Trends in Nearest Feature Classification for Face Recognition – Achievements and Perspectives

Mauricio Orozco-Alzate and César Germán Castellanos-Domínguez
Universidad Nacional de Colombia Sede Manizales
Colombia

1. Introduction

Face recognition has become one of the most intensively investigated topics in biometrics. Recent and comprehensive surveys found in the literature, such as (Zhao et al., 2003; Ruiz-del Solar & Navarrete, 2005; Delac & Grgic, 2007), provide a good indication of how active are the research activities in this area. Likewise in other fields in pattern recognition, the identification of faces has been addressed from different approaches according to the chosen representation and the design of the classification method. Over the past two decades, industrial interests and research efforts in face recognition have been motivated by a wide range of potential applications such identification, verification, posture/gesture recognizers and intelligent multimodal systems. Unfortunately, counter effects are unavoidable when there is a heavily increased interest in a small research area. For the particular case of face recognition, most of the consequences were pointed out by three editors of the well-known *Pattern Recognition Letters* journal. The following effects on the publication of results were discussed by Duin et al. (2006):

1. The number of studies in face recognition is exploding and always increasing. Some of those studies are rather obvious and straightforward.
2. Many of the submitted papers have only a minor significance or low citation value. As a result, journals receive piles of highly overlapping related papers.
3. Results are not always comparable, even though the same data sets are used. This is due to the use of different or inconsistent experimental methodologies.

A par excellence example of the situation described above is the overwhelming interest in linear dimensionality reduction, especially in the so-called small sample size (SSS) case. It is one of the most busy study fields on pixel-based face recognition. Indeed, the SSS problem is almost always present on pixel-based problems due to the considerable difference between dimensions and the number of available examples. In spite of that apparent justification, most of the published works in this matter are minor contributions or old ideas phrased in a slightly different way. Of course, there are good exceptions, see e.g. (Nhat & Lee, 2007; Zhao & Yuen, 2007; Liu et al., 2007). Our discussion here should not be interpreted as an attack to authors interested in dimensionality reduction for face recognition; conversely, we just want to explain why we prefer to focus in subsequent stages of the pattern recognition system instead of in dimensionality reduction. In our opinion, making a significant contribution in

Source: State of the Art in Face Recognition, Book edited by: Dr. Mario I. Chacon M.,
ISBN -3-902613-42-4, pp. 250, January 2009, I-Tech, Vienna, Austria

linear dimensionality reduction is becoming more and more difficult since techniques have reached a well-established and satisfactory level. In contrast, we consider that there are more open issues in previous and subsequent stages to representation such as preprocessing and classification.

At the end of the nineties, a seminal paper published by Li and Lu (1999) introduced the concept of feature line. It consists in an extension of the classification capability of the nearest neighbor method by generalizing two points belonging to the same class through a line passing by those two points (Li, 2008). Such a line is called *feature line*. In (Li & Lu, 1998), it was suggested that the improvement gained by using feature lines is due to their faculty to expand the representational ability of the available feature points, accounting for new conditions not represented by the original set. Such an improvement was especially observed when the cardinality of the training set (sample size) per class is small. Consequently, the nearest feature line method constitutes an alternative approach to attack the SSS problem without using linear dimensionality reduction methods. In fact, the dimensionality is increased since the number of feature lines depends combinatorially on the number of training points or objects per class. Soon later, a number of studies for improving the concept of feature lines were reported. A family of extensions of the nearest feature line classifier appeared, mainly encompassing the nearest feature plane classifier, the nearest feature space classifier and several modified versions such as the rectified nearest feature line segment and the genetic nearest feature plane. In addition, an alternative classification scheme to extend the dissimilarity-based paradigm to nearest feature classification was recently proposed.

In the remaining part of this chapter, we will explain in detail that family of nearest feature classifiers as well as their modified and extended versions. Our exposition is organized as follows. In Section 2, a literature review of prototype-based classification is given. It ranges from the classical nearest neighbor classifier to the nearest feature space classifier, reviewing also modifications of the distance measure and several editing and condensing methods. In addition, we provide a detailed discussion on the modified versions of the nearest feature classifiers, mentioning which issues are still susceptible to be improved. The framework of dissimilarity representations and dissimilarity-based classification is presented in Section 3. We present three approaches for generalizing dissimilarity representations, including our own proposal for generalizing them by using feature lines and feature planes. Finally, a general discussion, overall conclusions and opportunities for future work are given in Section 4.

2. Prototype-based face recognition

Several taxonomies for pattern classification methods have been proposed. For instance, according to the chosen representation, there is a dichotomy between structural and statistical pattern recognition (Bunke & Sanfeliu, 1990; Jain et al., 2000; Pękalska & Duin, 2005a). According to the criterion to make the decision, classification approaches are divided into density-based and distance-based methods (Duda et al., 2001). Similarly, another commonly-stated division separates parametric and nonparametric methods. This last dichotomy is important for our discussion on prototype based face recognition.

