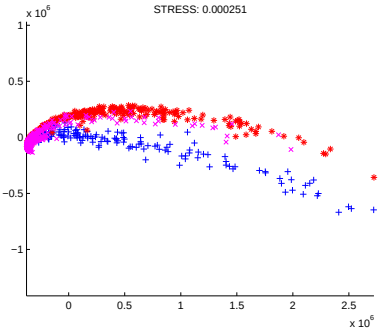
In the Sonar data, the largest lines selection is the worst performing method. Largest feature lines are not suitable to represent this spherical data since they do not provide discriminative information for classification. Also, the largest distances between the objects may be shrinked so errors are introduced. For example, for each feature line all the objects that are nearest to the feature line will have a similar small coordinate in the dimension determined by the feature line in a dissimilarity space. The largest lines are created by opposite objects in the border of the data; they pass nearby the data center and have a segment length similar to the sphere diameter. If the largest lines are selected, pairs of objects situated in the border of the data distribution in opposite sides and closer to the same largest line will have a very similar small coordinate in the dimension of the DS corresponding to that line. This does not represent reality since, in fact, they are very far from each other with a distance between them similar to the diameter of the sphere. This will lead to misclassifications. The proposed criterion provides more discriminative information for classification since it finds the most discriminative lines by optimizing a classification error in the training set.



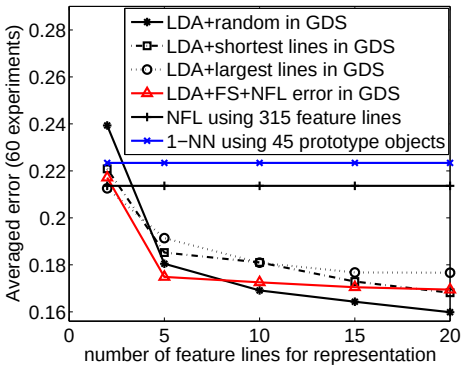
(a) 2D MDS projection of Heart data set



(b) Classification errors in the Heart data set



(c) 2D MDS projection of Galeras data set



(d) Classification errors in the Galeras data set

Figure 4.8: Data distributions in (a) and (c), and fraction of classification errors vs number of prototypes for the LDA in dissimilarity spaces generated by feature lines, and for 1-NN and NFL using the distances directly for different data sets ((b) and (d)). The proposed approach is the one in solid red curves with triangles.

To illustrate the advantages of the proposed criterion over the largest lines selection for spherical data, in a controlled experiment we generated an artificial data set with two partially overlapped spherical classes using the PRTools [64] routine *gendatc*. Each class is drawn from Gaussian distributions having different class mean and variances. The data was generated with 300 objects per class and 40 features per object. The classification results on this data can be seen in Fig. 7 together with the 2D plot of the data using MDS. Experimental setup is similar to the one used for the real data sets except that the experiments were repeated 10 times. From the results, we can see how while increasing the number of feature lines the performance of the largest lines selection remarkably deteriorates and the performance of the *FS+NFL error* improves. While more largest feature lines are added, the data representation looses more discrimination ability since more largest distances in different directions are shrinked. The interpolation problems are also influencing the results as the classes are overlapped. The proposed method also outperformed the shortest lines selection and the random selection.

In the Liver data set there are some outliers, so the largest lines are probably constructed using these objects as they are far from the other data. This may also be one of the causes of

the better performance of the proposed criterion over the largest lines. For data with outliers, the *FS+NFL error* may be more suitable as it must discard the lines constructed with outliers that lead to higher classification error in the training set.

In the Heart data set in Fig. 8 (b), the *FS+NFL error* selection works better than the shortest lines selection and the random selection for two and five feature lines, but the method is outperformed by the largest lines selector. This data is slightly elongated and has a high class overlap; for this reason, this result can be expected. Regularization parameters of the LDA may not be good as all the selection methods decrease performance for increasing number of feature lines.

In the Galeras volcano data set in Fig. 8, all the selection methods perform about the same. The method using the proposed criterion outperforms the other methods for five prototypes, but this improvement is not sustained for different numbers of feature lines. An interesting issue is that the LDA classifier in the GDS outperforms the 1-NN and the NFL using only five feature lines for representation. Other selection methods should be studied for this data since the ones compared perform very similar. This data presents a nonlinear structure (see Fig. 8(c)) and is also the first data used in our experiments with more than two classes.

Summarizing, the results suggest that the proposed selector (FS minimizing the NFL error on the training data) shows a competitive performance for the data used in our experiments. The proposed selection method seems to be a suitable alternative to represent data with spherical distributions where there are no dominating directions and data with outliers.

The robustness of the proposed method was studied in comparison with the best reference

Table 4.3: Minimum and Maximum standard deviations achieved by the methods in sixty experiments over different numbers of prototypes

Data	Proposal	Largest lines selector
Difficult	0.01-0.04	0.02-0.09
Laryngeal	0.03-0.04	0.04-0.07
Oral	0.06-0.07	0.07-0.13
Diabetes	0.03-0.04	0.04-0.07
Heart	0.03-0.06	0.03-0.06
Liver	0.04-0.05	0.04-0.05
Sonar	0.05-0.06	0.05-0.06
Galeras	0.01-0.02	0.01-0.02

method to perform the selection,that is the largest feature lines selector. Standard deviations were analyzed for the two methods, they are shown in Table 4.3. Standard deviations of the proposed method were smaller than or similar to the ones of the reference method. The proposed approach was more robust than the reference approach in four elongated data sets, and as robust as the reference approach in the other data sets.

The number of times that the proposed approach was superior than the best reference one was counted for the different data partitions and the different numbers of prototypes in the sixty experiments. Equal performances were not counted. It was found that for the data set with outliers (Liver), spherical data set (Sonar), and for the smallest number of prototype feature lines used in the elongated data sets (Difficult, Laryngeal, Diabetes, Oral), the proposed approach finds more times a better result compared to the reference approach. A remarkable difference is found for the spherical data set where the proposed method finds a superior result 229 times, whereas the reference method finds a better result only 62 times. This supports our findings that in these situations our approach is recommended. In the Heart data set that has a high class overlap, and for larger prototype sets in the elongated data sets, the reference approach finds a preferable solution more times than the proposed approach. This can be im-

proved to some extent if a more advanced optimization method is used to optimize the proposed criterion instead of the FS, because this method falls into local optima especially when a high class overlap is present.

Table 4.4: Execution times in seconds for Liver data set

number of prototype feature lines	2	5	10	15	20
Feature line selectors					
FS+NFL error in GDS	0.03	0.06	0.14	0.20	0.27
largest lines in GDS	0.02	0.00	0.00	0.00	0.00
smallest lines in GDS	0.00	0.00	0.00	0.00	0.00
random selection in GDS	0.00	0.00	0.00	0.00	0.00
Classifiers					
LDA+proposal(FS+NFL error)in GDS	0.03	0.04	0.03	0.03	0.04
NFL classifier(210 Feat. lines)	0.30				
1-NN classifier(30 prototypes)	0.10				

In Table 4.4, we present the execution times of both the selection and the classification methods (not training times, but classification times for all the test objects) for the Liver data set in a computer with an Intel Pentium Dual CPU 1.80 GHz processor, 2 GB RAM and Matlab version 7.9.1.705. We decided to use only one data set to show these results since for the other data the same phenomenon is observed. We only present the LDA with our selection method since for the other selection methods the time is similar. This happens because once a fixed amount of feature lines is selected, the selection method used does not influence the execution times of the classifier. From the execution times it can be seen that, when the selection is performed by the proposed criterion before classification for the LDA in the GDS, the classification times are smaller compared to the 1-NN and NFL. It should be pointed out for a fair comparison that the proposed selection method is more time consuming than the other selectors. Nevertheless, as this process is usually executed off-line and the computational complexity of the optimization using the FS is polynomial ($O(n^2)$) and not exponential, this usually does not represent a significant drawback.

In general, GDRs by feature lines are useful for correlated data. Our selection approach handles well this type of correlated or elongated data distribution, but it is more useful for very small sets of prototype feature lines where the selection of the largest ones may fail. For larger sets, the FS falls into local optima. However, if the proposed selection criterion (minimization of the NFL classification error in the training set) is optimized by a more advanced procedure instead of the FS, the method can be useful for larger sets as well. Examples of correlated data are high resolution images or histograms because close pixels and bins are usually correlated. From a PCA analysis of the data set, it can be deduced that it is correlated if the number of principal components needed to retain the 95 percent of the data variance is substantially smaller than the number of features. The proposed method can be useful or more appropriate than the other selection methods used as reference for the experiments in the following cases:

- data sets with low correlations such as in the case of low resolution images or noisy measurements (this can be seen from a PCA analysis, where there are no clearly dominant eigenvalues, each of them accounts for about the same amount of variance retained)
- data sets with a high intrinsic dimension (this can be seen from a PCA analysis, where to retain the 95 percent of the variance almost all the principal components are needed), and especially for spherical distributions (this can be seen from a 2D plot of the data)

- data sets with outliers (this can be seen from a 2D plot of the data, or from an analysis of distances between the objects)
- small sample sizes (number of objects for training much smaller than the number of features or prototypes in a dissimilarity space), because a really careful selection is needed to handle the fact that the data distribution may not be well sampled by such a small amount of training objects

However, if the data set has a very high class overlap (this can be deduced from a high 1-NN classification error), our proposed criterion may work well, but its optimization must be made by a more advanced procedure such as a genetic algorithm because the FS falls into local optima for complicated data. Also, the study of the performance of the proposed method for a high number of classes is not developed yet, so its application in these cases cannot be recommended.

4.1.8 Conclusions

It can be concluded from the studied datasets that the proposed feature lines selection approach minimizing the nearest feature line (NFL) classification error on the training data provides competitive or better results than the other selection methods studied for the construction of generalized dissimilarity spaces (GDSs). The method allows one to improve classification rates in the GDS and, at the same time, to decrease computational costs by selecting very small sets of feature lines. The proposed approach was the best method for data sets with spherical distributions (e.g. Gaussian distributed) and higher intrinsic dimension. It is also suitable for data with outliers. For elongated data, the proposed approach is useful for very small numbers of feature lines. For a very high class overlap, the proposed method is not recommended. These conclusions are restricted to the studied two-class problems. For multi-class problems, some of these statements may not hold. For data with a manifold structure, all the selection methods compared work similar. In many of the data sets used in our experiments, classifiers in GDS show improvements in terms of accuracies and computational costs over the nearest neighbour (1-NN) classifier as well as over the NFL classifier applied to the original data.
Further studies should be developed in the selection of feature lines for nonlinear structures. A topic of interest for future research is the creation of suitable classifiers for the GDS, that take into account how the dissimilarity vectors were created and how to deal with the “fake” distances introduced in the training data by the interpolation inaccuracies.

Acknowledgments

We acknowledge financial support from the FET programme within the EU FP7, under the project ”Similarity-based Pattern Analysis and Recognition - SIMBAD” (contract 213250) as well as the project ”Cálculo científico para caracterización e identificación en problemas dinámicos” (code Hermes-10722) granted by Universidad Nacional de Colombia.

4.2 Towards cluster-based prototype sets for dissimilarity space classification

This section has been published as "Towards cluster-based prototype sets for dissimilarity space classification", by Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Edel Garcia-Reyes and Robert P. W. Duin, in *Proceedings of the 18th Iberoamerican Congress on Pattern Recognition, CIARP 2013, LNCS*.

Abstract

The selection of prototypes for the dissimilarity space is a key aspect to overcome problems related to the curse of dimensionality and computational burden. How to properly define and select the prototypes is still an open issue. In this paper, we propose the selection of clusters as prototypes to create low-dimensional spaces. Experimental results show that the proposed approach is useful in the problems presented. Especially, the use of the minimum distances to clusters for representation provides good results.

4.2.1 Introduction

The representation of objects is crucial for the success of a pattern recognition system. The feature space representation is the most common approach since a large number of techniques can be used. Dissimilarity representations [3] arose as an alternative and have been showing a good performance in several problems, where the dissimilarities may be computed by directly matching original objects [3] or on top of feature representations [17]. Three main approaches are presented in [3], the most promising being the dissimilarity space (DS).

In the DS, an object is represented by a vector of dissimilarities with other objects called prototypes. If a large set of prototypes is used, it leads to a high-dimensionality of the DS implying that computational costs of classification are increased as well as storage costs. In addition, a high dimensionality leads to problems related to the "curse of dimensionality" and small sample sizes. Furthermore, high dimensional representations are likely to contain noise since the intrinsic dimensionality of the data is usually small, leading to overfitting.

Prototype selection is a way to overcome these drawbacks. It has been studied [11] for reducing dimensions of DS with encouraging results. Several methods have been proposed such as Kcentres, Forward Selection (FS), Editing and Condensing, among others [11]. In these studies, the selected prototypes are objects. However, some efforts are also put in a different direction and, instead of objects, linear models are built, selecting out some of them for representation [22]. These studies showed that it is a feasible alternative to use a small number of carefully selected feature lines as prototypes instead of the original objects.

In this paper we study the selection of clusters for the generation of a low-dimensional generalized dissimilarity space (GDS). Our hypothesis is that clusters may be useful to obtain low-dimensional GDSs in case datasets are structured in clusters. A similar approach was presented in [68], however it was specifically developed for graph distances while our research is not restricted to graphs. Besides, they do not take into account the selection of the best clusters, while our goal is to find the clusters which allow a good classification with a decreased dimension of the space. We also included the subspace distance to clusters. Different approaches to compute the distances of the training and test objects to the clusters are presented. The paper is divided as follows. Section 4.2.2 introduces the DS and prototype selection. Section 4.2.4 describes the construction of the datasets based on cluster distances. Experimental results and discussions are provided in Sec. 4.2.5 followed by concluding remarks in Sec. 4.2.8.

4.2.2 Dissimilarity space

The DS was conceived with the purpose to address classification of data represented by dissimilarities that may be non-Euclidean or even non-metric. The dissimilarities of a training set X with a set of prototypes $R = \{r_1, ..., r_k\}$ are interpreted as coordinates in the DS. Thereby, the number of prototypes selected determines the dimension of the space. The DS was postulated as a Euclidean vector space, enabling the use of statistical classifiers. The set of prototypes may satisfy $R \subseteq X$ or $R \cap X = \emptyset$. Once R is selected by any prototype selector, the dissimilarities of

both training and test objects with R are computed. Let x be any training or test object and d a suitable dissimilarity measure for the problem at hand, the representation d_x of the object in the dissimilarity space is:

$$d_x = [d(x, r_1)\ d(x, r_2)\ ...\ d(x, r_k)]. \tag{4.9}$$

4.2.3 Prototype selection

Many approaches have been considered [11,17] for the selection of prototypes in the DS. Variants of wrapper or supervised methods [11] have been proposed. Other approaches are considered that use the distances or distribution of the prototypes over the dataset [17]; note that in these cases the class labels of the prototypes may not be needed. An interesting option is the genetic algorithm (GA) presented in [20]. The GA is an evolutionary method which uses heuristics in order to evolve an initial set of solutions (sets of prototypes) to better ones by using operations such as mutation and reproduction. Moreover, it is adequate to handle non-metric dissimilarities and it can find complicated relationships between the prototypes. For these reasons we propose to use a GA to select the clusters together with the leave-one-out nearest neighbor (LOO 1-NN) error in the DS as selection criterion. We adopt the same parameters for the GA as in [20]. Clusters present nice properties that good prototypes must have. For example, they do not provide redundant information since redundant or close objects must lie together in the same cluster and they cover the representation space better than a small set of objects.

4.2.4 Construction of models based on clusters

In this section we describe our methodology to construct the new dissimilarity datasets based on cluster distances computed from the originally given dissimilarities. In this study, the clusters are created per class by the Affinity Propagation algorithm [69]. In the clustering process representatives and their corresponding clusters emerge from a message-passing procedure between pairs of samples until stopping criteria are met. This method is reported to provide good clustering results. Furthermore, it is also of our convenience that it semi-automatically selects the proper number of clusters, emerging from the message-passing procedure but also from a user preference of the cluster representatives. The original dissimilarities must be transformed into similarities in order to apply the clustering procedure. We set the preferences for each object (i.e. the potential to be selected as cluster center) equal to the median similarity.

Different types of distances are used to measure the resemblance of objects with clusters such as: the minimum, maximum, average and subspace distances. The minimum distance is computed as the distance between the object and its nearest object in the cluster. The maximum distance is defined as the distance between the object and its farthest object in the cluster. The average distance is defined as the average of the distances between the object and all the cluster objects. The subspace distance is explained more carefully. Theory about it is sparse in the literature [70, 71], especially for the case of data given in terms of non-metric dissimilarities. Therefore, one contribution of this paper is to describe the methodology to compute the (speeded-up) distance of objects to subspaces when data is provided in terms of non-metric dissimilarities.

The methodology to compute the subspace distance to clusters is as follows. First, a subspace is created for every cluster in order to compute the subspace distances. To achieve this, the set of dissimilarities D inside a particular cluster is transformed into equivalent dot products (which can be interpreted as similarities) and centered according to the "double-centering" formula for each cluster:

$$S_{ij} = -\frac{1}{2}\left(D_{ij}^2 - \frac{1}{n}C_i - \frac{1}{n}C_j + \frac{1}{n^2}C_iC_j\right), \tag{4.10}$$

where D_{ij} is the dissimilarity between the cluster objects x_i and x_j, $C_i = \sum_j D_{ij}^2$, which is the i-th row sum of the dissimilarity matrix for the cluster objects, n is the number of objects in the cluster, and S_{ij} are the centered dot products. The eigendecomposition of S is performed and eigenvectors are sorted in descendent manner according to their eigenvalues. Only the eigenvectors associated with eigenvalues $\lambda > 0$ are used to compute the projections of new points to the subspace via the *Nyström* formula [72].

Each embedding coordinate of a cluster object x_i used to compute the dot product matrix or kernel is given by $e_{ik} = \sqrt{\lambda_k}v_{ik}$ as for multidimensional scaling (MDS) [70], where λ_k is the k-th eigenvalue and v_{ik} is the i-th element of the k-th eigenvector of S, but the embedding for a new point is obtained via the *Nyström* approximation which is interpreted as the Kernel PCA projection [71] using S as the kernel matrix. The *Nyström* formula was generalized for extending MDS as suggested in [71], therefore, each embedding coordinate e_{ik} is computed by:

$$e_{ik}(x) = \frac{\sqrt{\lambda_k}}{\lambda_k}\sum_{i=1}^{n} v_{ik}S(x, x_i), \tag{4.11}$$

where x_i are the objects belonging to the cluster and $S(x, x_i)$ is computed from a continuous version of the "double-centering" formula:

$$S(x, x_i) = -\frac{1}{2}\left(d(x, x_i)^2 - \frac{1}{n}\sum_j d(x, x_j)^2 - \frac{1}{n}\sum_j D_{ij}^2 + \frac{1}{n^2}\sum_{ij} D_{ij}^2\right). \tag{4.12}$$

$S(x, x_i)$ is a data-dependent kernel where $d(\cdot, \cdot)$ is the dissimilarity function. This *Nyström* embedding is applied to speed-up the embedding computation instead of recomputing the eigendecomposition including x in the whole process. However, in our case, the embedding is not directly used, instead, the embedding coordinates are used to compute the distance to the subspace L. The squared distance of an object to the subspace $d_L(x, L)^2$ is formulated as the difference between the squared length of the vector (its squared norm) given by $S(x, x)$ and the length of its projection on the space via *Nyström*:

$$d_L(x, L)^2 = S(x, x) - \sum_{k=1}^{m}\left(\frac{\sqrt{\lambda_k}}{\lambda_k}\sum_{i=1}^{n} v_{ik}S(x, x_i)\right)^2. \tag{4.13}$$

4.2.5 Experimental results

4.2.6 Datasets and experimental setup

The dissimilarity datasets were selected for the experiments based on the existence of clusters in the data. The Ionosphere dataset consists in radar data [73] where the $L1$ distance is used. The Kimia dataset is based on the shape contexts descriptor [74] computed for the Kimia shapes data [75]. The dissimilarity is based on sums of matching costs for the best matching points defining two shapes, plus the amount of transformation needed to align the shapes. The dissimilarity data set Chickenpieces-20-60 [43] is composed by edit distances from string representations of the angles between segments defining the contours of chicken pieces images. The Ringnorm dataset is the one presented in [76]; it is originally a 20-dimensional, 2-class data, where the first class is normally distributed with zero mean and covariance matrix 4 times the identity. The second class has unit covariance matrix and mean close to zero. We use only the first 2 features and the $L2$ distance. The characteristics of the datasets as well as the cardinality of the training sets used are presented in Table 5.3.