Parametric methods include discriminant functions or decision boundaries with a predefined form, e.g. hyperplanes, for which a number of unknown parameters are estimated and plugged into the model. In contrast, nonparametric methods do not predefine a model for the decision boundary; conversely, such a boundary is directly constructed from the training data or generated by an estimation of the density function. The first type of nonparametric approaches encompasses the prototype based classifiers; a typical example of the second type is the Parzen window method.

Prototype based classifiers share the principle of keeping copies of training vectors in memory and constructing a decision boundary according to the distances between the stored prototypes and the query objects to be classified (Laaksonen, 1997). Either the whole training feature vectors are retained or a representative subset of them is extracted to be prototypes. Moreover, those prototypes or training objects can be used to generate new representative objects which were not originally included in the training set. Representations of new objects are not restricted to feature points; they might be lines, planes or even other functions or models based on the prototypes such as clusters or hidden Markov models (HMMs).

2.1 The nearest neighbor classifier

The simplest nonparametric method for classification should be considered k-NN (Cover & Hart, 1967). Its first derivation and fundamental theoretical properties gave origin to an entire family of classification methods, see Fig. 1. This rule classifies x by assigning it the class label ĉ most frequently represented among the k nearest prototypes; i.e., by finding the k neighbors with the minimum distances between x and all prototype feature points $\{x_{ci}, 1 \le c \le C, 1 \le i \le n_c\}$. For k=1, the rule can be written as follows:

$$d\left(x, x_{\hat{c}\hat{i}}\right) = \min_{1 \le c \le C;\, 1 \le i \le n_c} d\left(x, x_{ci}\right), \tag{1}$$

where $d(x,x_{ci})=\|x-x_{ci}\|$ is usually the Euclidean norm. In this case, the number of distance calculations is $n = \sum_{c=1}^{C} n_c$.

The k-NN method has been successfully used in a considerable variety of applications and has an optimal asymptotical behavior in the Bayes sense (Devroye et al., 1996); nonetheless, it requires a significant amount of storage and computational effort. Such a problem can be partly solved by using the condensed nearest neighbor rule (CNN) (Hart, 1968). In addition, the k-NN classifier suffers of a potential loss of accuracy when a small set of prototypes is available. To overcome this shortcoming, many variations of the k-NN method were developed, including the so-called nearest feature classifiers. Such methods derived from the original k-NN rule can be organized in a family of prototype-based classifiers as shown in Fig. 1.

2.2 Adaptive distance measures for the nearest neighbor rule

In order to identify the nearest neighbor, a distance measure has to be defined. Typically, a Euclidean distance is assumed by default. The use of other Minkowski distances such as Manhattan and Chebyshev is also convenient, not just for interpretability but also for computational convenience. In spite of the asymptotical optimality of the k-NN rule, we never have access to an unlimited number of samples. Consequently, the performance of the k-NN rule is always influenced by the chosen metric.

Several methods for locally adapting the distance measure have been proposed. Such an adaptation is probabilistically interpreted as an attempt to produce a neighborhood with an

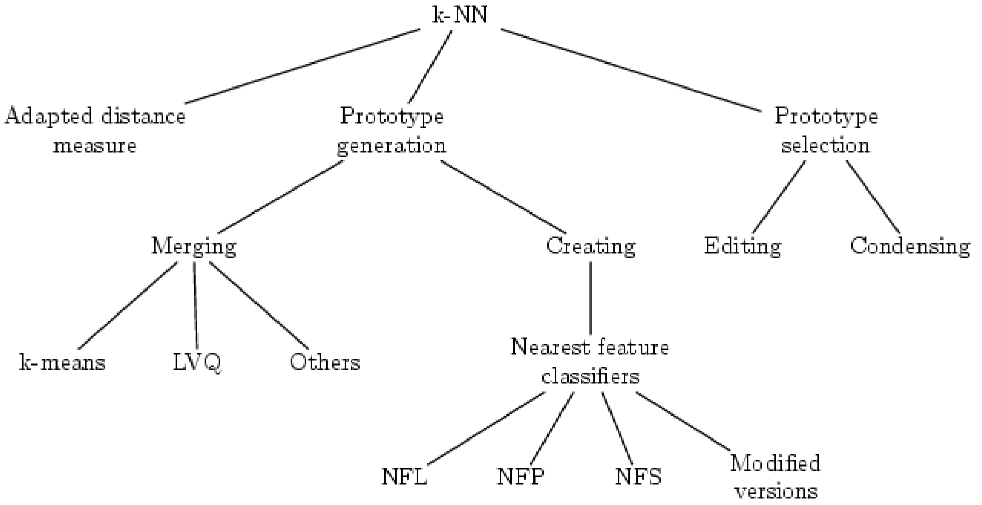


Fig. 1. Family of prototype-based classifiers.