Table 4.5: Properties of the datasets used in this study, Symm. and Metric refers to whether the data is symmetric or metric, the $|T|$ column refers to the training set cardinality used for the experiments.

Datasets	# Classes	# Obj. per class	Symm.	Metric	$\|T\|$
Ionosphere	2	225,126	yes	yes	140
Kimia	18	18×12	no	no	90
Rings	2	440,449	yes	yes	222
ChickenPieces-20-60	5	117,76,96,61,96	no	no	158

As classifier we used the support vector machine (SVM) classifier. For the SVM we used a linear kernel and a fixed appropriately selected cost parameter $C = 1$. Note that despite the fact that the curse of dimensionality was mentioned as a limitation of high-dimensional spaces, the SVM classifier is able to handle high dimensions well. This makes our comparisons more fair for the high-dimensional representations. However, the limitation was mentioned since in many applications one may want to use classifiers that suffer from the curse of dimensionality and resorting to low-dimensional representations by prototype selection is one option to overcome the problem. Our proposals are the following cluster-based methods: selection by GA of clusters created using minimum, maximum, average and subspace distances of training objects to the clusters. The cluster-based methods are compared with some of the best prototype selectors presented in the literature (which select objects as prototypes), with representatives of unsupervised and supervised methods: Forward selection [11] optimizing the LOO 1-NN error in the DS, Kcentres prototype selector [11], random selection, selection by GA of the best clusters centers, and selection by GA of the best prototypes from the whole candidate set. In addition, we compared the approach using all candidate objects as prototypes.

A set of 5 to 20 prototype clusters/objects are selected. However, the total number returned by the affinity propagation is about 25 clusters. Averaged errors and standard deviations over 30 experiments are reported in Table 4.6 for the dimension where the best overall classification result with the SVM was obtained. Objects in each dataset are randomly split 30 times into training, representation, and test sets. Clusters are computed on the representation set which also contains the candidate objects for prototypes, the best clusters and objects are selected optimizing the criteria for the training set by which the classifiers are trained, and the final classification errors are computed for the test sets. We performed a t-test to find if the differences between the mean errors of the best overall result and the mean errors achieved by the other approaches was statistically significant, the level of significance used is 0.05. In the case that a cluster-based method was the best, the statistical significance is computed with respect to the non cluster-based approaches.

4.2.7 Results and discussion

In Table 4.6 it can be seen that classification results in the GDS generated by selected clusters outperform the classification results in DS with selected objects as prototypes for the same dimensions of the spaces. For the Ionosphere and Kimia datasets the best method uses clusters with minimum distance, this is in agreement with previous findings for graph dissimilarities in [68]. In the Ionosphere and Kimia datasets, the selection of clusters using maximum distance is usually among the worse alternatives. This may be expected since it may be very sensitive to outliers. However, in the Rings dataset the clusters based on maximum distances provide the best overall result. In the case of Chicken Pieces, the best results are obtained using all objects as prototypes, perhaps because this dataset has a high intrinsic dimension (176) according to the number of significant eigenvalues of the covariance matrix in the DS. Therefore, in order to

Table 4.6: Mean and standard deviation of errors over 30 experiments. The best overall result is reported for each dataset with the corresponding results of the other methods for the same dimension of the space (in parenthesis). When the difference of the best result with the other standard approaches is statistically significant, it is reported in bold.

Selectors \ Datasets	Ionosph(15)	Kimia(20)	Rings(20)	Chicken Pieces(20)
Clusters minimum	$\mathbf{0.063 \pm 0.028}$	$\mathbf{0.047 \pm 0.032}$	0.265 ± 0.0205	0.11 ± 0.025
Clusters maximum	0.09 ± 0.029	0.11 ± 0.054	$\mathbf{0.263 \pm 0.0236}$	0.15 ± 0.028
Clusters average	0.072 ± 0.023	0.06 ± 0.045	0.274 ± 0.0181	0.09 ± 0.024
Clusters subspace	0.073 ± 0.022	0.07 ± 0.048	0.276 ± 0.0193	0.088 ± 0.023
Random	0.086 ± 0.026	0.12 ± 0.057	0.274 ± 0.0181	0.17 ± 0.039
GA (whole set)	0.082 ± 0.028	0.1 ± 0.043	0.274 ± 0.0181	0.16 ± 0.028
GA (cluster centres)	0.085 ± 0.032	0.094 ± 0.05	0.275 ± 0.0177	0.15 ± 0.029
Forward selection	0.09 ± 0.027	0.12 ± 0.054	0.274 ± 0.0184	0.16 ± 0.036
Kcentres	0.082 ± 0.029	0.13 ± 0.061	0.274 ± 0.0181	0.15 ± 0.036
All	0.083 ± 0.033	0.068 ± 0.042	0.274 ± 0.0181	$\mathbf{0.077 \pm 0.017}$

obtain good results, high-dimensional spaces are needed. However, the average and subspace distance to clusters outperformed the other approaches that create low-dimensional spaces.

Cluster-based approaches create irregular kernels which nonlinearly map the data to the GDS in a better way than the object-based approaches for the same dimensions. We computed the nonlinear mapping for the Rings data from the underlying feature space to a Hilbert space using a second degree polynomial kernel and applied SVM classification with this kernel and regularization parameter optimized. We corroborate that the results were very similar to the ones obtained using clusters in the dissimilarity space. Cluster-based prototypes allow one to apply linear classifiers with good results to originally nonlinear data. The same can be achieved by kernels and SVM if the dissimilarities are Euclidean (they are transformed to the equivalent kernel). However, the original SVM will not work anymore for a non-Euclidean dissimilarity matrix but a nonlinear mapping to the DS or GDS can still be achieved for non-Euclidean data (e.g. the Kimia dataset).

The main disadvantage of using cluster-based prototypes compared to object-based prototypes for spaces of the same dimension is the computational cost, since, when using clusters, more dissimilarities must be measured. In this case, for training and test objects, the dissimilarities with all the objects in the clusters must be computed in order to find the minimum, maximum and average dissimilarity. However, when compared to the approach using all objects as prototypes, the computational cost of the cluster-based approach is smaller because some clusters are discarded in the selection process and, thereby, less dissimilarity computations are made for training and test objects. Since the dissimilarity matrix is computed in advance before prototype selection is executed, the proposed approach as well as the standard prototype selection methods have limitations in case of very large datasets. This remains open for further research.

4.2.8 Conclusions

For the selection of prototypes not only the optimization method and the criterion are important, but also how the prototypes are devised is vital. We found that clusters may be useful to obtain low-dimensional GDSs in the case of datasets that present clusters. Our approach is useful for problems where the use of cluster-based prototypes make sense according to the data distribution. Note that our results hold for small and moderate training set sizes. When large training sets are available, they may compensate for bad mappings using objects as prototypes.

In general, we found that the minimum, average and subspace distances to clusters perform well in real-world datasets. However, there is no "best" approach among the cluster-based methods, it seems that the best option depends on specific data characteristics. Our intuition is that the minimum distance seems to be more meaningful for measuring distances with sets of objects with a shape such as the clusters. The cluster-based approaches improve the results of using DS of the same dimension but created by selected objects as well as DS using all the objects as prototypes (high-dimensional). Future works may be devoted to study the sensitivity to the choice of different clustering methods as well as the influence of numbers and sizes of the clusters.

Chapter 5

Devising and selecting the prototypes in extended dissimilarity spaces

5.1 On using asymmetry information for classification in extended dissimilarity spaces

This section has been published as "On Using Asymmetry Information for Classification in Extended Dissimilarity Spaces", by Yenisel Plasencia-Calaña, Edel Garcia-Reyes, Robert P. W.Duin and Mauricio Orozco-Alzate, in *Proceedings of the 17th Iberoamerican Congress on Pattern Recognition, CIARP 2012, LNCS*.

Abstract

When asymmetric dissimilarity measures arise, asymmetry correction methods such as averaging are used in order to make the matrix symmetric. This is usually needed for the application of pattern recognition procedures, but in this way the asymmetry information is lost. In this paper we present a new approach to make use of the asymmetry information in dissimilarity spaces. We show that taking into account the asymmetry information improves classification accuracy when a small number of prototypes is used to create an extended asymmetric dissimilarity space. If the degree of asymmetry is higher, improvements in classification accuracy are also higher. The symmetrization by averaging also works well in general, but decreases performance for highly asymmetric data.

5.1.1 Introduction

Dissimilarity representations [3] arose as an alternative to feature-based representations when the definition and extraction of good features is difficult or intractable while a robust dissimilarity measure can be defined more easily for the problem at hand. Research in this field has focused on several topics: prototype selection [11, 47] or generation [31], classification in dissimilarity spaces [14, 77], among others. One open issue corresponds to the information usage in dissimilarity matrices: they can be asymmetric but most of the traditional classification and clustering methods are thought for symmetric dissimilarity matrices. In case of asymmetry, the typical approach is to symmetrize the matrix with any known symmetrization method, and then apply the methods on the symmetric variant. This might carry a loss of useful information.

Asymmetric dissimilarity or similarity measures can arise in several situations; see [78] for a general analysis of the causes of non-Euclidean data. Measures resulting from a matching process may appear to be asymmetric due to a suboptimal procedure. Also, measures designed using expert knowledge for the problem might not be symmetric. One example is fingerprint matching [47], where measures are often asymmetric. When various dissimilarity matrices are combined, the final matrix can also be asymmetric. One of the most widely used methods for symmetrization is the average method. In [3], before embedding asymmetric dissimilarity matrices into Pseudo-Euclidean spaces, the average method is used to make the matrix symmetric. In [11], the dissimilarity matrix is symmetrized using the average method in order to allow the use of some prototype selection algorithms in the dissimilarity space (DS). Other authors, in the context of kernel-based classification, proposed the use of a positive semidefinite matrix K^tK, where K denotes a nonsymmetric kernel [79].

Different variants of the Multidimensional Scaling algorithm have incorporated asymmetry in an intuitive way, by defining a skew symmetric term [80]. In [81], the authors proposed modifications to Self Organizing Map and Sammon Mapping in order to deal with asymmetric proximities showing that the proposed algorithms outperformed their symmetric variants. In [82], the authors compared several symmetrization methods of asymmetric kernel matrices for their use in the context of Support Vector Machines. They also proposed a simple supervised symmetrization method that outperformed the other methods compared.

One question that arises is whether the asymmetry information can be useful for classification in dissimilarity spaces, instead of ignoring it or using a symmetrization method. Another question is how we can use the asymmetry information in the context of classification in dissimilarity spaces. In this paper we propose a new approach for using asymmetry information in what we called the extended asymmetric dissimilarity space (EADS). As the dimension of the EADS space is twice the dimension of the original DS, the use of prototype selection is needed in order to reduce the dimensions before the EADS is constructed. Results are provided comparing classification errors in both the DS and EADS for four standard asymmetric dissimilarity data sets.

5.1.2 Dissimilarity space and extended dissimilarity space

Dissimilarity representations arose from the idea that the classes are constituted by similar objects, so the nearness information is more fundamental than features to discriminate between the classes [3]. In this context, the DS was proposed in [3] as follows. Let $R = \{r_1, r_2, ..., r_k\}$ be the representation set: a collection of prototypes that may be a subset of the training set T. Let d be a dissimilarity measure for the problem at hand. The DS is created by a mapping of the objects to the space defined by the dissimilarities to the prototypes, where each dimension corresponds to the dissimilarities to a given prototype. The representation d_x of an object x is:

$$d_x = [d(x, r_1)\ d(x, r_2)\ ...\ d(x, r_k)]. \tag{5.1}$$

The DS was postulated as a Euclidean vector space, making suitable the use of traditional classifiers for feature spaces like Bayesian ones. The cardinality of the representation set defines the dimension of the DS. For the reduction of the representation set, prototype selection methods are used. They allow one to determine the desired tradeoff between classification accuracy and representation cardinality.

In this subsection we present the EADS. The motivation for this proposal is that when projecting asymmetric data in the DS, asymmetry information is lost because we only use dissimilarities from the objects to the prototypes, and not from prototypes to objects. If the matrix is previously symmetrized, we are also neglecting the asymmetry present in the data. In order to take advantage of the asymmetry information in both directions, we explore the use of an extended representation of the initial asymmetric dissimilarity matrix in an EADS. We propose to create the EADS using the prototypes selected from the original dissimilarity matrix as it is given. Then, having those prototypes $R = \{r_1, r_2, ..., r_k\}$, the representation of an object in the EADS is defined by:

$$d_x = [d(x, r_1)\ d(x, r_2)\ ...\ d(x, r_k)\ d(r_1, x)\ d(r_2, x)\ ...\ d(r_k, x)]. \tag{5.2}$$

In order to represent the training set and the objects submitted for classification in the EADS, we need to measure the dissimilarities from the objects to the prototypes and from the prototypes to the objects. As a result, the dimension of the EADS space is twice the dimension of the DS. Classifiers can be trained in the EADS in the same way they are trained in the DS.

5.1.3 Datasets and experimental setup

Our goal is to compare the discriminative power of the EADS with the discriminative power of the non-symmetrized version and the one symmetrized by averaging. Classification errors are presented using different numbers of prototypes in DS and EADS. Prototypes are the same for both spaces, but in the DS only dissimilarities in one direction are used. In the EADS, dissimilarities from the two directions are used. This leads to a space of dimension twice the size of the DS dimension.

For the experiments we used four data sets: Chickenpieces-20-60, Chickenpieces-35-45, CoilYork, and Zongker. The dissimilarity data set Chickenpieces-20-60 [43] is computed from a set of images in binary format representing silhouettes from five different parts of the chicken. From these images, the edges are approximated by segments of length 20 pixels, and a string representation of the angles between the segments is derived. The dissimilarity matrix is composed by edit distances between these strings. The cost function between the angles is defined as the difference in degrees in case of substitution and as 60 in case of insertion or deletion. The Chickenpieces-35-45 was obtained with the same methodology but for this data the segments are of length 35 and the cost of insertion and deletion is 45.

The CoilYork data set is composed by dissimilarities between a set of graphs derived from

four objects of the COIL database, the graphs are the Delaunay triangulations derived from corner points of the images [44]. The dissimilarity matrix is constructed by graph matching, using the algorithm proposed in [45].

The Zongker digit similarity data between 2000 handwritten NIST digits of 10 classes, is based on deformable template matching. The similarity measure is the result of an iterative optimization of the non-linear deformation of the grid [53]. It is transformed into a dissimilarity matrix as proposed in [3], but with the slight modification of discarding the symmetrization step.

Some important characteristics of the data sets can be found in Table 5.3. The Asymmetry column shows an asymmetry coefficient ac for each data set, this was computed using the following equation:

$$ac = \frac{1}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{|d_{ij} - d_{ji}|}{\min(d_{ij}, d_{ji}) + \epsilon}, \tag{5.3}$$

where n is the number of objects in the data set. We assume that dissimilarities between different objects will not be zero. In case that it is known beforehand that zero dissimilarities between different objects may arise, a term with a very small value such as $\epsilon = 0.0001$ must be added in the denominator to avoid the indefinite result of the division by zero.

As classifier, the Linear Bayes Normal (BayesL) was used in both the DS and EADS. It

Table 5.1: Characteristics of the data sets, the $|X|$ column is the number of training objects, and $|T|$ is the number of test objects.

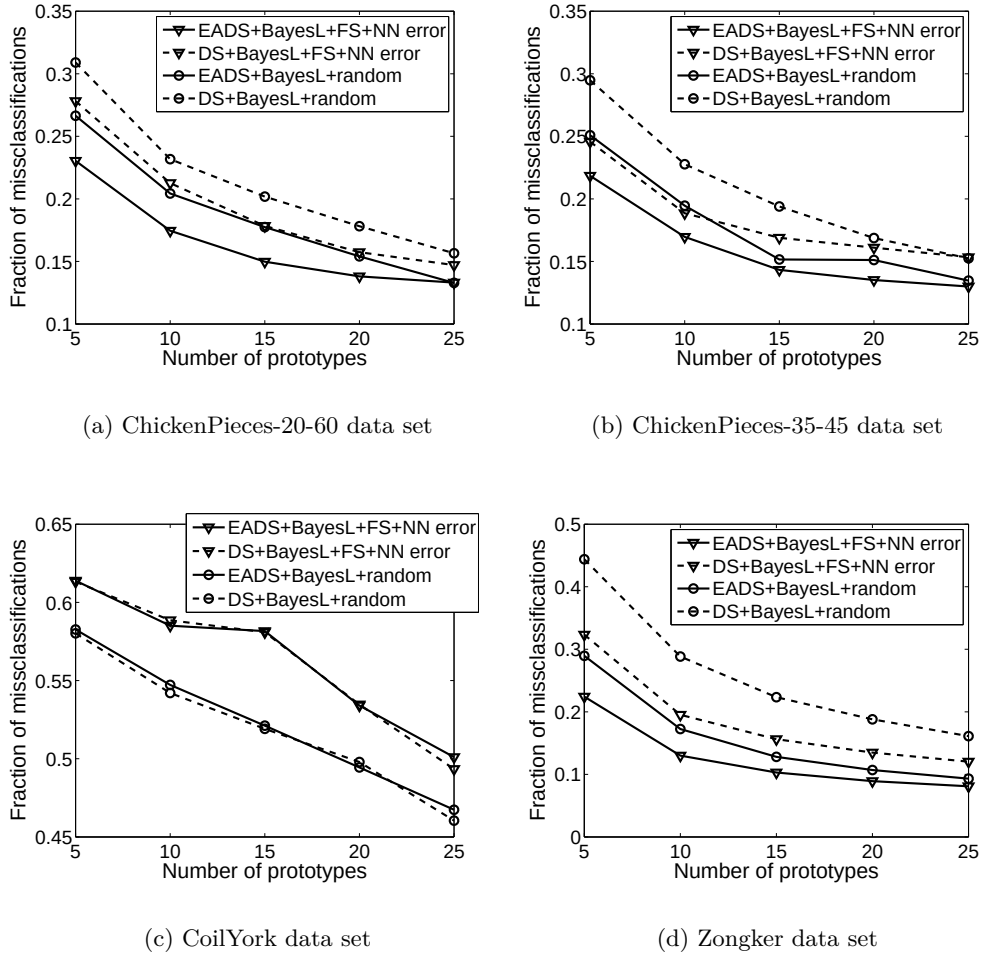
Data sets	# Classes	# Obj. per class	Asymmetry	$\|X\|$	$\|T\|$
ChickenPieces-20-60	5	117,76,96,61,96	0.05	222	224
ChickenPieces-35-45	5	117,76,96,61,96	0.08	222	224
CoilYork	4	4x72	0.009	144	144
Zongker	10	10x200	0.18	400	1600

is a simple and fast classifier that is optimal for normally distributed classes with equal covariances. Experiments were repeated twenty times using equal-sized random partitions for training and testing for ChickenPieces and CoilYork data sets, and twenty and eighty percent for training and testing respectively in the Zongker data set. Results were averaged over the twenty experiments. As prototype selectors, two different methods are used: the systematic forward selection optimizing the leave-one-out nearest neighbour error on the training set as in [11] (FS+NN error), and the random selection. The methods selected 5, 10, 15, 20 and 25 prototypes. The BayesL and prototype selectors were trained using the training data, and the classification results were computed in the test set for the DS and EADS generated using the prototypes selected with the different methods. Regularization parameter of BayesL is 0.01.

5.1.4 Results and discussion

Figure 5.1 shows the curves of error rates for an increasing number of prototypes in the original asymmetric representation in the DS and the representation in the EADS. Figure 5.2 shows the curves of error rates for an increasing number of prototypes comparing the symmetrized representation in the DS using the average and the representation in the EADS. Solid lines represent the approaches in EADS; dashed lines represent the approaches in DS. The same symbol is assigned for the results in DS and EADS using the same prototype selector. Standard deviations are between 0.007 and 0.08.

From the results in Fig. 5.1 we can see that in three of the four data sets —the ChickenPieces-



(a) ChickenPieces-20-60 data set

(b) ChickenPieces-35-45 data set

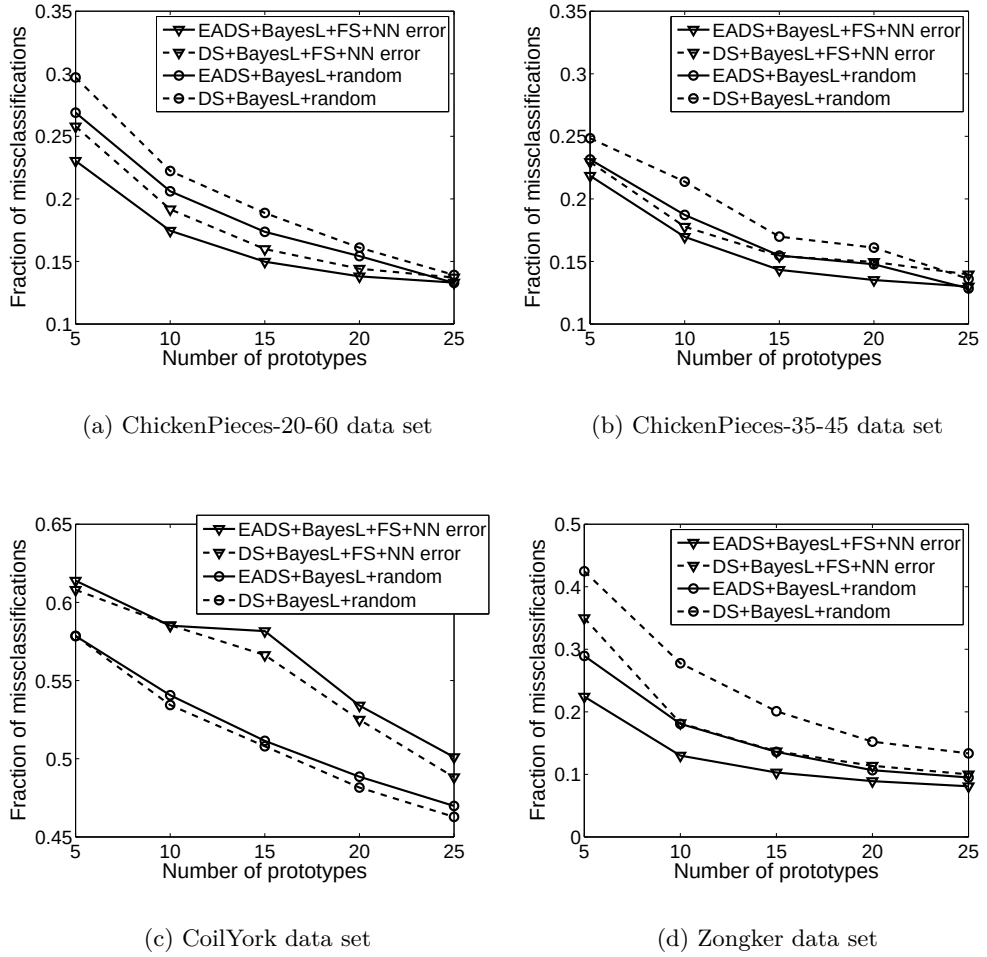
(c) CoilYork data set

(d) Zongker data set

Figure 5.1: Classification results for the original asymmetric representation in the DS and the EADS in the data sets, the dimension of the associated DS is equal to the number of prototypes, and the dimension of the associated EADS is twice the number of prototypes.

20-60, ChickenPieces-35-45, and Zongker— classification in EADS outperforms classification in DS using both the systematic and the random prototype selectors. These are the data sets with the higher degree of asymmetry as measured by the asymmetry coefficient. In the CoilYork data set, which has the smallest asymmetry degree, the results in the EADS were a little worse than those in the DS. Except for the CoilYork data set, when the number of prototypes increases, the difference between the error rates in EADS and DS decreases. This implies that the asymmetry information is more useful if small sets of prototypes are used, and having more dimensions compensates for not using asymmetry information.

From Fig. 5.2 we can see that once the dissimilarities are symmetrized by averaging, incorporating the asymmetry information does not improve classification in the same extent as by using the non-symmetrized version. This shows that the symmetrization by averaging is a good alternative for dealing with asymmetric data. In the CoilYork data set, the EADS performed worse than the DS using the symmetrized dissimilarities. In this case, where the asymmetry coefficient has a very small value, the use of asymmetry information leads to a slight decrease in classification performance. The symmetrization by averaging becomes less useful when the asymmetry degree of the data increases as it can be deduced from the similar classification errors in the original DS (see Fig. 5.1, (d)) and the DS symmetrized by averaging (see Fig. 5.2,



(a) ChickenPieces-20-60 data set

(b) ChickenPieces-35-45 data set

(c) CoilYork data set

(d) Zongker data set

Figure 5.2: Classification results for the representation in the DS symmetrized by averaging and the EADS in the data sets, the dimension of the associated DS is equal to the number of prototypes, and the dimension of the associated EADS is twice the number of prototypes.

(d)) in the Zongker data.

From the results, we made a characterization of the relationship between the amount of asymmetry present in each data set measured by the asymmetry coefficient and the improvements obtained in classification in the EADS compared to the non-symmetrized DS. First, we sorted the asymmetry coefficients of each data set in increasing order, and plotted the classification improvements in EADS compared to DS measured by the differences between the curves for the same prototype selection method in both spaces. The sum of these differences was plotted for each data set with its related asymmetry coefficient, see Fig. 5.3.

In the function we can see a positive linear correlation between the two variables, as the value of the asymmetry coefficient increases, the value of the improvements in classification also increases. The value of the correlation coefficient was 0.99. This means that it is important to take the asymmetry information into account in order to improve classification rates when the asymmetry degree is perceivable, and while the data is more asymmetric the classification improvement increases. In the CoilYork dataset we obtained a negative value of improvement equal to -0.01, since the EADS performed slightly worse than the DS.

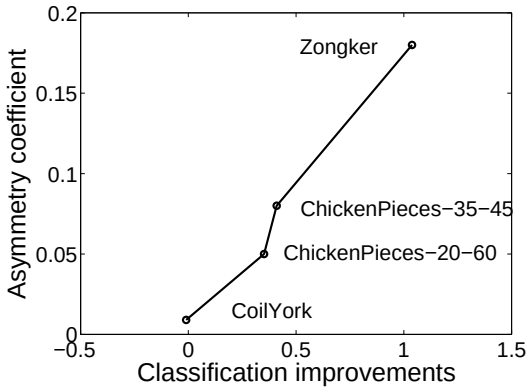


Figure 5.3: Classification improvements in EADS compared to DS as a function of the asymmetry coefficient

5.1.5 Conclusions

We proposed the EADS that proved to be suitable for exploiting the asymmetry information in the dissimilarities. This is especially useful for small prototype sets. For a data set with a very small degree of asymmetry, it might not be necessary and can even be slightly detrimental to use asymmetry information. Another conclusion is that the symmetrization by averaging is a good alternative for dealing with asymmetric data, although it becomes less useful when the asymmetry degree of the data increases. In our results, the improvements achieved in classification in EADS are positively correlated to the degree of asymmetry in each data set. The use of EADS can be beneficial when one has a very small set of informative prototypes with a highly asymmetric data set. The symmetrization operation may depend on the cause of asymmetry, e.g. averaging can be good for asymmetry caused by noise, the minimum can be useful for asymmetry caused by a shortest path optimization to compute the dissimilarities. Further work may be devoted to study these operations and the usefulness of the EADS for asymmetry caused by expert knowledge incorporated in the dissimilarity measure, noise or suboptimal procedures.

5.2 On the informativeness of asymmetric dissimilarities

This section has been published as "On the informativeness of asymmetric dissimilarities", by Yenisel Plasencia Calaña, Veronika Cheplygina, Robert P. W. Duin, Edel Garcia-Reyes, Mauricio Orozco Alzate, David M. J. Tax, and Marco Loog, in *Proceedings of the second International Workshop on Similarity Based Pattern Analysis and Recognition, SIMBAD 2013, LNCS*.

Abstract

A widely used approach to cope with asymmetry in dissimilarities is by symmetrizing them. Usually, asymmetry is corrected by applying combiners such as average, minimum or maximum of the two directed dissimilarities. Whether or not these are the best approaches for combining the asymmetry remains an open issue. In this paper we study the performance of the extended asymmetric dissimilarity space (EADS) as an alternative to represent asymmetric dissimilarities for classification purposes. We show that EADS outperforms the representations found from the two directed dissimilarities as well as those created by the combiners under consideration in several cases. This holds specially for small numbers of prototypes; however, for large numbers of prototypes the EADS may suffer more from overfitting than the other approaches. Prototype selection is recommended to overcome overfitting in these cases.

5.2.1 Introduction

Statistical and structural representations of patterns are two complementary approaches in pattern recognition. Recently, dissimilarity representations [3, 10] arose as a bridge between these representations. Dissimilarities can be computed from the original objects, but also on top of features or structures such as graphs or strings. This provides a way for bridging the gap between structural and statistical approaches. Dissimilarities are also a good alternative when the definition and selection of good features can be difficult or intractable (e.g. the search for the optimal subset of features has a computational complexity of $O(2^n)$, where n is the number of features) while a robust dissimilarity measure can be defined more easily for the problem at hand.

The classification of objects represented in a dissimilarity space (DS) has been an active research topic [11, 14, 29, 47, 77], but not much attention has been paid to the treatment of the asymmetry that can be present in the dissimilarities. Most traditional classification and clustering methods are devised for symmetric dissimilarity matrices, and therefore cannot deal with asymmetric input. In order to be suitable for these methods, asymmetric dissimilarities need to be symmetrized, for instance by averaging the matrix with its transpose. However, in the dissimilarity space, symmetry is not a required property and therefore a wider range of procedures for classification can be applied.

Asymmetric dissimilarity or similarity measures can arise in several situations; see [78] for a general analysis of the causes of non-Euclidean data. Asymmetry is common in human judgments. Including expert knowledge in defining a (dis)similarity measure, such as for fingerprint matching [47], may lead to asymmetry. In general, matching processes may often lead to asymmetric dissimilarities. Exact matches are often impossible and suboptimal procedures may lead to different matches from A to B than from B to A.

Symmetrization by averaging is widely used before embedding asymmetric dissimilarity data into (pseudo-)Euclidean spaces [3]. The use of a positive semi-definite matrix K^TK, where K denotes a nonsymmetric kernel [83] is also proposed in the context of kernel-based classification to make the kernel symmetric. A comparative study of methods for symmetrizing the kernel matrix for the application of the support vector machine (SVM) classifier can be found in [82]. While such methods that require symmetrized matrices show good results, it remains an open question whether asymmetry is an undesirable property, or that it, perhaps, contains useful information that is disregarded by symmetrization.

In this paper, we explore using the information from asymmetric dissimilarities by concatenating them into an extended asymmetric dissimilarity space (EADS). Following up on [24], we investigate a broader range of circumstances where EADS may be a good choice for representation, and compare EADS to the directed dissimilarities, as well as to several symmetrization methods. The representation is studied for two shape matching and two multiple instance learn-

ing (MIL) problems. We show that EADS is able to outperform the directed and symmetrized dissimilarities, especially in cases where both directed dissimilarities are informative. It must be noted that EADS doubles the dimensionality of the problem, which may not be desirable. Therefore, we also include results using prototype selection in order to compare dissimilarity spaces with the same dimensionality, and show that EADS also leads to competitive results in the examples considered.

We begin with a number of examples that lead to asymmetric dissimilarities in Subsection 5.2.2. The dissimilarity space is explained in Subsection 5.2.5. Ways of dealing with asymmetry are then described: symmetrization (Subsection 5.2.7) and the proposed EADS (Subsection 5.2.8). The datasets used, experimental setup, results and discussion are provided in Subsections 5.3.5 and 5.2.10, followed by the conclusions in Subsection 5.2.11.

5.2.2 Asymmetric dissimilarities

Although our notions of geometry may indicate otherwise, asymmetry is a natural characteristic when the concept of similarity or proximity is involved. Just think of a network of roads, where the roads can be one-way streets and one street is longer than the other. It is then clear that traveling from A to B may take longer than returning from B to A. Asymmetric dissimilarities also appear in human judgments [84]: it may be more natural to say that "Dutch is similar to German" than "German is similar to Dutch" because more people might be familiar with the German language and it is therefore a better point of reference for the comparison. Interestingly, this is also evidenced by the number of hits in Google: about ten times as many for the "Dutch is similar to German" sentence. When searching for these sentences in Dutch, the reverse is true. Here we provide two examples of pattern recognition domains which may also naturally lead to asymmetric dissimilarities.

5.2.3 Shapes and images

One possible cause of asymmetry is that the distances used directly on raw data such as images may be expensive to compute accurately. For example in [43], the edit distance used between shapes is originally symmetric. The distance has the problem that the returned values are different if the starting and ending points of the string representation of the shape are changed. In order to overcome this drawback, an improved rotation invariant distance was proposed. The computation of the new distance suffers from a higher computational complexity. Therefore, suboptimal procedures are applied in practice and, as a consequence, the distances returned are asymmetric.

In template matching, the dissimilarity measure may be designed to compute the amount of deformation needed to transform one image into the other as in [53]. The amount of deformation required to transform image I_j into image I_k is generally different from the amount of deformation needed to transform image I_k into I_j. This makes the resulting dissimilarity matrix asymmetric.

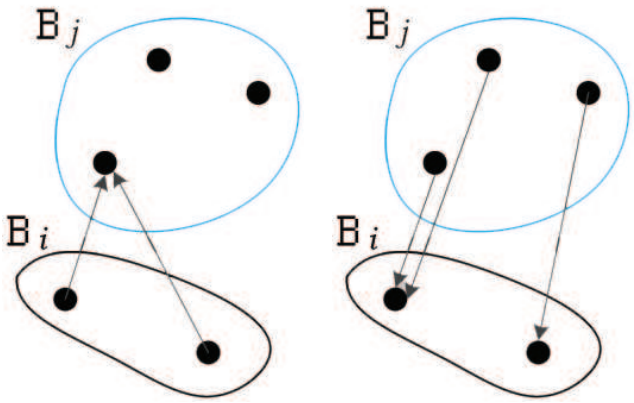
5.2.4 Multiple instance learning

Multiple instance learning (MIL) [85] extends traditional supervised learning methods in order to learn from objects that are described by a set (*bag*) of feature vectors (*instances*), rather than a single feature vector only. The bag labels are available, but the labels of the individual instances are not. A bag with n_i instances is therefore represented as (B_i, y_i) where $B_i = \{x_{ik}; k = 1...n_i\}$. In this setting, traditional supervised learning techniques cannot be applied directly.

It is often assumed that the instances have (hidden) labels which influence the bag label. For instance, one assumption is that a bag is positive if and only if at least one of its instances

is positive. Such positive instances are also called concept instances. One application for MIL is image classification. An image with several regions or segments can be represented by a bag of instances, where each instance corresponds to a segment in the image. For images that are positive for the "Tiger" class, concept instances are probably segments containing (parts of) a tiger, rather than segments containing plants, trees and other surroundings.

One of the approaches to MIL is to learn on bag level, by defining kernels [86] or (dis)similarities [87, 88] between bags. Such dissimilarities are often defined by matching the instances of one bag to instances of another bag, and defining a statistic (such as average or maximum) over these matches. This creates asymmetric dissimilarities, as illustrated in Fig.5.4.



(a) Dissimilarity from B_i to B_j

(b) Dissimilarity from B_j to B_i

Figure 5.4: Asymmetry in bag dissimilarities. The minimum distances of one bag's instances are shown. In this paper, the bag dissimilarity is defined as the average of these minimum distances.

The direction in which the dissimilarity is measured defines which instances influence the dissimilarity. When using a positive prototype, it is important that the concept instances are involved, as these instances are responsible for the differences between the classes. Therefore, for positive prototypes it is expected that the dissimilarity from the prototype to the bag is more informative than the dissimilarity from the bag to the prototype. A more detailed explanation of this intuition is given in [88].

5.2.5 Dissimilarity space

The DS was proposed in the context of dissimilarity-based classification [3]. It was postulated as a Euclidean vector space, implying that classifiers proposed for feature spaces can be used there as well. The motivation for this proposal is that the proximity information is more important for class membership than features [3]. Let $R = \{r_1, ..., r_k\}$ be the representation set, where k is its cardinality. This set is usually a subset of the training set T, though a semi-supervised approach with more prototypes than training objects may be preferable [89]. In order to create the DS, using a proper dissimilarity measure d, the dissimilarities of training objects to the prototypes in R are computed. The object representation is a vector of the object's dissimilarities to all the prototypes. Therefore, each dimension of the DS corresponds to the dissimilarities to some prototype. The representation d_x of an object x is:

$$d_x = [d(x, r_1) \ ... \ d(x, r_k)] \tag{5.4}$$

5.2.6 Prototype selection

Prototype selection has been proposed for the dimension reduction of DS [11]. Supervised (wrapper) and unsupervised (filter) methods can be considered for this purpose as well as different optimization strategies to guide the search. They select the "best" prototypes according to their criterion. The selected prototypes are used for the generation of the DS. Prototype selection allows one to obtain low-dimensional spaces avoiding as much as possible a decrease in performance (e.g. classification accuracy). Therefore, they are very useful to achieve a trade-off between the desirable properties of compact representation and reasonable classification accuracy. The approach considered in this study for selecting prototypes is the forward selection optimizing the leave-one-out (LOO) nearest neighbour (1-NN) error (so supervised) in the dissimilarity space for the training set. It starts from the empty set, and sequentially adds the prototype that together with the selected ones ensures the best 1-NN classification accuracy.

5.2.7 Combining the asymmetry information

For two point sets, there are different ways to combine the two directed asymmetric dissimilarities. The maximum, minimum and average are used extensively and are very intuitive. Let $A = \{a_1, ..., a_k\}$ and $B = \{b_1, ..., b_l\}$ be two sets of points, and $D_1 = d(A, B)$ and $D_2 = d(B, A)$ the two directed dissimilarities. The maximum, minimum and average combiners are defined in (5.5) to (5.7) respectively:

$$max(A, B) = \max(D_1, D_2) \tag{5.5}$$

$$min(A, B) = \min(D_1, D_2) \tag{5.6}$$

$$avg(A, B) = \frac{1}{2}(D_1 + D_2) \tag{5.7}$$

All these rules for combining asymmetry information ensure a symmetric measure.

5.2.8 Extended asymmetric dissimilarity space

For the purpose of combining the asymmetry information in both directions, we study the EADS. From the two directed dissimilarities D_1, D_2, we have that $D_i \to X_i \in \mathbb{R}^k, i = 1, 2$ represents the mapping of the dissimilarities to the dissimilarity space. The EADS is constructed by: $[D_1\ D_2] \to X_1 \times X_2 \in \mathbb{R}^{k \times 2}$, which means that the extended space is the Cartesian product of the two directed spaces. Given the prototypes $R = \{r_1, ..., r_k\}$, the representation of an object in the EADS is defined by:

$$\mathbf{d}_x = [d(x, r_1)\ ...\ d(x, r_k)\ d(r_1, x)\ ...\ d(r_k, x)] \tag{5.8}$$

In the case that we have the full dissimilarity matrix using all training objects as prototypes, the EADS is constructed from the concatenation of the original matrix and its transpose. Rows of this new matrix correspond to the representation of objects in the EADS. As a result, the dimension of the EADS is twice the dimension of the DS. Classifiers can be trained in the EADS in the same way they are trained in the DS. By doubling the dimension, the expressiveness of the representation is increased. This may be particularly useful when the number of prototypes is not very large. When the number of prototypes is large compared to the number of training objects, the EADS is expected to be more prone to overfitting than any of the symmetrized approaches.

Despite the fact that in the EADS symmetric distances or similarity measures can be used on top of the asymmetric representation, this does not mean that we are not exploiting the asymmetry information present in the original dissimilarities. The original asymmetric dissimilarities in the two directions are used in the object representation that is the input for classifiers in the EADS. These classifiers can use any symmetric distance or kernel computed on top of the representation.

Note that if the asymmetry does not exist in the measure, the representation of objects in the EADS contains the same information replicated. These redundancies in the best case lead to the same classification results as in the standard DS using only one direction [24]. However, it may even be counterproductive since it may lead to overfitting and small sample size problems for some classifiers. Therefore, doubling the dimension is not the cause for possible classification improvements when using the EADS. The fact that the two asymmetric dissimilarities are taken into account in the representation is what may help the classifiers to improve their outcomes.

5.2.9 Datasets and experimental setup

In this subsection we first describe the datasets and how the corresponding dissimilarity matrices are obtained. This is followed by the experimental setup and a discussion of the results. The dissimilarity dataset Chickenpieces-35-45 is computed from the Chickenpieces image dataset [43]. The images are in binary format representing silhouettes from five different parts of the chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). From these images the edges are extracted and approximated by segments of length 35 pixels, and a string representation of the angles between the segments is derived. The dissimilarity matrix is composed by edit distances between these strings. The cost function between the angles is defined as the difference in degrees in case of substitution, and as 45 in case of insertion or deletion.

The Zongker digit dissimilarity data is based on deformable template matching. The dissimilarity measure was computed between 2000 handwritten NIST digits in 10 classes. The measure is the result of an iterative optimization of the non-linear deformation of the grid [53].

AjaxOrange is a dataset from the SIVAL multiple instance datasets [90]. The original dataset has 25 distinct objects (such as bottle of dish soap called AjaxOrange) portrayed against 10 different backgrounds, and from 6 different orientations, resulting in 60 images for each object. This dataset has been converted into 25 binary MIL datasets by taking one class (AjaxOrange) in this case as the positive class (with 60 bags), and all others (with 1440 bags) as the negative one. Each image is represented by a bag of segments, and each segment is described by a feature vector with color and texture features.

The dissimilarity of two images is computed by what we call the meanmin dissimilarity, which is similar to modified versions of the Hausdorff distance:

$$d_{meanmin}(B_i, B_j) = \frac{1}{|B_i|} \sum_{x_{ik} \in B_i} \min_{x_{jl} \in B_j} d(x_{ik}, x_{jl}) \tag{5.9}$$

where $d(x_{ik}, x_{jl})$ is the squared Euclidean distance between two feature vectors.

Winter Wren is one of the binary MIL bird song datasets [91], created in a similar one-against-all way as SIVAL. Here, a bag is a spectrogram of an audio fragment with different birds singing. A bag is positive for a particular bird species (e.g. Winter Wren) if its song is present in the fragment. There are 109 fragments where the Winter Wren song is heard, and 439 fragments without it. Also here we use (5.9) to compute the dissimilarities.

The datasets and their properties are shown in Table 5.3. For each dissimilarity matrix we

computed its asymmetry coefficient as follows:

$$AC = \frac{1}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{|d_{ij} - d_{ji}|}{\min(d_{ij}, d_{ji}) + \epsilon} \tag{5.10}$$

where n is the number of objects in the dataset. This coefficient measures the average normalized difference of the directed dissimilarities and is 0 for symmetric data.

The formulation in (5.10) assumes that $d_{ij} \neq 0$ for $i \neq j$, which may not necessarily be true for dissimilarity data. In the case that $d_{ij} = d_{ji}$, a term ϵ with a very small value such as 0.0001 must be added in the denominator to avoid divisions by zero.

Table 5.2: Properties of the datasets used in this study, AC refers to the asymmetry coefficient from (5.10); the larger the AC, the larger the asymmetry.

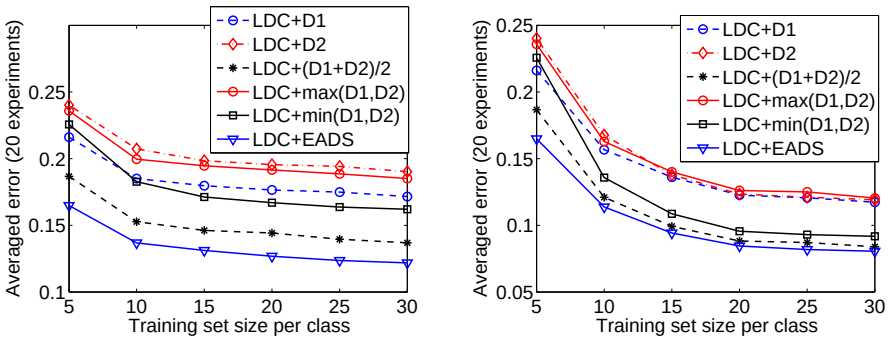
Dataset	# Classes	# Obj. per class	AC
ChickenPieces-35-45	5	117, 76, 96, 61, 96	0.08
Zongker	10	10×200	0.18
AjaxOrange	2	60, 1440	0.31
Winter Wren	2	109, 439	0.23

For each of the dissimilarity datasets, we evaluate the performances using asymmetric dissimilarity measures D_1 and D_2, the symmetrized measures (using minimum, average and maximum) and the EADS.

The classifiers compared are the linear discriminant classifier (LDA, but denoted LDC in our experiments) and the SVM, both in the dissimilarity space and implemented in PRTools [92]. For LDC we use regularization parameters $R = 0.01$ and $S = 0.9$, for SVM we use a linear kernel and a regularization parameter $C = 100$. These parameters show reasonable performances on all the datasets under investigation, and are, therefore, constant across all experiments and not optimized to fit a particular dataset.

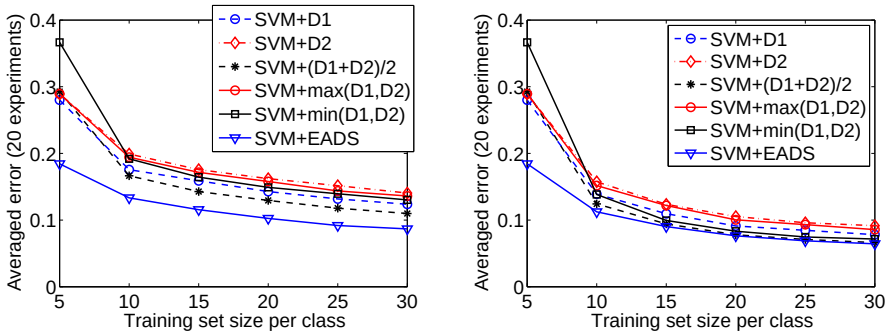
We provide learning curves over 20 runs for each dissimilarity / classifier combination, for increasing training sizes from 5 to 30 objects per class. In each of the learning curves, the number of prototypes is fixed to either 5 or 20 per class in order to explore the behavior with a small and a large representation set size. This means that the dimensionality of the dissimilarity space is the same for D_1, D_2 and the symmetrized versions, but twice as much for the EADS. The approaches compared are:

- DS resulting from the computation of dissimilarities in the direction from the objects to the prototypes (D_1).
- DS resulting from the computation of dissimilarities from the prototypes to the objects (D_2).
- DS resulting from averaging the dissimilarities in the two directions ($(D_1 + D_2)/2$).
- DS resulting from the maximum of the two dissimilarities ($\max(D_1, D_2)$).
- DS resulting from the minimum of the two dissimilarities ($\min(D_1, D_2)$).
- The extended asymmetric dissimilarity space (EADS).



(a) 5 prototypes per class

(b) 20 prototypes per class



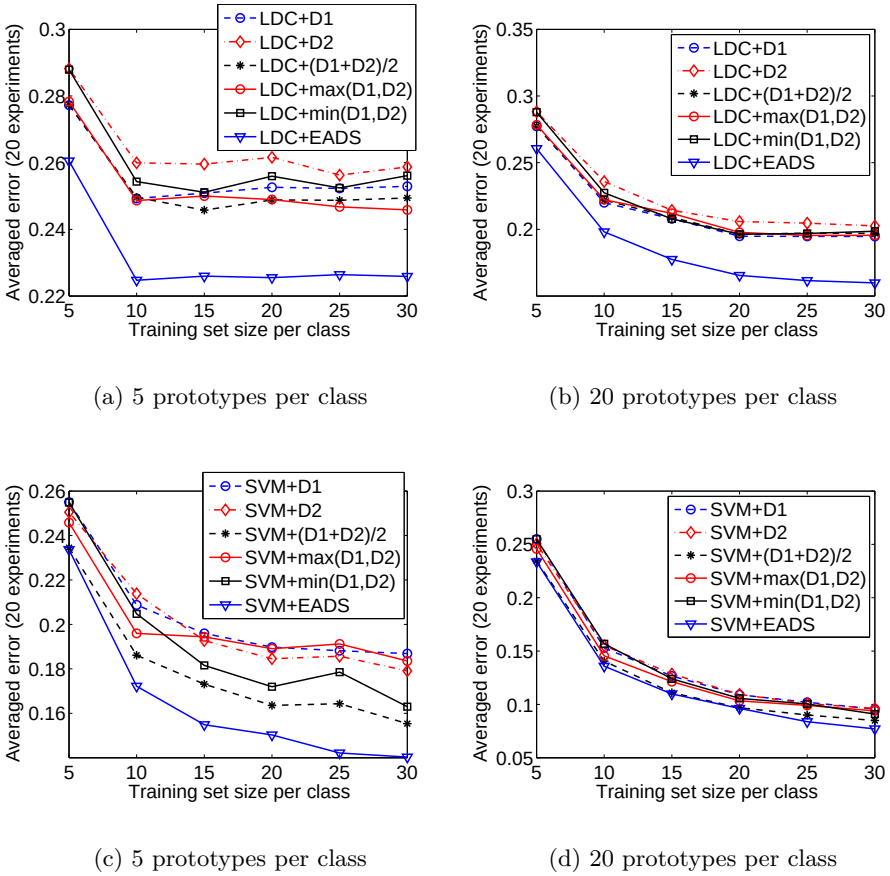
(c) 5 prototypes per class

(d) 20 prototypes per class

Figure 5.5: LDC and SVM classification results in dissimilarity spaces for Zongker dataset

5.2.10 Results and discussion

In Figs. 5.5 and 5.6 it can be seen from the results on the Zongker and Chicken Pieces datasets that the EADS outperforms the other approaches. This is especially true for a small number of prototypes (see Figs. 5.5 and 5.6 (a) and (c)). The results of the different approaches become more similar for the representation set of 20 prototypes per class, especially when SVM is used (see Figs. 5.5 and 5.6 (d)). The EADS is better than the individual spaces created from the directed dissimilarities, one explanation for this is that the directed dissimilarities provide complementary information so together they are more useful than individually. The EADS contains more information of the relations between the objects than an individual directed DS. The maximum operation is usually very sensitive to noise and outliers what explains its bad performance. The maximum dissimilarity makes objects belonging to the same class more different. These higher differences inside the class are likely to contain noise since objects of the same class should potentially be more similar. The average is more robust than maximum since it combines the information from both directed dissimilarities avoiding in some degree the influence of noise and outliers. Still, by averaging we may hamper the contribution of a very good directed dissimilarity if there is a noisy counterpart. The EADS may improve upon the average because the EADS does not obstructs the contribution of a good directed dissimilarity. The minimum operator is usually worse than EADS and averaging. One possible cause is that by using the minimum, the representation of all the objects is homogenized to some extent because for objects belonging to different classes the separability is decreased by selecting the minimum dissimilarity. Therefore, some discriminatory power is lost.



(a) 5 prototypes per class

(b) 20 prototypes per class

(c) 5 prototypes per class

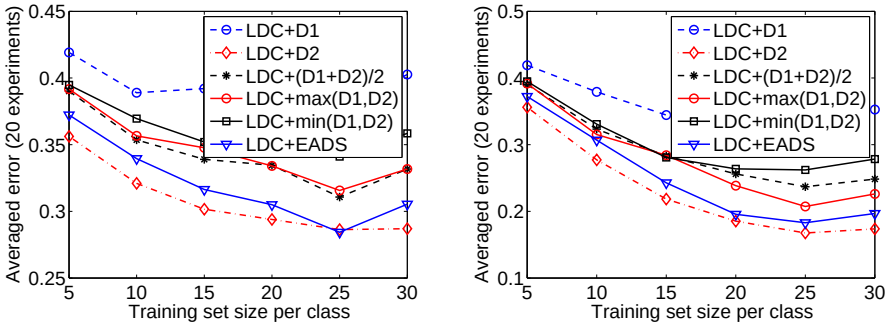
(d) 20 prototypes per class

Figure 5.6: LDC and SVM classification results in dissimilarity spaces for Chicken Pieces dataset

In AjaxOrange, it is an important observation that D_2 is more informative than D_1, especially for the LDC classifier (see Fig. 5.7 (a) and (b)). D_2 means that the dissimilarities are measured from the prototypes to the bags. The *meanmin* dissimilarity in (5.9) therefore ensures that, for a positive prototype, the positive instances (the AjaxOrange bottle) influence the dissimilarity value by definition, as all instances of the prototype have to be matched to instances in the training bag. Measuring the dissimilarity to positive prototypes, on the other hand, may result in very similar values for positive and negative bags because of identical backgrounds, therefore creating class overlap.

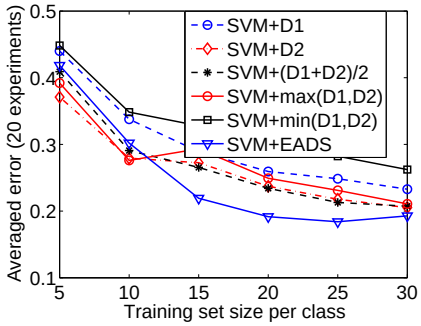
Because D_1 contains potentially harmful information, the combining methods do not succeed in combining this information from D_1 and D_2 in a way that is beneficial to the classifier. This is particularly evident for the LDC classifier (see Fig. 5.7 (a) and (b)), where only EADS has similar (but still worse) performances than D_2. For the SVM classifier, EADS performs well only when a few prototypes are used, but as more prototypes (and more harmful information from D_1) are involved, there is almost no advantage over D_2 alone.

From the results reported in Fig. 5.8 for Winter Wren, we again see that D_2 is more informative than D_1. However, what is different in this situation is that both directions contain useful information for classification, this is evident due to the success of the average, maximum and EADS combiners. The difference lies in the negative instances (fragments of other birds species, or background objects in the images) of positive bags. While in AjaxOrange, background objects are non-informative, the background in the audio fragments may be informative for the class of the sound. In particular, it is possible that some bird species are heard together more often: e.g. there is a correlation of 0.63 between the labels of Winter Wren and Pacific-slope

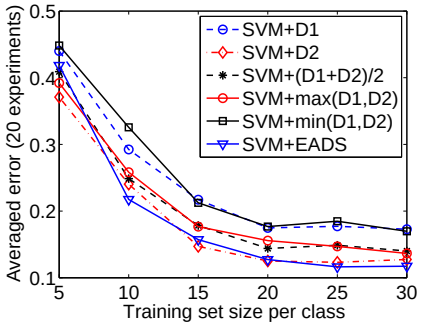


(a) 5 prototypes per class

(b) 20 prototypes per class



(c) 5 prototypes per class



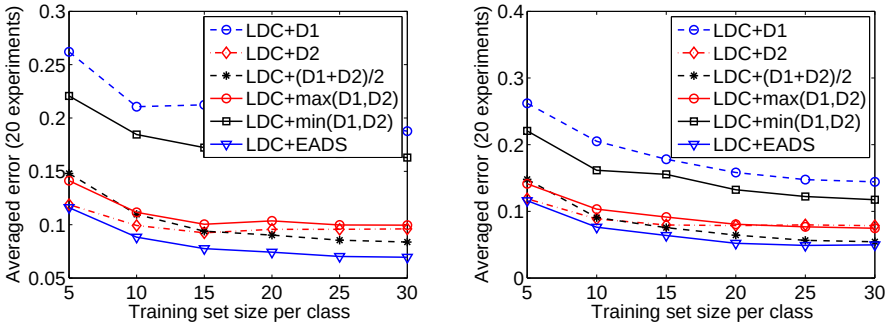
(d) 20 prototypes per class

Figure 5.7: LDC and SVM classification results in dissimilarity spaces for AjaxOrange dataset

Flycatcher. Therefore, also measuring dissimilarities to the prototypes produces dissimilarity values that are different for positive and negative bags.

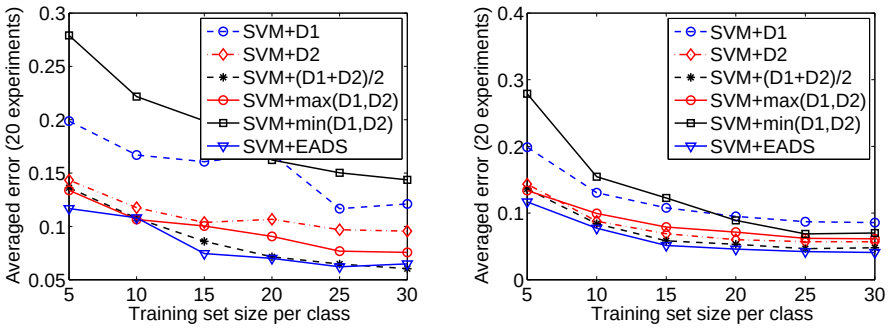
The increased dimensionality of the EADS is one of the main problems of this approach, as in small sample size cases the increased dimensionality may lead to overfitting. In order to overcome this, prototype selection can be considered. We developed other experiments using prototype selection for all the spaces compared. A fixed training set size of 200 objects was used, leading to spaces of dimensionality 5, 10, 15, 20 and 25. The choice to perform the selection of the prototypes was the forward selection optimizing the LOO 1-NN classification error in the training set. One example of standard and MIL dissimilarity datasets were considered: the Zongker and Winter Wren. Prototypes are selected for EADS as it is usually done for a standard DS. Prototypes using the two directed dissimilarities are available as candidates but the prototype selection method may discard one of the two or maybe both if they are not discriminative according to the selection criterion. The EADS is compared now with the other spaces on the basis of the same dimensionality.

From the results in Fig. 5.9 (a) it can be seen that, for the Zongker dataset, the best approaches are the EADS and the average. An interesting observation is that this dataset is intrinsically high-dimensional because the number of principal components (PCs) that retain 95% of the data variance is equal to 529. The average approach adds more information in each dimension since every dissimilarity encodes a combination of two. This implies that, for the dimensions considered that are small compared to 529, it performs as good as the EADS. On the contrary, the Winter Wren dataset is intrinsically low-dimensional, since the number of PCs retaining 95% of the data variance is equal to 3. This is a possible explanation of why the EADS



(a) 5 prototypes per class

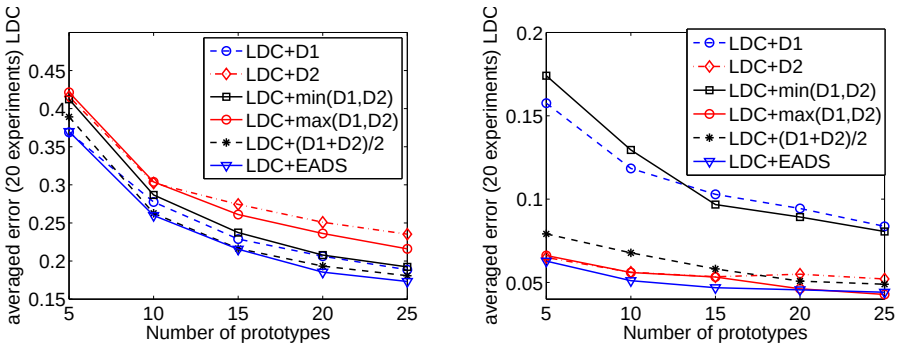
(b) 20 prototypes per class



(c) 5 prototypes per class

(d) 20 prototypes per class

Figure 5.8: LDC and SVM classification results in dissimilarity spaces for the Winter Wren dataset



(a) Zongker

(b) Winter Wren

Figure 5.9: Classification results after prototype selection for the Zongker and Winter Wren datasets

is the best in this case (see Fig. 5.9 (b)), because the average approach is likely to introduce some noise.

One interesting issue of using prototype selection in EADS is that not only the dimensions are decreased, but also the accuracy of the EADS itself may be improved especially in the datasets where one of the directed dissimilarities is the best and the other is very bad (e.g.

MIL datasets). The EADS without prototype selection in these cases may be worse than the best directed dissimilarity (see Fig. 5.7 (a) and (b)). However, by using a suitable prototype selection method in the EADS, only the prototypes from the best directed dissimilarity should be kept, and noisy prototypes from the bad directed dissimilarity should be discarded. This should make the results of the EADS similar to those of the best directed dissimilarity. This can be achieved if a proper prototype selection method is used. In the prototype selection executed for the Winter Wren, where one directed dissimilarity is remarkably better than the other, this can partially be seen. For example, in one run, the method selected 18 prototypes from the best directed dissimilarity in the set of 25 prototypes selected. Future work will include the study of suitable prototype selectors for EADS.

5.2.11 Conclusions

In this paper we study the EADS as an alternative to different approaches for dealing with asymmetric dissimilarities. The EADS outperforms the other approaches for a small number of prototypes in standard dissimilarity datasets, when both dissimilarities are about equally informative.

In MIL datasets, conclusions are slightly different because of the way the dissimilarities are created. It may be the case that the best option is one of the directed dissimilarities. However, if there is no knowledge on which directed dissimilarity is the best, the EADS may be the best choice. This especially holds when only a low number of prototypes is available.

It should be noted that the EADS increases the dimensionality as opposed to other combining approaches, therefore increasing the risk of overfitting. Prototype selection should be considered to keep the dimensionality low. After prototype selection, the EADS also shows good results in examples of intrinsically low- and high-dimensional datasets. However, for intrinsically high-dimensional datasets, averaging is also worth considering as combining rule.

Our main conclusion is that asymmetry is not an artefact that has to be removed in order to apply embedding or kernel methods to the classification problem. On the contrary, asymmetric dissimilarities may contain very useful information, and it is advisable to consider the dissimilarity representation as a means to fully use this information.

5.3 Reduced representation of multiscale non-metric data by prototype selection

This section will be submitted to the Signal Processing journal as: "Reduced representation of multiscale dissimilarity data by prototype selection", by Yenisel Plasencia Calaña, Yan Li, Robert P. W. Duin, Mauricio Orozco Alzate, Marco Loog, and Edel Garcia-Reyes.

Abstract

The representation of data under multiple scales provides the possibility of using more information for data analysis. However, by having more information, the computational cost of classification may potentially increase. In addition, if extended representations are used, other problems such as the "curse of dimensionality" and overfitting may occur. A further challenge is posed when the multiscale data is given in the form of non-metric dissimilarities since the standard approaches, which assume metric dissimilarities, cannot be applied. In this paper, we present a new approach to overcome these problems. We propose the Extended Multiscale Dissimilarity Space (EMDS) which takes into account only the "best" information from all the scales in a classification sense. Experimental results show evidence that our proposal, the reduced EMDS, outperforms the individual scales in terms of accuracy while maintaining the same computational cost. In addition, our approach outperforms the combination of the scales in the averaged dissimilarity space in terms of computational efficiency as well as in classification accuracy when the individual scales perform significantly different.

5.3.1 Introduction

Recently, the use of multiscale data in pattern recognition problems and specifically the use of multiscale dissimilarity data started to receive more attention from the scientific community [93–98]. The term multiscale refers to data represented at different scales or resolutions. This approach provides more information that, if used properly, contributes to improve the data modeling. However, the following question remains open: how to properly use multiscale dissimilarities for classification without increasing the computational cost? In the literature on supervised pattern recognition for multiscale data, we can find two main approaches: scale selection [98,99], and scale combining [96,97]. Scale selection has been tackled, for example, by Multiple Kernel Learning (MKL) [99,100], which is similar to the problem of selecting the best kernels for a given problem. For scale combining, all the different scales may be combined in the form of similarity or kernel matrices using, for instance, MKL as well [101].

There are general approaches that deal with a dissimilarity matrix; for instance, the k Nearest Neighbour classifiers (k-NN) directly applied to the matrix, and the classifiers in the Dissimilarity Space (DS) [3,11]. In the case of the k-NN, their outputs for the different scales can be combined to find the final decision, or we can decide to use the classifier on the best scale only. In the case of the DS, as proposed in [97], different classifiers can be trained for each individual DS related to a given scale, and the final classifiers are the ones with the best results. The advantage of this selection approach is that computational costs are proportionally diminished according to the final number of scales selected, however the information of the scales that are not selected is lost.

Another possibility to combine the scales is by computing a weighted average of the dissimilarities [97]. The disadvantage of this approach is its high computational cost since, for an incoming test object, the dissimilarities with all the objects in all the scales must be computed. This becomes very expensive for dissimilarity measures with a high computational complexity. Besides, it is not clearly established how to combine different dissimilarities. Another approach, to which little attention has been paid so far, is constructing an Extended Multiscale Dissimilarity Space (EMDS) from the dissimilarity matrices [102]. Despite the fact that the first results presented in [102] using the EMDS were discouraging, we consider that a smart selection of the set of prototypes can lead to better results.

We are not interested in combining classifiers computed for the different scales, which has been thoroughly studied in previous works [98,103–105], since we want to avoid the costs involved in computing the dissimilarity measures for all the scales. These costs may be too high, especially for the case of measures incorporating expert knowledge and invariances, e.g. when

matching shapes with different rotations and selecting the result for the best match to make the measure rotation invariant. In addition, this type of procedures usually assumes that the dissimilarity measure is metric while our goal is to develop procedures able to deal with any general dissimilarity measure. We also want to avoid the costs of submitting the data to multiple classifier systems which, in addition, require combining the outputs of the individual classifiers.

We are only interested in the case where the multiscale data is provided by an expert in the form of (possibly non-metric) dissimilarity matrices for each scale, where standard approaches, which assume distances instead of general dissimilarities (e.g. asymmetric or disobeying the triangle inequality), cannot be applied. Even methods based on kernels [99, 100] cannot be applied without modifying the original proximities since only for a Euclidean distance matrix, a positive definite kernel that preserves the distances in some space can be obtained (e.g. by classical scaling [3]). We also discard the use of k-NN classifiers directly applied to the matrices since they are computationally expensive. In addition, k-NN classifiers restrict classification to this family, while many problems, for example those with small sample sizes, may benefit from other classifiers such as linear ones. However, the DS seems to be a reasonable option for the classification of multiscale dissimilarity data since it is fast, it provides the possibility of using a plethora of classifiers and not only the k-NN ones, and it allows to explore in depth the EMDS as a new possibility for combining multiscale information. Due to the mentioned drawbacks of the other approaches, we will focus on the EMDS approach.

In this paper, we propose a new approach to represent potentially non-metric multiscale dissimilarity data based on a reduced EMDS, which is created after supervised prototype selection by a Genetic Algorithm (GA) [106–108]. The reduction is performed in a way that the most important information from all the scales is preserved using the most informative prototypes according to a supervised criterion. In our approach, a smart compromise is obtained between scale selection and scale combination. This approach is capable of outperforming the combination of all scales by averaging because the latter may result in more information loss. This can be explained intuitively since, for instance, a very good prototype in one scale may be a very bad prototype in another scale for several reasons (e.g. the scale was not appropriate to provide discriminative information for the problem under consideration, the scale magnified the noise present in the object measurements, etc). Consequently, by averaging the dissimilarity values, the contribution of the good prototype is lost. We avoid this by selecting the best prototypes per scale and using their information without modifications. Moreover, with our approach, less distance computations are needed, since, for new incoming objects, the dissimilarities with the selected prototypes are not computed in all the scales as in the averaging approach. Our approach may also outperform the selection approach due to the fact that the information from all the scales is used.

The remaining part of the paper is organized as follows. Subsection 5.3.2 introduces the extended multiscale dissimilarity space. Subsection 5.3.3 presents the related work on prototype selection. Subsection 5.3.4 presents the description of the proposed GA with the supervised selection criterion. Subsection 5.3.5 presents the data, experimental setup, results and analysis. Conclusions are drawn in Subsection 5.3.7.

5.3.2 Extended multiscale dissimilarity space

The DS was proposed by Pekalska and Duin [3] as an alternative to represent dissimilarity data. The DS is an adequate option to handle measures computed from matching processes or measures incorporating expert knowledge that are non-Euclidean or even non-metric. In this approach, it is not needed that dissimilarities are Euclidean themselves, because they are interpreted as coordinates in the dissimilarity space. In the DS we can apply all the statistical pattern recognition procedures that are suited for Euclidean spaces.

Let X be the space of objects in consideration which may not be a feature vector space but a more complicated one such as a graph space or other nonstandard one. A set of prototypes $R = \{r_1, r_2, \ldots, r_l\} \in X$, also called representation set, is used for the creation of the DS. A training set $T = \{x_1, x_2, \ldots, x_n\} \in X$ is represented in the DS by the dissimilarities of objects in T with objects in R. In general, for a representation set of l prototypes, and a suitable dissimilarity measure for the problem $d : X \times X \to \mathbb{R}_0^+$, we obtain a dissimilarity matrix $D(T, R)$; the mapping to a DS is represented as $\phi_R^d : X \to \mathbb{R}^l$. The representation of an object x in the DS is the vector of its dissimilarities with the prototypes:

$$\phi_R^d(x) = [d(x, r_1)\ d(x, r_2)\ \ldots\ d(x, r_l)]. \tag{5.11}$$

In case of multiscale data, dissimilarities are computed for each scale independently. If we have ten scales, then we obtain ten different dissimilarity matrices. Classifiers can be computed in the individual DS for each scale. In the experimental comparison in [97], it was found that this provides worse results than classification in the DS constructed by straightforward averaging of all scales. However, when using the average, we have the problem of the high computational cost because dissimilarities with prototypes from all the scales must be measured.

Our focus is in the construction of an EMDS. The extended space representation is created from the individual representations in a DS for each scale. For a multiscale problem with M scales, denoting $D_m = D_m(T, R)$ the dissimilarity matrix computed for scale m, we have $D_1, D_2, \ldots, D_M$, normalized dissimilarity matrices. The representation of training objects in the EMDS is created by the concatenation of the individual dissimilarity matrices for each scale: $[D_1\ D_2\ \ldots\ D_M]$. The embedding of any object is obtained by the mapping $\Theta_R^d : X \to \mathbb{R}^{lM}$, which returns the vector of the dissimilarities with the prototypes from all the scales:

$$\Theta_R^d(x) = [d(x^1, r_1^1) \ldots d(x^1, r_l^1) \ldots d(x^M, r_1^M) \ldots d(x^M, r_l^M)], \tag{5.12}$$

which can be rewritten as:

$$\Theta_R^d(x) = [\phi_{R_1}^d(x^1)\ \phi_{R_2}^d(x^2) \ldots \phi_{R_M}^d(x^M)], \tag{5.13}$$

where $R_m = \{r_1^m, r_2^m, \ldots, r_l^m\} \in X_m, m = 1...M$, is the representation set in scale m and X_m the space of objects for scale m; $x^m \in X_m, m = 1...M$, are the representations of x under the different scales. In the framework of dissimilarity spaces, there are at least three possible approaches to exploit multiscale data given in the form of dissimilarity matrices:

- selection: $f(D_1, D_2, \ldots, D_M) = D_j$
- combination by weighted averaging: $g(D_1, D_2, \ldots, D_M) = \sum_{i=1}^{M} \alpha_i D_i$
- extension: $h(D_1, D_2, \ldots, D_M) = [D_1\ D_2\ \ldots\ D_M]$

It can be seen that the extended space has a dimensionality higher than the dimensionality of the other approaches. This dimensionality increases in proportion to the number of scales. This poses a need of prototype selection in the EMDS, in order to decrease the dimension of the space and the computational effort for classification. However, this must be accomplished in a smart way, in order to take advantage of the multiscale information.

5.3.3 Related work on prototype selection

The representation of objects in the EMDS is created by a concatenation of their representations in individual DSs related to different scales. Therefore, the main problem with the EMDS is its

high dimensionality. It is a cause of overfitting and the "curse of dimensionality" phenomenon. The term overfitting refers to the fact that the classifier describes the noise in the data instead of important underlying information. Due to this, the classifier cannot generalize well to unseen data. The "curse of dimensionality" states that the number of objects needed to train a classifier often grows exponentially with the number of dimensions. Another problem is the increase of the computational costs involved in classification. To avoid these problems in a DS, prototype selection methods have been studied [11, 21].

In order to be able to use the multiscale information avoiding a high dimensionality of the EMDS, a prototype selection must be performed to create a reduced EMDS. The only selection method which converges to the global optimum solution is the exhaustive search, but it is computationally unfeasible. In addition, the EMDS presents different conditions compared to an standard DS. Therefore, we have to discard promising prototype selection procedures that work directly with the dissimilarity information such as the KCentres and ModeSeek proposed in [11], and the Farthest First Transversal (FFT) [16], unless they are applied on a single scale. These methods require a direct comparison of the prototypes being analyzed, which in the EMDS case may belong to different scales and, thereby, are not comparable. Another good method, the Forward Selection (FS) [11], is not adequate for the EMDS due to the high dimensionality of this space which is proportional to the number of prototypes and scales. The most appropriate option that we found is the selection of prototypes by GAs optimizing as criterion a classification error in the DS.

We consider that GAs are specially suitable for prototype selection in dissimilarity representations, since, similar objects represent similar information and they can be chosen indistinctively as prototypes and, therefore, a thorough search is not needed. GAs have been used in similar problems such as feature selection [108] or prototype selection for k-NN classification [39, 40]. The GA for prototype selection in a DS was proposed in [20], where it showed a good performance in standard DSs of moderate dimensionality. However, its performance for very high dimensional spaces such as the extended ones has not been studied.

5.3.4 Proposed method

We propose to select the prototypes in the EMDS taking into account information provided by all the scales. In this way, we avoid the problem of the correlation or non-representativeness of the prototypes. We propose a GA optimizing a 1-NN leave one out (LOO) classification error in the EMDS for a validation set. For simplicity, the validation set coincides in our case with the training set but it may be also different. The "1-NN error" criterion is computed in the DSs for each candidate set of prototypes. The selected set of prototypes will be the one leading to the smallest error. The criterion to be minimized can be formulated as follows:

$$\jmath = \sum_{x_i \in T} CE(x_i),$$

$$CE(x_i) = \begin{cases} 1, & \lambda_T(x_i) \neq \lambda_T(x_k) \\ 0, & \lambda_T(x_i) = \lambda_T(x_k) \end{cases}, x_k = \underset{x_j \in T \setminus \{x_i\}}{\operatorname{argmin}} \; d(x_i, x_j) \tag{5.14}$$

where $\lambda_T(x_i)$ and $\lambda_T(x_k)$ are the class label of x_i, x_k respectively, and x_k is the object with minimum Euclidean distance to x_i in the DS. The criterion $\jmath$ is therefore the 1-NN classification error on the validation set T in the dissimilarity space using the LOO approach. This criterion takes both the multiscale and the labels information into account since the classification error is computed for EMDS with different sets of prototypes coming from different scales. It is also a relatively fast criterion compared to other criteria that take into account class separability such as the Mahalanobis distance.

The GA is an evolutionary method which uses heuristics to converge to better solutions, resembling biological processes such as reproduction and mutation. Each possible solution (individual, chromosome) is a set of prototypes of fixed cardinality l codified in a $l-tuple$ of prototypes indexes. The GA starts the search in an initial population of individuals. In each evolution cycle, the GA evaluates the population using the fitness function which in our problem corresponds to the proposed supervised criterion. The population undergoes reproduction (with best fitted individuals) and mutation processes until criteria are met. As reproduction and selection strategies we use uniform crossover and elitist selection. The sub-optimality of the GA may not be as problematic as in feature selection problems since similar objects may have similar properties as prototypes and they can be chosen indistinctively. Moreover, for large spaces such as the EMDS, this is a good alternative in terms of computational time. The corresponding pseudo-code is presented in Algorithm 4.

Algorithm 4: Genetic Algorithm for prototype selection minimizing LOO 1-NN error as fitness function

Input: D: dissimilarity matrix among samples and candidates to prototypes; k: desired number of prototypes, S: number of individuals in the population, rp: reproduction probability, mp: mutation probability, $iter$: number of generations

Output: bestindividual: set of prototypes indexes

```
// randomly generate the population
P ← GenerateInitialPopulation(D, k, S);
bestindividual ← P[1];
// find the best solution from the population and assign it to
   bestindividual
foreach currentindividual in P do
    critvalcurrent ← FitnessbyNNError(currentindividual, D);
    critvalbest ← FitnessbyNNError(bestindividual, D);
    if critvalcurrent < critvalbest then
        bestindividual ← currentindividual;
    end
end
while number of generations < iter do
    // Evolution cycle
    foreach currentindividual in P do
        // Reproduction, replace a gene of currentindividual with
           probability rp by a gene of the best
        Reproduce(bestindividual, currentindividual, rp);
        // Mutation, change a gene of currentindividual with probability
           mp
        Mutate(currentindividual, mp);
    end
    // find the best solution from the population and assign it to
       bestindividual
end
```

5.3.5 Data and experimental setup

In this section, the multiscale data sets used in our experiments are described. The different approaches for prototype selection are presented. The experimental setup, results and discussion are also provided.

Three different multiscale data sets were used in the experiments. They are the Colon, Texture and Chicken Pieces data sets. Their descriptions are given below:

Colon. This data set [109, 110] represents colon tissue data; it was provided by Dr. Marius Nap from the Atrium Medical Center in Heerlen, The Netherlands. The objects are microscope image patches of size 1024×1024 belonging to four classes: normal, inflamed, adenomatous, and cancer. The Laplacian of different scales was applied to each image patch, and the city-block (L1) distance between the histograms of the response images was used as the dissimilarity measure.

Texture. This is the Brodatz texture data set downloaded from [111]. It has 111 images that we consider as classes. The 640×640 images were partitioned into 9 subimages that are used as class objects, each having a size of 210×210. The Leung-Malik [112] filter set at different scales was applied to the images by us, and the Chi square distance between the histograms of the response images was computed.

Chicken Pieces. This data set is computed from the Chicken Pieces image data set [43]. The images are in binary format representing silhouettes from five different parts of the chicken: wing (117 samples), back (76), drumstick (96), thigh and back (61), and breast (96). From these images the edges are extracted and approximated by segments of different pixel length, and string representations of the angles between the segments are derived. A set of resolutions for the string representations is used. The dissimilarity matrix is composed by edit distances between these strings. A description of the data sets is presented in Table 5.3.

In the experiments, the reduced EMDS obtained after performing the prototype selection is

Table 5.3: Properties of the multiscale datasets, the last column ($|V|$) refers to the validation set cardinality used for the experiments.

Data sets	# Classes	# Obj	# scales	$\|V\|$ in EMDS
Colon	4	375×4	9	100×9
Texture	111	9×111	6	222×6
Chicken Pieces	5	446	11	170×11

compared to the other multiscale approaches, the space of averaged multiscale dissimilarities, and the individual spaces for each scale, always using the same dimensionality. For consistency, we compare the same Linear Discriminant classifier (LDC), which is the Bayes classifier assuming normal densities with identical covariance matrices, and the 1-NN in the different spaces and data sets. All the dissimilarity matrices were normalized to avoid scaling problems.

Since the datasets present a small size, they were 20 times randomly divided into two sets: a training set, that was used for selecting the extended prototypes in the different scales, to optimize the selection criterion and to build the final classifiers in the EMDS, and a test set, which was only used to compute the final classification error. The prototype selectors executed are:

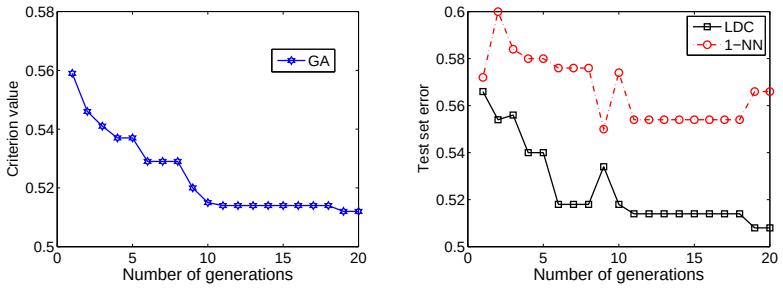
- GA in the EMDS
- random selection in the EMDS
- GA in the averaged DS
- random selection in the averaged DS

- random selection in the individual DS for each scale

Note that the prototype selector used in the comparison with the GA is the random selection because, as we discussed at the end of Sec. 5.3.3, many procedures rely on a square (all vs. all) dissimilarity matrix, and others are too expensive for large spaces. The random selection resamples the space and performs well for large representation sets [11]. Different parameters have been proposed for the GA [106, 113]. However, they can be problem-dependent, thereby we decided to use parameters proposed in previous works on prototype selection [21]:

- Initial population: 30 individuals or solutions
- Probability of reproduction: 0.5
- Probability of mutation: 0.05
- Stopping condition: 20 generations reached

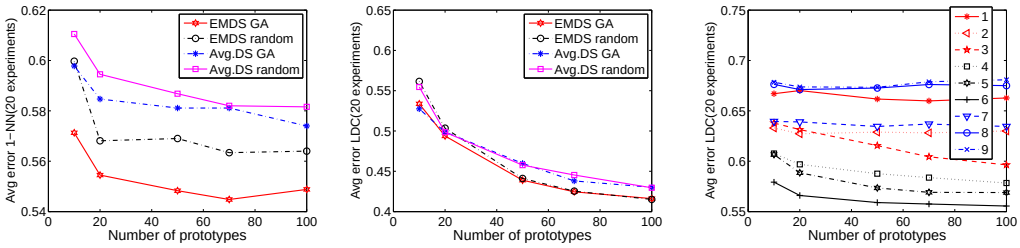
5.3.6 Results and discussion



(a) Criterion values or fitness function values for increasing number of GA generations

(b) Test set errors for increasing numbers of GA generations

Figure 5.10: GA criterion convergence and associated test set errors when selecting ten prototypes

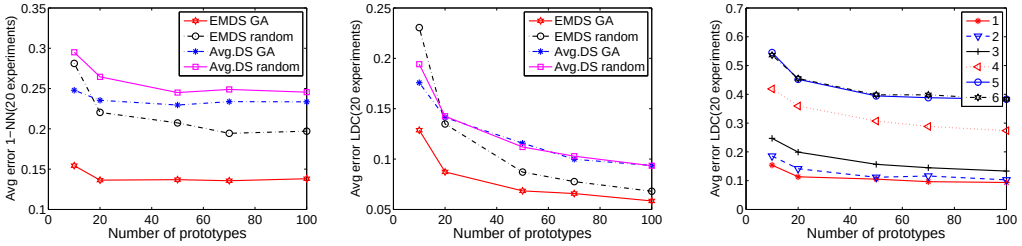


(a) Nearest neighbour classifier, 250 training objects per class

(b) Linear Discriminant classifier, 250 training objects per class

(c) Linear Discriminant classifier on individual scales, 250 training objects per class

Figure 5.11: Classification results on the extended (EMDS), averaged (avg. DS) and individual dissimilarity spaces for Colon data set

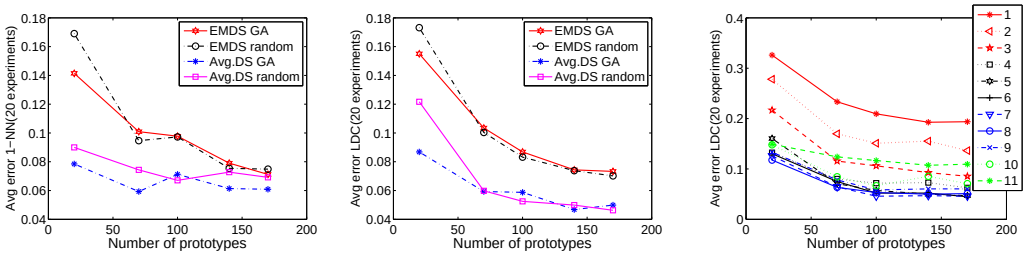


(a) Nearest neighbour classifier, 6 training objects per class

(b) Linear Discriminant classifier, 6 training objects per class

(c) Linear Discriminant classifier on individual scales, 6 training objects per class

Figure 5.12: Classification results on the extended (EMDS), averaged (avg. DS) and individual dissimilarity spaces for Texture data set



(a) Nearest neighbour classifier, 350 training objects

(b) Linear Discriminant classifier, 350 training objects

(c) Linear Discriminant classifier on individual scales, 350 training objects

Figure 5.13: Classification results on the extended (EMDS), averaged (avg. DS) and individual dissimilarity spaces for Chicken Pieces data set

Figure 5.10 presents the evolution of the criterion value and the related test set errors obtained for each generation. It can be seen that the criterion is effective for improving the classification results of a test set. However, it is clear that the the test set errors curves are not smooth because a perfect monotonic relation between criterion values and test set errors is rather difficult to obtain for a genetic algorithm which may fall into local optima. The small or moderate sample sizes worsen the issue since the 1-NN classifier used as criterion may overfit the data.

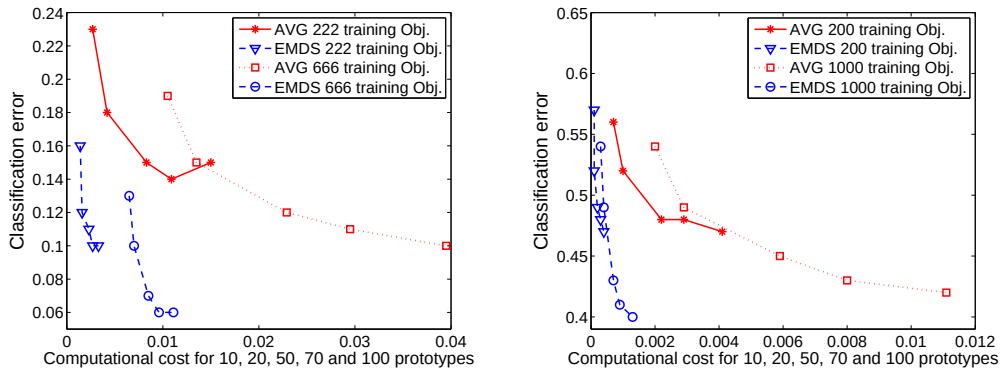
Figures 5.11-5.13 present the results obtained for the data sets used in our study. Classification errors are presented for increasing numbers of prototypes in multiscale spaces as well as in the individual spaces from the different scales. Standard deviations were not included to maintain the clarity of the plots, but they vary between 0.02 and 0.05 for Chicken Pieces, 0.01 and 0.03 for Colon, and between 0.007 and 0.05 for Texture data set.

In Fig. 5.11, for the Colon data set, it is possible to see that the EMDS outperforms the averaged DS and the individual scales. Selection by GA achieves better results than random selection for the 1-NN classifier. It is worth noting that after creating the spaces with the same cardinality and training the classifiers, classification in the EMDS is much cheaper than classification in the averaged DS, since the dissimilarities of a new test object must be measured with the prototypes from all the scales in the averaged DS.

Results for the Texture data set show a clear example where selection of prototypes by the

GA in the EMDS provides remarkably better results than the other approaches (Fig. 5.12). Here we can see a recurrent phenomenon, the GA selection is more beneficial for the 1-NN than for the LDC. This can be expected, since the 1-NN classifier is more sensitive to noise and outliers than the LDC, therefore it benefits more from a careful selection of the prototypes. The difference of results in the EMDS and averaged DS is less substantial for the LDC, which is the best performing classifier. The EMDS is still better than the averaged DS. The EMDS with prototype selection handles better an imperfect selection of the classifier. In this dataset as well as in the Colon dataset, the EMDS significantly improves the results of the individual scales.

Results for the Chicken Pieces data set in Fig. 5.13 show an opposite behaviour. The averaged DS outperforms the EMDS. Our explanation is that, in this dataset, only four scales present a decreased classification performance while seven scales perform similarly well. These large number of good performing scales influence the average dissimilarity computation heavier than the four worse ones. Therefore, the final averaged space behaves similar to the best scales. However, for the Colon and Texture datasets, the individual scales perform significantly different from each other, and the averaged space suffers from this while the reduced EMDS is able to capture the important information for classification. The GA selection is beneficial for both spaces.



(a) Texture data set

(b) Colon data set

Figure 5.14: Classification error vs computational time in seconds for classifying a new object for the extended multiscale spaces and averaged spaces when different numbers of prototypes are used

indent Fig. 5.14 shows the errors vs. computation times of classifying a new object while varying the number of prototypes for the EMDS and the averaged space. Results are shown for the Colon and the Texture data sets only since, for the Chicken Pieces, we did not have access to the code for computing the dissimilarities. The results are shown for the best prototype selector in each space varying the number of prototypes. It can be seen that the computation times for the EMDS are much smaller than the ones for the averaged space, while the classification accuracy is similar or sometimes better.

Table 5.4 shows the decreasing ranking of the scales according to the number of times prototypes of that scale were selected by the GA in the different data sets. Table 5.5 shows an example of the number of times that prototypes in the first 6 most significant scales were selected as a part of a solution for Colon dataset. It can be seen that whether the best scales are the first, the middle or the last depends on the data. For the Texture and Colon data set,

Table 5.4: Decreasing ranking of scales according to the number of times that prototypes of each scale were selected

Colon dataset	scales										
10 prototypes	4	5	2	6	7	3	9	8	1		
100 prototypes	4	7	6	2	3	9	1	8	5		
Texture dataset											
10 prototypes	1	2	3	4	6	5					
100 prototypes	1	2	3	6	4	5					
Chicken Pieces dataset											
20 prototypes	9	1	6	8	5	7	3	10	11	4	2
170 prototypes	9	10	5	4	1	6	7	2	11	3	8

Table 5.5: Best 6 scales (sorted by decreasing order) and number of times (specified in brackets) that prototypes of these scales were selected for Colon dataset

10 prototypes	4(26)	5(25)	2(24)	6(23)	7(23)	3(22)
100 prototypes	4(259)	7(237)	6(233)	2(226)	3(225)	9(219)

the best scales are the first and the middle ones respectively. For the Chicken Pieces, the last and middle scales are the best ones. An interesting issue is that the first scales selected in each data set are not only the ones with the smallest classification errors (see Figs. 5.11-5.13 (c)). Scales with a high classification error such as scale 7 for Colon or scale 1 for Chicken Pieces are also selected in the first positions. It means that individually they may not be good but, when combined, they provide complementary information that is useful for classification. A tendency of the larger scales to be more stable in their rankings can be seen for smaller and larger numbers of prototypes.

5.3.7 Conclusions

In this paper, we proposed a new approach to cope with potentially non-metric multiscale dissimilarity data. The multiscale data is represented in a reduced EMDS considering the best prototypes that contribute with complementary information from the given scales. A GA with a supervised criterion was proposed to accomplish the selection of the set of prototypes.

The classification results in the proposed EMDS outperform the results in the individual scales for the same dimensionality. This means that the approach is as affordable as approaches that deal with data in only one scale and at the same time provides better classification results. In addition, the proposed approach outperformed the combination of the scales by averaging when the individual scales provide significantly different information. Therefore, despite being less computationally expensive, our approach is able to outperform the combination of scales by averaging in the mentioned scenario.

In addition, our approach provides better interpretability of the selected prototypes since they provide information both about the object and the scale where it was important. This type of information is useful for experts, therefore the selection of prototypes in the EMDS presents a value beyond classification. This is in contrast to the average approach since it is not possible to determine in which scales the selected prototypes are more representative. When the majority of the scales perform similarly well, the averaged DS is the best option in a classification sense.

The scales selected by the GA are different for each problem. In general, all the scales contribute with some information in the EMDS, even the ones that when used individually have a poor performance. One limitation of this approach is that despite the fact that we exploit

the multiscale information we are still limited by the fact the we use objects as prototypes. If models are used as prototypes instead of objects, and they are combined with the multiscale information, the results could be further improved.

Chapter 6

Discussion

6.1 Conclusions

In the thesis, we studied the selection of prototypes for classification of data in the dissimilarity space. The main question which guided our study was:

- Can we create better prototypes and/or selection procedures if we take the nature and characteristics of the dissimilarity data into account in the process?

Our general hypothesis was that by taking into account the nature and spatial distribution of the dissimilarity data we can obtain better prototypes, faster procedures to select them, and improved DS in the sense of better compromising between accuracy and efficiency of classification. From our research, we found that this is true, especially when selecting small numbers of prototypes.

We found that the prototype selection procedures which make use of the nature of the data, especially of the dissimilarity character, are fast and accurate for the selection of a good representation set. In addition, the prototype models which reflect the spatial distribution of objects outperformed previous approaches based on objects as prototypes such as the KCentres, Forward Selection (FS) and Genetic Algorithms (GAs). Finally, we found that taking into account the nature of asymmetric and multiscale dissimilarities in the definition of the extended prototypes and combined with selection methods provides more successful compact representations than when ignoring their specificities. However, we found that there is no "best performing" prototype selection method, it depends on the nature of the problem and data characteristics. We will return on this in the next section which presents different type of problems (two-class, multi-class) and data characteristics (small sample sizes, large sizes, spatial distributions, linear, non-linear, overlapped classes, asymmetric, multiscale).

From the work developed in Chapter 3 we found that GAs offer a promising alternative to other selection methods based on objects as prototypes since they are able to find complicated representation sets. We found that these algorithms can intrinsically exploit the knowledge on our problem, leading to a fast convergence to good solutions for any cardinality of the set of prototypes. They are especially suitable for selecting prototypes out of very large candidate sets. This is in contrast to the FS which is computationally expensive especially for large candidate sets. The fact that neighbouring or nearby objects present a similar representational power makes the GA more appropriate than FS procedures to accomplish the selection, since, any of two close objects can be selected interchangeably. However, the FS needs a thorough evaluation which may be needed for feature selection, but is certainly not needed for prototype selection since we have the information (in the dissimilarities) of which prototypes are neighbours.

All the interesting properties of the GAs and specifically the good trade-off it finds, led us to propose new GA-based scalable methods for prototype selection in section 3.2. To achieve

this, two scalable criteria were proposed and tested on large scale datasets. We found that the unsupervised criterion based on maximizing the length of the minimum spanning tree (and therefore the diversity of the prototypes) is very fast compared to other criteria and provides good representation sets especially for multiclass problems. The proposed supervised GA for prototype selection, combined with the use of a large set of objects to compute the criterion, is beneficial for data with a significant overlap among the classes.

One conclusion of this thesis from the study developed in Chapter 4 is that knowledge about the spatial distribution of objects (specifically elongated or cigar-like distributions and clusters) is beneficial to select proper models to be used as prototypes. Note that this knowledge is obtained only from the dissimilarities since we assume there is no underlying vector space available. This knowledge can be acquired by analyzing the dissimilarity values using different techniques: from a visual inspection of the MDS plots of the dissimilarity data, from the intrinsic dimension of the data, and even from the classification error of the 1-NN classifier. This may help users to decide if some particular model is suitable for the data at hand. Selection methods must be used to discard noninformative or even harmful prototypes. To achieve this for a generalized dissimilarity representation by feature lines, we proposed a feature lines selection criterion which demonstrated its usefulness in handling several types of data distributions including elongated datasets.

In the case that it is known in advance that the dataset to be analyzed presents clusters, the selection of cluster-based prototypes is a good alternative. Any of the available classifiers for feature spaces can be applied on the DS, and specifically in the generalized DS by clusters. We found that cluster prototypes are able to unveil better nonlinear structures and make the data more linearly separable in dissimilarity spaces. Also object prototypes do this, but clusters achieve higher linearity of data for the same dimension. This is similar to what can be achieved by kernels and SVM. However, as it was originally conceived, the kernel-based SVM cannot handle the non-Euclidean nature of general dissimilarity data while the mapping defined by cluster distances overcomes this limitation of SVM. In general, in Chapter 4, we propose to create additional information using previous knowledge on the spatial distribution of data to cope with the prototype selection problem.

From Chapter 5 we found that extended dissimilarity spaces allow one to properly combine prototypes with differently measured dissimilarities. In the case of asymmetric dissimilarity datasets the prototypes are defined by using the best performing directed dissimilarities. We found that this approach is more beneficial for classification than previous approaches that ignore the directed asymmetric dissimilarities. In the case of multiscale dissimilarity data, the best prototypes are devised from all the scales. Despite the fact that the extended multiscale spaces by using all the candidate prototypes are very high dimensional, it is shown that with a smart selection of the prototypes the dimension of the space can be dramatically decreased while maintaining the benefits provided by the multiscale information. Prototype selection is proposed such that the best performing dissimilarities (e.g. the best directed asymmetric dissimilarity or the best dissimilarity information from the scales) to prototypes are maintained while the dimensionality of the DS is reduced. The obtained results suggest that combining dissimilarities in this way is a good alternative. In Chapter 5, we use additional information that already exists trying to maintain the specificities of this information in contrast to previous approaches (e.g. averaging the dissimilarity values).

Note that throughout this thesis we do not perform the prototype selection procedures on a per class basis. Instead, our proposal is to perform the selection on the whole dataset. We think this is beneficial because we implicitly deal with imbalanced problems and for many of our procedures it is not mandatory to have a perfect sampling of the data. Therefore, in many examples, our procedures can even deal with classes which are distributed differently, e.g. the GA+MST. This does not mean that class label information is not important. On the contrary,

for complicated problems we found that it is important to select prototypes which are more close to their true classes than to impostor classes.

6.2 Guidelines

We introduce a set of guidelines which can help researchers to decide which approach is suitable for their specific datasets. In addition, we advice the use of linear discriminant classifiers (LDC) or quadratic discriminant classifiers (QDC) after the data is mapped to a dissimilarity space of sufficiently low dimensionality.

1. *Linearly separable two-class datasets.* Select two or just few prototypes with GA+MST or GA+sup from section 3.2.

2. *Two-class datasets with small sample sizes and elongated distributions or overlapped classes.* Select two or just few feature lines with method proposed in section 4.1.

3. *Nonlinearly separable two-class datasets (e.g. concentric rings, or one class surrounded by the other).* Select few clusters using minimum distances as in section 4.2.

4. *Multi-class datasets with similar ball-shaped class distribution and separable classes.* Select an appropriate number of prototypes by intrinsic dimension estimation, and perform selection with GA+MST or GA+sup from section 3.2, FFT from section 2.2 can also be used. Another option is to use cluster prototypes with minimum or average distances to clusters as in section 4.2 in case a small dimensionality of the dissimilarity space is required sacrificing time of dissimilarity vectors computation.

5. *Multi-class large datasets with similar ball-shaped class distribution and separable classes.* Select an appropriate number of prototypes (depending on the intrinsic dimensionality of the data + computationally feasible according to the user needs), and perform selection with GA+MST or GA+sup from section 3.2.

6. *Multi-class (potentially large) datasets with similar ball-shaped class distribution and overlapped classes.* Select an appropriate number of prototypes, and perform selection with GA+sup from section 3.2.

7. *Multi-class (potentially large) datasets where classes are separable and present different spatial distribution or are not well sampled.* Select appropriate number of prototypes, and perform selection with GA+MST from section 3.2.

8. *Asymmetric dissimilarity datasets.* Perform prototype selection in the extended asymmetric dissimilarity space as in section 5.1 and 5.2 especially if the asymmetry coefficient is large.

9. *Multiscale dissimilarity datasets.* Perform prototype selection in the extended multiscale dissimilarity space as in section 5.3 when scales perform significantly different or perform the selection on the averaged dissimilarity space otherwise.

6.3 Open issues

A promising direction for future studies is the creation of new approaches which are able to exploit the accurate information that can be extracted from large datasets such as the data distribution. Approaches such as GAs or deep learning could be used to find accurately the best prototypes that can be constructed by combining different models of the original objects. The final set may contain prototypes coming from different types of models which are learnt from the data instead of being handcrafted using the previous knowledge on the problem (see chapter 3).

A more fundamental open issue is whether supervision is really needed for selecting the prototypes, and, if so, in which cases it is more profitable. Supervision usually poses an extra computational cost and requires labeled sets.

The main aspects that require more research in the creation and selection of clusters are: 1) the sensitivity to the clustering procedure and 2) the sensitivity to the sizes and number of clusters. The best choice seems to be different depending on the data characteristics. A smarter way for measuring the dissimilarity with clusters should be developed. Especially, since the minimum distances usually perform well, some substructures (e.g. smaller clusters) may be identified inside each cluster to represent it. This will have a positive impact on the computational cost of the procedure since it will decrease the number of dissimilarities to be measured. Another interesting topic is how to select the clusters. In this thesis we used as criterion the minimization of the classification error found by the 1-NN classifier. However, provided that we found that the clusters distances create dissimilarity spaces where the data is better linearly separable than in the original space, the optimization of a classification error computed by linear classifiers may improve the results even further. Another issue that remains open is whether taking into account the negative part of subspaces may improve the representation based in subspace distances. A possible way to incorporate the distances to the negative part of the subspace into the representation is by concatenating the positive and the negative parts in an extended representation.

In the proposed extended asymmetric dissimilarity spaces, one open question is: when it is useful to apply the approach? It was found that it may be useful for shape matching incorporating expert knowledge and invariances where the two directions are about equally informative, and sometimes for multiple instance learning. In addition, we found that the larger the asymmetry coefficient, the higher the improvements we obtain by resorting to the extended space. However, a deeper understanding is needed to find out when the use of asymmetry is a better alternative than the standard symmetrization methods. Our intuition is that the cause of the asymmetry plays a crucial role to define its importance for classification.

The creation of extended multiscale dissimilarity spaces also poses some challenges. How to make the approach more suitable for data with similarly performing scales? Prototype clusters in different scales may be an alternative to study in these cases, since they may provide more information than objects in the low dimensional spaces which are obtained after prototype selection. Another option is to first select the scales with respect to their diversity.

For datasets that are continuously growing (e.g. biometric datasets) but require accurate responses at each moment for a new classification query one could wonder when and how to update the already selected prototypes. As far as we know, this problem has not been investigated.

In general, the study towards selecting prototypes in the DS is a promising research direction. The new insights provided in this thesis can be used as a basis to advance this field further. A combined approach including the strategies presented in this thesis (e.g. clusters as prototypes computed from extended dissimilarity spaces and selection by the proposed GAs on top of this) may improve upon the individual strategies.

Bibliography

[1] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley-Interscience Publication (2000)

[2] Bunke, H., Sanfeliu, A., eds.: Syntactic and Structural Pattern Recognition Theory and Applications. World Scientific (1990)

[3] Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition: Foundations and Applications (Machine Perception and Artificial Intelligence). World Scientific Publishing Co., Inc., River Edge, NJ, USA (2005)

[4] Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(1) (Jan 2000) 4–37

[5] Duin, R.: Non-euclidean problems in pattern recognition related to human expert knowledge. In Filipe, J., Cordeiro, J., eds.: Enterprise Information Systems. Volume 73 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2011) 15–28

[6] Duin, R.P.W., Pekalska, E., Loog, M. Advances in Computer Vision and Pattern Recognition. In: Non-Euclidean Dissimilarities: Causes, Embedding and Informativeness. Springer London (2013) 13–44

[7] Bunke, H., Günter, S., Jiang, X.: Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In: Proceedings of the Second International Conference on Advances in Pattern Recognition. ICAPR '01, London, UK, UK, Springer-Verlag (2001) 1–11

[8] Duin, R.P.W., Pkalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recogn. Lett. **33**(7) (May 2012) 826–832

[9] Duin, R., Bicego, M., Orozco-Alzate, M., Kim, S.W., Loog, M.: Metric learning in dissimilarity space for improved nearest neighbor performance. In Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 8621 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 183–192

[10] Duin, R.P.W., Pkalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recogn. Lett. **33**(7) (May 2012) 826–832

[11] Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recogn. **39**(2) (2006) 189–208

[12] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **13**(1) (September 2006) 21–27

[13] Hart, P.E.: The condensed nearest neighbor rule. IEEE Transactions on Information Theory **IT-14**(3) (May 1968) 515–516

[14] Pekalska, E., Paclik, P., Duin, R.P.W.: A generalized kernel approach to dissimilarity-based classification. J. Mach. Learn. Res. **2** (March 2002) 175–211

[15] Pekalska, E., Paclik, P., Duin, R.: A Generalized Kernel Approach to Dissimilarity Based Classification. J. of Machine Learning Research **2**(2) (2002) 175–211

[16] Olivetti, E., Nguyen, T.B., Garyfallidis, E.: The approximation of the dissimilarity projection. In: Second Workshop on Pattern Recognition in NeuroImaging. PRNI '12, Washington, DC, USA, IEEE Computer Society (2012) 85–88

[17] Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pekalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recogn. **39**(10) (2006) 1827–1838

[18] Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. Neural Networks, IEEE Transactions on **13**(2) (Mar 2002) 415–425

[19] Pekalska, E., Harol, A., Duin, R., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative. In: Joint IAPR Int. Workshops on SSPR. (2006) 871–880

[20] Plasencia-Calaña, Y., Garcia-Reyes, E., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: Proceedings of the 2010 20th International Conference on Pattern Recognition. ICPR '10, Washington, DC, USA, IEEE Computer Society (2010) 177–180

[21] Plasencia-Calaña, Y., Orozco-Alzate, M., Méndez-Vázquez, H., García-Reyes, E., Duin, R.P.W.: Towards scalable prototype selection by genetic algorithms with fast criteria. In Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M., eds.: Structural, Syntactic, and Statistical Pattern Recognition. Volume 8621 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 343–352

[22] Plasencia-Calaña, Y., Orozco-Alzate, M., García-Reyes, E., Duin, R.P.W.: Selecting feature lines in generalized dissimilarity representations for pattern recognition. Digit. Signal Process. **23**(3) (May 2013) 902–911

[23] Plasencia-Calaña, Y., Orozco-Alzate, M., García-Reyes, E., Duin, R.: Towards cluster-based prototype sets for classification in the dissimilarity space. In Ruiz-Shulcloper, J., Sanniti di Baja, G., eds.: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Volume 8258 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 294–301

[24] Plasencia-Calaña, Y., García-Reyes, E.B., Duin, R.P.W., Orozco-Alzate, M.: On using asymmetry information for classification in extended dissimilarity spaces. In Álvarez, L., Mejail, M., Gómez, L., Jacobo, J.C., eds.: CIARP. Volume 7441 of LNCS., Springer (2012) 503–510

[25] Plasencia-Calaña, Y., Cheplygina, V., Duin, R.P.W., García-Reyes, E.B., Orozco-Alzate, M., Tax, D.M.J., Loog, M.: On the informativeness of asymmetric dissimilarities. In: Proceedings of the Second International Conference on Similarity-Based Pattern Recognition. SIMBAD'13, Berlin, Heidelberg, Springer-Verlag (2013) 75–89

[26] Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J., Torkkola, K.: LVQ_PAK: The Learning Vector Quantization program package. Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science (January 1996)

[27] Devijver, P., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice Hall (1982)

[28] Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. In: Machine Learning. (2000) 257–286

[29] Riesen, K., Neuhaus, M., Bunke, H.: Graph embedding in vector spaces by means of prototype selection. In Escolano, F., Vento, M., eds.: GbRPR. Volume 4538 of Lecture Notes in Computer Science., Springer (2007) 383–393

[30] Spillmann, B., Neuhaus, M., Bunke, H., Pekalska, E., Duin, R.P.W.: Transforming strings to vector spaces using prototype selection. In Yeung, D.Y., Kwok, J.T., Fred, A.L.N., Roli, F., de Ridder, D., eds.: SSPR/SPR. Volume 4109 of Lecture Notes in Computer Science., Springer (2006) 287–296

[31] Orozco-Alzate, M., Duin, R.P.W., Castellanos-Domínguez, G.: A generalization of dissimilarity representations using feature lines and feature planes. Pattern Recogn. Lett. **30**(3) (2009) 242–254

[32] Kim, S.W.: On using a dissimilarity representation method to solve the small sample size problem for face recognition. In: ACIVS. (2006) 1174–1185

[33] Jain, A.K., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(2) (1997) 153–158

[34] Graepel, T., Herbrich, R., Schölkopf, B., Smola, A., Bartlett, P., Müller, K.R., Obermayer, K., Williamson, R.C.: Classification on proximity data with lp Ű machines. In: Proceedings of the Ninth International Conference on Artificial Neural Networks. (1999) 304–309

[35] Hochbaum, Shmoys: A best possible heuristic for the k-center problem. Mathematics of Operations Research **10**(2) (1985) 180–184

[36] Orozco-Alzate, M.: Generalized Dissimilarity Representations for Pattern Recognition. PhD thesis, Universidad Nacional de Colombia Sede Manizales (October 2008)

[37] Bunke, H., Riesen, K.: Graph classication based on dissimilarity space embedding. In: N. da Vitoria Lobo et al., editor, SSSPR, LNCS 5342. (2008) 996–1008

[38] Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large scale feature selection. Pattern Recognition Letters **10** (1989) 335–347

[39] Muresan, D.A.: Genetic algorithms for nearest neighbor. Technical report, California Institute of Technology, CA (1997)

[40] Kuncheva, L.I., Jain, L.C.: Nearest neighbor classifier: Simultaneous editing and feature selection. Pattern Recognition Letters **20** (1999) 1149–1156

[41] Pekalska, E., Duin, R.P.: Datasets and tools for dissimilarity analysis in pattern recognition, simbad technical report. (2009)

[42] Scannell, J.W., Blakemore, C., Young, M.P.: Analysis of connectivity in the cat cerebral cortex. Journal of Neuroscience **15** (1995) 1463

[43] Bunke, H., Buhler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recogn. **26**(12) (December 1993) 1797–1812

[44] Xiao, B., Hancock, E.R.: Geometric characterisation of graphs. In: ICIAP. (2005) 471–478

[45] Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **18**(4) (1996) 377–388

[46] Lee, W.J., Duin, R.P.: An inexact graph comparison approach in joint eigenspace. In: SSPR & SPR '08: Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Berlin, Heidelberg, Springer-Verlag (2008) 35–44

[47] Bunke, H., Riesen, K.: Graph classification based on dissimilarity space embedding. In: Structural, Syntactic, and Statistical Pattern Recognition. SSPR & SPR '08, Berlin, Heidelberg, Springer-Verlag (2008) 996–1007

[48] García-Pedrajas, N., Haro-García, A.: Scaling up data mining algorithms: review and taxonomy. Progress in Artificial Intelligence **1**(1) (2012) 71–87

[49] Spillmann, B., Neuhaus, M., Bunke, H., Pekalska, E., Duin, R.P.W.: Transforming strings to vector spaces using prototype selection. In Yeung, D.Y., Kwok, J.T., Fred, A.L.N., Roli, F., de Ridder, D., eds.: SSPR/SPR. Volume 4109 of Lecture Notes in Computer Science., Springer (2006) 287–296

[50] Ma, B., Hero, A., Gorman, J., Michel, O.: Image registration with minimum spanning tree algorithm. In: Proceedings of the International Conference on Image Processing. Volume 1. (2000) 481–484

[51] Zare Borzeshi, E., Piccardi, M., Riesen, K., Bunke, H.: Discriminative prototype selection methods for graph embedding. Pattern Recogn. **46**(6) (June 2013) 1648–1657

[52] Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognitive Neuroscience **3**(1) (January 1991) 71–86

[53] Jain, A.K., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. Pattern Anal. Mach. Intell. **19** (December 1997) 1386–1391

[54] Alimoglu, F., Alpaydin, E.: Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In: Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium. (1996)

[55] Messer, K., Matas, J., Kittler, J., Jonsson, K.: XM2VTSDB: The extended M2VTS database. In: Second International Conference on Audio and Video-based Biometric Person Authentication. (1999) 72–77

[56] Frank, A., Asuncion, A.: UCI machine learning repository (2010)

[57] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (November 1998) 2278–2324

[58] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning. (2011)

[59] Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Proc. IEEE Conf. CVPR. (2011)

[60] Du, H., Chen, Y.Q.: Rectified nearest feature line segment for pattern classification. Pattern Recognition **40**(5) (2007) 1486–1497

[61] Orozco-Alzate, M., Duin, R.P.W., Castellanos-Domínguez, C.G.: On selecting middle-length feature lines for dissimilarity-based classification. In: XII Simposio de Tratamiento de Señales, Imágenes y Visión Artificial, STSIVA 2007, Universidad del Norte, Capítulo de la Sociedad de Procesamiento de Señales, IEEE Sección Colombia; Departamento de Eléctrica y Electrónica – Fundación Universidad del Norte y Rama Estudiantil IEEE – UniNorte (September 2007)

[62] Li, S.Z., Lu, J.: Face recognition using the nearest feature line method. IEEE Trans. Neural Networks **10**(2) (1999) 439–443

[63] Chien, J.T., Wu, C.C.: Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans. Pattern Anal. Machine Intell. **24**(12) (2002) 1644–1649

[64] Duin, R.P.W., Juszczak, P., de Ridder, D., Paclik, P., Pekalska, E., Tax, D.M.J.: PRTools4: a Matlab Toolbox for Pattern Recognition. Technical report, Information and Communication Theory Group: Delft University of Technology, The Netherlands (2004) `http://www.prtools.org/`.

[65] de Veld, D.C.G., Skurichina, M., Witjes, M.J.H., Duin, R.P.W., Sterenborg, D.J.C.M., Star, W.M., Roodenburg, J.L.N.: Autofluorescence characteristics of healthy oral mucosa at different anatomical sites. Lasers in Surgery and Medicine **32**(5) (2003) 367–376

[66] Kuncheva, L.I.: Real medical data sets. Technical report, School of Computer Science: Bangor University, Bangor, UK (2005)

[67] Cox, T., Cox, M.: Multidimensional scaling. Chapman and Hall, London (1994)

[68] Riesen, K., Bunke, H.: Graph classification by means of lipschitz embedding. Trans. Sys. Man Cyber. Part B **39**(6) (December 2009) 1472–1483

[69] Frey, B.J.J., Dueck, D.: Clustering by passing messages between data points. Science **315** (January 2007) 972–976

[70] Cox, T.F., Cox, M.: Multidimensional Scaling, Second Edition. 2 edn. Chapman and Hall/CRC (2000)

[71] Bengio, Y., Paiement, J.F., Vincent, P., Delalleau, O., Roux, N.L., Ouimet, M.: Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering. In: Advances in Neural Information Processing Systems, MIT Press (2003) 177–184

[72] Baker, C.T.H.: The numerical treatment of integral equations. Clarendon Press, Oxford ; New York : (1977)

[73] Sigillito, V.G., Wing, S.P., Hutton, L.V., Baker, K.B.: Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest (1989) 262–266

[74] Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4) (April 2002) 509–522

[75] Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5) (May 2004) 550–571

[76] Breiman, L.: Bias, variance, and arcing classifiers. Technical report (1996)

[77] Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Trans. Syst. Man Cybern. C, Appl. Rev. **38**(6) (2008) 729–744

[78] Duin, R.P.W., Pekalska, E.: Non-Euclidean dissimilarities: causes and informativeness. In: Proceedings of the 2010 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition. SSPR & SPR'10, Berlin, Heidelberg, Springer-Verlag (2010) 324–333

[79] Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. **10**(5) (1999) 1000–1017

[80] Okada, A., Imaizumi, T.: Nonmetric multidimensional scaling of asymmetric proximities. Behaviormetrika **14**(21) (1987) 81–96

[81] Martin-Merino, M., Muñoz, A.: Self organizing map and sammon mapping for asymmetric proximities. In: Proceedings of the International Conference on Artificial Neural Networks. ICANN '01, London, UK, UK, Springer-Verlag (2001) 429–435

[82] Muñoz, A., de Diego, I.M., Moguerza, J.M.: Support vector machine classifiers for asymmetric proximities. In Kaynak, O., Alpaydin, E., Oja, E., Xu, L., eds.: Artificial Neural Networks and Neural Information Processing Ů ICANN/ICONIP 2003. Volume 2714 of LNCS. Springer Berlin Heidelberg (2003) 217–224

[83] Schölkopf, B., Mika, S., Burges, C.J.C., Knirsch, P., Müller, K.R., Rätsch, G., Smola, A.J.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. **10**(5) (1999) 1000–1017

[84] Bowdle, B., Gentner, D.: Informativity and asymmetry in comparisons. Cogn. Psychol. **34**(3) (1997) 244–286

[85] Dietterich, T., Lathrop, R., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1-2) (1997) 31–71

[86] Gärtner, T., Flach, P., Kowalczyk, A., Smola, A.: Multi-instance kernels. In: Proc. of the 19th Int. Conf. on Machine Learning. (2002) 179–186

[87] Tax, D.M.J., Loog, M., Duin, R.P.W., Cheplygina, V., Lee, W.J.: Bag dissimilarities for multiple instance learning. In: Similarity-Based Pattern Recognition. Volume 7005 of LNCS., Springer (2011) 222–234

[88] Cheplygina, V., Tax, D.M.J., Loog, M.: Class-dependent dissimilarity measures for multiple instance learning. Volume 7626 of LNCS., Springer (2012) 602–610

[89] Dinh, C., Duin, R.P.W., Loog, M.: A study on semi-supervised dissimilarity representation. In: International Conference on Pattern Recognition. (2012)

[90] Rahmani, R., Goldman, S., Zhang, H., Krettek, J., Fritts, J.: Localized content based image retrieval. In: Proc. of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval, ACM (2005) 227–236

[91] Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X., Raich, R., Hadley, S., Hadley, A., Betts, M.: Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. J. Acoust. Soc. Am. **131** (2012) 4640

[92] Duin, R., Juszczak, P., Paclik, P., Pekalska, E., Ridder, D.D., Tax, D., Verzakov, S.: A Matlab toolbox for pattern recognition. PRTools version **3** (2000)

[93] Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vision **30**(2) (November 1998) 79–116

[94] Huang, X., Zhang, L., Li, P.: A multiscale feature fusion approach for classification of very high resolution satellite imagery based on wavelet transform. Int. J. Remote Sens. **29**(20) (October 2008) 5923–5941

[95] Loog, M., Li, Y., Tax, D.: Maximum membership scale selection. In Benediktsson, J., Kittler, J., Roli, F., eds.: Multiple Classifier Systems. Volume 5519 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 468–477

[96] Liu, Y.m., Ye, L.b., Zheng, P.y., Shi, X.r., Hu, B., Liang, J.: Multiscale classification and its application to process monitoring. Journal of Zhejiang University SCIENCE C **11** (2010) 425–434

[97] Li, Y., Duin, R.P.W., Loog, M.: Combining multi-scale dissimilarities for image classification. In: Proceedings of the 2012 21th International Conference on Pattern Recognition. ICPR '12, IEEE Computer Society (2012)

[98] Li, Y., Tax, D.M., Loog, M.: Scale selection for supervised image segmentation. Image and Vision Computing **30**(12) (2012) 991 – 1003

[99] Castellani, U., Ulas, A., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., Brambilla, P.: Selecting scales by multiple kernel learning for shape diffusion analysis. In Pennec, X., Joshi, S., Nielsen, M., eds.: Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy - Geometrical and Statistical Methods for Modelling Biological Shape Variability, Toronto, Canada (September 2011) 148–158

[100] Gönen, M., Alpaydin, E.: Multiple kernel learning algorithms. J. Mach. Learn. Res. **12** (July 2011) 2211–2268

[101] Nilufar, S., Ray, N., Zhang, H.: Object detection with DoG scale-space: A multiple kernel learning approach. IEEE Transactions on Image Processing **21**(8) (2012) 3744–3756

[102] Ibba, A., Duin, R.P.W.: A multiscale approach in combining classifiers in dissimilarity representations. In Gevers, T., Bos, H., Wolters, L., eds.: ASCI 2009, 15th Annual Conf. of the Advanced School for Computing and Imaging. (2009)

[103] Guillaumin, M., Verbeek, J., Schmid, C.: Is that you? metric learning approaches for face identification. In: International Conference on Computer Vision. (sep 2009) 498–505

[104] Zhu, P., Zhang, L., Hu, Q., Shiu, S.C.K.: Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. In Fitzgibbon, A.W., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: ECCV (1). Volume 7572 of Lecture Notes in Computer Science., Springer (2012) 822–835

[105] Zhu, P., Hu, Q.: Adaptive neighborhood granularity selection and combination based on margin distribution optimization. Information Sciences **249**(0) (2013) 1 – 12

[106] De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, Ann Arbor, MI, USA (1975) AAI7609381.

[107] Kuncheva, L.I., Bezdek, J.C.: On prototype selection: Genetic algorithms or random search? IEEE Transactions on Systems, Man, and Cybernetics **C28**(1) (1998) 160–164

[108] Hong, J.H., Cho, S.B.: Efficient huge-scale feature selection with speciated genetic algorithm. Pattern Recogn. Lett. **27**(2) (January 2006) 143–150

[109] Kalkan, H., Nap, M., Duin, R.P.W., Loog, M.: Automated colorectal cancer diagnosis for whole-slice histopathology. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI 2012). Volume LNCS 7512., Nice, France, Springer, Springer (2012) 550–557

[110] Kalkan, H., Nap, M., Duin, R.P.W., Loog, M.: Automated classification of local patches in colon histopathology. In: 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, IEEE, IEEE (2012) 61–64

[111] Available at: http://www.ux.uis.no/ tranden/brodatz.html

[112] Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. Int. J. Comput. Vision **43**(1) (June 2001) 29–44

[113] Grefenstette, J.: Optimization of control parameters for genetic algorithms. IEEE Trans. Syst. Man Cybern. **16** (January 1986) 122–128

Summary

Automatic pattern classification for a given problem domain aims at assigning a class or category membership to a new unseen object from the same domain. This is performed in three main stages: data preprocessing, representation and classification. The data preprocessing highly depends on the data type (e.g. images, signals) which makes its study highly domain dependent. The representation and classification stages are more general and the same type of representations or classifier can be studied for different problems. This thesis focusses on the representation stage as better representation will result in better classification performances. Traditionally, pattern recognition made use of vector space representation and structural representations. Drawbacks, like the possible unavailability of distinguishing features and lack of learning tools for the structural representations, have led to alternatives such as the Dissimilarity Representation (DR), which is a relational representation where the objects are represented by the (potentially non-Euclidean and non-metric) dissimilarities to a set of prototypes.

One of the possibilities when considering DRs is the Dissimilarity Space (DS) approach. It was postulated as an Euclidean space where an object is represented by its dissimilarities to a set of prototypes. The DS is attractive since it gives a good trade off between accuracy and computational cost of the representation especially when the prototypes are carefully selected. In this thesis we study how to define and select the prototypes for creating good representations in the sense of compromise between good classification accuracy and low computational cost. Our main research question is: Can we create better prototypes and/or selection procedures if we take the nature and characteristics of the dissimilarity data into account in the process?

This thesis presents new prototype selection methods based on Genetic Algorithms (GAs). As prototypes are homogeneously spread over the dissimilarity space and similar objects have similar representational power, randomized methods such as GA are a powerful approach for selecting prototypes. These properties were further exploited by proposing two new scalable methods based on GAs with two new scalable criteria to be evaluated in the GA fitness function, i.e. maximizing the weight of the minimum spanning tree of the set of prototypes, and, maximizing matching labels of objects and their assigned prototypes after a nearest prototype clustering is performed. We found that for multiclass problems our proposed criteria based on maximizing diversity of the prototypes was crucial to select good prototypes.

The second part of the thesis studies the creation and selection of models as prototypes for classification in generalized dissimilarity spaces. A new method, based on the nearest feature line classifier, is proposed to select feature lines as prototypes. Feature lines are suitable for data under representational limitations. We also studied the creation and selection of clusters as prototypes. We considered different ways to measure distances of objects with clusters: the minimum, maximum, and average statistics. A new method based on the *Nyström* formula was proposed to measure a subspace distance of objects with the positive part of the subspace created by the objects inside the cluster considering only the information contained in the given dissimilarities. The results of the study showed that cluster-based prototypes where always better than object-based prototypes when comparing DSs of the same dimension.

The last part of the thesis studies the creation and selection of extended prototypes. First, the extension is achieved by considering directed asymmetric dissimilarities, where we obtain

two dissimilarity values computed from the objects to the prototypes and viceversa. Prototype selection in extended asymmetric dissimilarity spaces is studied as an alternative to symmetrization by averaging, minimum and maximum as well as the two individual directed DS. Supervised procedures are studied to perform the selection since they are able to select the prototypes with their best associated direction for the computation of dissimilarities. It was concluded from this study that there is useful information in asymmetry and the dissimilarity space with prototype selection is a means to use this information. In addition, we studied another alternative to use extended prototypes for multiscale dissimilarity data. In this case, the prototypes are selected in an extended multiscale dissimilarity space (EMDS). A GA optimizing a classification criterion was proposed due to its ability to cope with large candidate sets of prototypes. It finds the best prototypes with their best related scales in order to take advantage of multiscale data provided in the form of dissimilarities. We found that our proposal of using a reduced EMDS by prototype selection was useful for problems where the scales perform significantly different.

This thesis has contributed to gain insights on the topic of prototype selection for classification in the DS. We found that diversity plays a key role for the selection of prototypes, and this explains why a set of random prototypes is already a good starting point and why GAs are powerful methods to optimize this set further. In addition, we discovered that by creating and selecting more complicated models (depending on the data characteristics), such as clusters or feature lines as prototypes, the classification accuracies are increased. This points to promising research directions such as automatically learning the best prototype models. Finally, we showed that it is beneficial for classification to include extra knowledge, such as asymmetry in data or multiscale dissimilarities, in an extended dissimilarity representation.

Samenvatting

De automatische patroonherkenning richt zich, binnen een gegeven probleemdomein, op het toekennen van een klasse of een categorie aan nieuwe, nog niet geanalyseerde objecten uit hetzelfde domein. Dit voltrekt zich in drie stappen: de voorbewerking van de data, de representatie en de classificatie. De voorbewerking hangt sterk af van het soort data (bijv. beelden of signalen). Hierdoor wordt de analyse sterk afhankelijk van het domein. De stappen van representatie en classificatie zijn meer algemeen waardoor dezelfde typen representaties en classificatoren kunnen worden bestudeerd voor verschillende problemen. Dit proefschrift richt zich op de representatiestap omdat een betere representatie tot een betere classificator zal leiden. Traditioneel worden in de patroonherkenning de vectorruimte of een structuurbeschrijving als representatie gebruikt. Nadelen, zoals het niet beschikbaar zijn van specifieke kenmerken en het gebrek aan classificatoren voor structurele representaties, hebben geleid tot de dissimilariteitsrepresentatie (DR). Dit is een relationele representatie waarbij objecten worden gerepresenteerd door hun verschillen met een verzameling prototypes.

Een van de mogelijkheden voor een DR is de dissimilariteitsruimte. Dit is een Euclidische ruimte waarin objecten worden gerepresenteerd door dissimilariteiten met prototypes. Een van de interessante aspecten van deze representatie is dat hij, door een zorgvuldige selectie van prototypes, een flexibele trade-off mogelijk maakt tussen de computationele kosten en de bereikte classificatienauwkeurigheid. In dit proefschrift wordt de wijze bestudeerd waarop prototypes kunnen worden gedefinieerd en geselecteerd zodat een goed compromis wordt gerealiseerd tussen classificatie nauwkeurigheid en computationele kosten. Onze belangrijkste onderzoekvraag is: kunnen betere prototypes en/of selectie procedures worden gevonden door aard en karakteristieken van de dissimilariteit in aanmerking te nemen?

Dit proefschrift presenteert nieuwe selectiemethoden voor prototypes gebaseerd op genetische algoritmen (GAs). Omdat prototypes op een homogene manier in de ruimte zijn verdeeld hebben naburige, en daardoor soortgelijke objecten, ongeveer hetzelfde representatieve vermogen. GAs maken hieruit een min of meer willekeurige keuze en zijn daardoor een krachtig hulpmiddel voor het selecteren van prototypes.

De eigenschappen van de DR zijn verder benut en gebruikt voor twee nieuwe, schaalbare GA methoden gebruikmakend van schaalbare criteria. Deze zijn het maximaliseren van de gewichten van de 'minimum spanning tree' van de verzameling prototypes en het maximaliseren van de match tussen de labels van objecten en die van prototypes na een cluster analyse. Bij de analyse van meerklassenproblemen kon worden geconstateerd dat voor de onderzochte criteria het maximaliseren van de diversiteit cruciaal was.

In het tweede deel van het proefschrift is de constructie en selectie van modellen onderzocht, te gebruiken als prototypes voor een gegeneraliseerde DR. Allereerst wordt een nieuwe methode voorgesteld gebaseerd op de 'nearest feature line' classificator. 'Feature lines' zijn onder bepaalde voorwaarde geschikt voor representatie. Daarnaast is de analyse en selectie van clusters ten behoeve van prototypes bestudeerd. Diverse manieren zijn beschouwd om afstanden tussen objecten en clusters te meten. Een nieuwe methode, gebaseerd op de Nyström vergelijking is onderzocht. Hiermee kunnen afstanden worden bepaald tussen objecten en een cluster door op basis van uitsluitend de gegeven dissimilariteiten een nieuwe ruimte te creëren. De resultaten

van deze studies lieten zien dat op clusters gebaseerde prototypes altijd beter waren dan bij het directe gebruik van objecten als prototypes bij dissimilariteitsruimtes van dezelfde dimensie.

In het laatste deel van het proefschrift zijn de constructie en selectie van geaugmenteerde prototypes beschreven. Eerst wordt hun uitbreiding beschouwd door de beide, gerichte, asymmetrische dissimilariteiten te gebruiken die worden verkregen door objecten met prototypes te vergelijken en andersom. Prototype selectie in geaugmenteerde asymmetrische dissimilariteitsruimtes is bestudeerd als een alternatief van symmetrisatie d.m.v. middeling, het nemen van minima of maxima, of het kiezen van een van de twee gerichte dissimilariteiten. Het kon worden geconcludeerd dat de asymmetrie nuttige informatie bevat en dat prototype selectie een manier is om deze informatie te gebruiken. Hiernaast is nog een studie verricht om geaugmenteerde prototypes te gebruiken voor meerschalige dissimilariteiten. Een GA op basis van een classificatiecriterium is gebruikt vanwege zijn vermogen om uit grote verzamelingen kandidaten te selecteren. Hierdoor kunnen de beste prototypes met de beste bijbehorende schaal worden gevonden, gebruikmakend van de multischaal dissimilariteiten. De onderzochte methode bleek nuttig te zijn voor problemen waarbij de individuele schalen significant verschillend waren.

Dit proefschrift heeft een bijdrage geleverd aan de selectie van prototypes voor dissimilariteitsruimtes. De diversiteit speelt een belangrijke rol bij deze selectie. Dit verklaart waarom een initiële random keuze al een goed resultaat levert. GAs bieden een krachtig hulpmiddel om deze verder te optimaliseren. Daarnaast hebben we gevonden dat door het construeren en selecteren van meer ingewikkelde modellen (afhankelijk van de data eigenschappen) zoals clusters en 'feature lines', en deze te gebruiken als prototypes, de classificatiefout kan worden vergroot. Dit wijst naar veelbelovende onderzoeksrichtingen als het automatisch leren van de beste prototype modellen. Tenslotte is aangetoond dat het voordelig is voor de classificatie om extra kennis in de vorm van asymmetrieën en multischaal dissimilariteiten te gebruiken voor een geaugmenteerde representatie.

Acknowledgments

Curriculum Vitae

Yenisel Plasencia Calaña was born in La Habana, Cuba, on May 11, 1982. From 2000 to 2006 she studied Computer Science Licenciate at the University of Havana, La Habana, Cuba, where she obtained in 2006 the Lic. degree, which is a first-level university degree, equivalent to a Master degree. During her studies, she was a teacher of Programming for university students in their 3rd year. Her thesis project was a system for the analysis of malicious software (malware) where she incorporated information for the automatic analysis of malicious code depending on their signature, as well as the audit of the system. Since 2006, she works as a researcher and software developer at the Advanced Technologies Application Center (CENATAV) in the area of Pattern Recognition and Biometrics. From 2006 to 2008 she worked in the topic of illumination insensitive face recognition in linear and nonlinear subspaces.

In 2008, after the visit of Dr. ir. Robert P. W. Duin to her institution, she started her PhD studies in cooperation with the Pattern Recognition group at the Information and Communication Theory Department, Faculty of Electrical Engineering, Mathematics and Computer Science, at Delft University of Technology (TUDelft). She studied the selection of prototypes for the creation of the dissimilarity space as PhD topic, under the supervision of Dr. Edel Garcia Reyes from CENATAV and Dr. ir. Robert P. W. Duin from TUDelft. In the middle of 2010 she took a year of maternity leave granted by the Cuban government, maintaining, however, communication with her PhD supervisors in order to meet the PhD commitments. After that, she incorporated to her regular work activities including the PhD studies.

She has supervised two computer science diploma thesis and is currently supervising another one. She has been appointed three times as a tribunal member for diploma thesis defense. She was selected as a member of the scientific council of her institution in 2014. Currently, she is tutoring a PhD student and is leading or participating since 2014 in other four research lines related to Biometrics. She is a reviewer for international conferences and journals. She has obtained two scientific awards including the annual award of the Cuban academy of sciences in 2012 as co-author of a project leaded by Dr. Heydi Méndez Vázquez from CENATAV. She is currently a member of: the Cuban Society of Mathematics and Computer Science (SCMC), the Cuban Association of Pattern Recognition (ACRP) and the International Association of Pattern Recognition (IAPR). She has participated in national and international conferences.

Her research interests include: dissimilarity-based representations for pattern recognition, face biometrics, and scalable methods for pattern recognition.

List of publications in chronological order:

1. Heydi Méndez-Vázquez, Yenisel Plasencia-Calaña, Francisco José Silva Mata, and Yadira Condes Molleda. A rank based classifier combination to improve face recognition. Memorias de RECPAT 2007, ISBN 978-959-286-006-3.

2. Heydi Méndez-Vázquez, César San Martín, Josef Kittler, Yenisel Plasencia-Calaña and Edel García-Reyes. Face Recognition with LWIR Imagery using Local Binary Patterns. Proceedings of the IAPR International Conference on Biometrics (ICB-2009). Advances in Biometrics, LNCS 5558, 327-336.

3. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Robert P. W. Duin, Heydi Mendez, César San Martin, and Claudio Soto. Dissimilarity representations for thermal signature recognition at a distance. Proceedings of the fifteenth annual conference of the Advanced School for Computing and Imaging 2009, ISBN/EAN: 978-90-810849-4-9.

4. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Robert P. W. Duin, Heydi Mendez, César San Martin, and Claudio Soto. A Study on Representations for Face Recognition from Thermal Images. Progress in Pattern Recognition, image analysis and applications: 14th Iberoamerican Congress on Pattern Recognition, CIARP 2009. LNCS 5856, ISBN 978-3-642-10267-7.

5. Yenisel Plasencia-Calaña, Edel Garcia Reyes. Subspaces and manifolds for ilumination insensitive face recognition. RNPS No.2142 ISSN 2072-6287, 2010 (in spanish)

6. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Mauricio Orozco Alzate, Robert P. W. Duin. Prototype Selection Methods for Dissimilarity Space Classification. RNPS No.2142 ISSN 2072-6287, 2010.

7. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Mauricio Orozco Alzate, Robert P. W. Duin. Prototype Selection for Dissimilarity Representation by a Genetic Algorithm. Proceedings of the 20th IEEE International Conference on Pattern Recognition (ICPR'10), ISBN: 978-1-4244-7542-1.

8. César San Martin, Roberto Carrillo, Pablo Meza, Heydi Méndez Vázquez, Yenisel Plasencia-Calaña, Edel Garcia Reyes, Gabriel Hermosilla. Recent Advances on Face Recognition using Thermal Infrared Images. Reviews, Refinements and New Ideas in Face Recognition, Peter M. Corcoran (Ed.) (2011), ISBN: 978-953-307-368-2, InTech.

9. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Robert P. W. Duin, and Mauricio Orozco-Alzate. A New Method for Prototype Selection in Dissimilarity Spaces. RECPAT 2011, cd memorias de COMPUMAT 2011, ISBN 978-959-250-658-9.

10. Yoanna Martínez-Díaz, Heydi Méndez-Vázquez, Yenisel Plasencia-Calaña, Edel Garcia Reyes. Dissimilarity Representations based on Multi-Block LBP for face detection, 17th Iberoamerican Congress in Pattern Recognition CIARP 2012 LNCS 7441, pp. 106–113. Springer, (2012).

11. Yenisel Plasencia-Calaña, Edel Garcia Reyes, Robert P. W. Duin, and Mauricio Orozco-Alzate. On Using Asymmetry Information for Classification in Extended Dissimilarity Spaces. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science Volume 7441, 2012, pp 503-510. ISBN: 978-3-642-33275-3

12. Yenisel Plasencia-Calaña, M. Orozco-Alzate, E. Garcia-Reyes, and Robert P. W. Duin. Selecting feature lines in generalized dissimilarity representations for pattern recognition. Digital Signal Processing, vol. 23, no. 3, 2013, 902-911.

13. Yenisel Plasencia-Calaña, Veronika Cheplygina, Robert P. W. Duin, Edel García-Reyes, Mauricio Orozco-Alzate, David M J Tax, Marco Loog. On the Informativeness of Asymmetric Dissimilarities, Similarity-Based Pattern Recognition, Second International Workshop, SIMBAD 2013, Edwin Hancock and Marcello Pelillo (Eds.). LNCS 7953, pp. 75–89. Springer, Heidelberg (2013)

14. Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Edel García-Reyes, and Robert P. W. Duin. Towards Cluster-Based Prototype Sets for Classification in the Dissimilarity Space. CIARP 2013, Part I, J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.), LNCS 8258, pp. 294–301. Springer, Heidelberg (2013)

15. Leonardo Chang, Nelson Mendez, Yenisel Plasencia-Calaña, Heydi Mendez-Vazquez. Facial Landmarks Detection using Extended Profile LBP-based Active Shape Models. CIARP 2013, Part I, J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.), LNCS 8258, pp. 294–301. Springer, Heidelberg (2013)

16. Veronika Cheplygina, Yenisel Plasencia-Calaña, Robert P. W. Duin, Edel Garcia-Reyes, Mauricio Orozco-Alzate, David Tax and Marco Loog. On Asymmetry in Dissimilarities. Benelux conference on artificial intelligence (2013)

17. Nelson Méndez Llanes, Leonardo Chang, Yenisel Plasencia-Calaña, Heydi Méndez-Vázquez. Face landmarks detection using an improved ASM local description. RECPAT 2014 (in spanish).

18. Mairelys Hernández Durán, Heydi Méndez-Vázquez, Yenisel Plasencia-Calaña. Current trends in low-resolution face recognition. RECPAT 2014 (in spanish).

19. Maria De Marsico, Daniel Riccio, Heydi Méndez Vázquez and Yenisel Plasencia-Calaña. GETSEL: Gallery Entropy for Template SElection on Large datasets. IEEE International Joint Conference on Biometrics (IJCB 2014), At Clearwater, Florida, USA.

20. Yenisel Plasencia-Calaña, Mauricio Orozco-Alzate, Heydi Méndez-Vázquez, Edel García-Reyes, Robert P. W. Duin. Towards Scalable Prototype Selection by Genetic Algorithms with Fast Criteria. In: Joint IAPR International Workshop on Statistical Techniques in Pattern Recognition (SPR 2014), Structural, Syntactic and Statistical Pattern Recognition S+SSPR 2014, August 20-22, 2014. Joensuu, Finland. Pasi Fränti, et al. (Eds.), LNCS, Vol. 8621, 2014, pp. 343-352. Berlin Heidelberg, Springer-Verlag, 2014.

21. Mairelys Hernández Durán, Heydi Méndez-Vázquez, Yenisel Plasencia-Calaña. State-of-the-art in Low-resolution Face Recognition. RNPS No.2142 ISSN 2072-6287, 2015. Methods