a posteriori probability approximately constant (Wang et al., 2007). Among the methods aimed to local adaptation, the following must be mentioned:

a. The flexible or customized metric developed by Friedman (1994). Such a metric makes use of the information about the relative relevance of each feature. As a result, a new method is generated as a hybrid between the original k-NN rule and the tree-structured recursive partitioning techniques.

b. The adaptive metric method by Domeniconi et al. (2002). They use a χ^2 distance analysis to compute a flexible metric for producing neighborhoods that are adaptive to query locations. As a result, neighborhoods are constricted along the most relevant features and elongated along the less relevant ones. Such a modification locally influences class conditional probabilities, making them smother in the modified neighborhoods.

c. Approaches for learning distance metrics directly from the training examples. In (Goldberger et al., 2004), it was proposed a method for learning a Mahalanobis distance by maximizing a stochastic variation of the k-NN leave-one-out error. Similarly, Weinberger et al. (2005) proposed a method for learning a Mahalanobis distance by applying semidefinite programming. These concepts are close to approaches for building trainable similarity measures. See for example (Paclík et al., 2006b; Paclík et al., 2006a).

d. A simple adaptive k-NN classification algorithm based on the concept of statistical confidence (Wang et al., 2005; Wang et al., 2006). This approach involves a local adaptation of the distance measure, similarly to the other methods mentioned above. However, this method also includes a weighting procedure to assign a weight to each nearest neighbor according to its statistical confidence.

e. In (Wang et al., 2007), the same authors of the adaptation by using statistical confidence proposed a simple and elegant approach based on a normalization of the Euclidean or Manhattan distance from a query point to each training point by the shortest distance between the corresponding training point to training points of a different class. Such a new normalized distance is not symmetric and therefore is generally not a metric.

f. Other adaptations of the distance and modifications of the rule include the works by Hastie & Tibshirani (1996), Sánchez et al. (1998), Wilson & Martínez (1997), Avesani et al. (1999) and Paredes & Vidal (2000).

2.3 Prototype generation

The nearest neighbor classifier is sensitive to outliers, e.g. erroneously chosen, atypical or noisy prototypes. In order to overcome this drawback, several techniques have been proposed to tackle the problem of prototype optimization (Pękalska et al., 2006). A fundamental dichotomy in prototype optimization divides the approaches into prototype generation and prototype selection. In this section we will discuss some techniques of the first group. Prototype selection techniques are reviewed in §2.4.

Prototype generation techniques are fundamentally based on two operations on an initial set of prototypes: first, merging and averaging the initial set of prototypes in order to obtain a smaller set which optimizes the performance of the k-NN rule or; second, creating a larger set by creating new prototypes or even new functions or models generated by the initial set, see also Fig. 1. In this section we refer only to merging techniques. The second group — which includes the nearest feature classifiers— deserves a separated section. Examples of merging techniques are the following:

a. The k-means algorithm (MacQueen, 1967; Duda et al., 2001). It is considered the simplest clustering algorithm. Applied to prototype optimization, this technique aims to find a subset of prototypes generated from the original ones. New prototypes are the means of a number of partitions found by merging the original prototypes into a desired number of clusters. The algorithm starts by partitioning the original representation or prototype set $R=\{p_1, p_2, \ldots p_N\}$ into M initial sets. Afterwards, the mean point, or centroid, for each set is calculated. Then, a new partition is constructed by associating each prototype with the nearest centroid. Means are recomputed for the new clusters. The algorithm is repeated until it converges to a stable solution; that is, when prototypes no longer switch clusters. It is equivalent to observe no changes in the value of means or centroids. Finally, the new set of merged or averaged prototypes is composed by the M means: $R_\mu=\{\mu_1, \mu_2, \ldots, \mu_M\}$.

b. The learning vector quantization (LVQ) algorithm (Kohonen, 1995). It consists in moving a fixed number M of prototypes p_i towards to or away from the training points x_i. The set of generated prototypes is also called *codebook*. Prototypes are iteratively updated according to a learning rule. In the original learning process, the delta rule is used to update prototypes by adding a fraction of the difference between the current value of the prototype and a new training point x. The rule can be written as follows:

$$p_i(t+1)=p_i(t) + \alpha(t)[x(t) - p_i(t)], \tag{2}$$

where α controls the learning rate. Positive values of α move p_i towards x; conversely, negative values move p_i away from x.

In statistical terms, the LVQ learning process can be interpreted as a way to generate a set of prototypes whose density reflects the shape of a function s defined as (Laaksonen, 1997):

$$s(x) = P_j f_j(x) - \max_{k \neq j} P_k f_k(x), \tag{3}$$

where P_j and f_j are the a priori probability and the probability density functions of class j, respectively. See (Holmström et al., 1996) and (Holmström et al., 1997) for further details.

c. Other methods for generating prototypes include the learning k-NN classifier (Laaksonen & Oja, 1996), neuralnet-based methods for constructing optimized prototypes (Huang et al., 2002) and cluster-based prototype merging procedures, e.g. the work by Mollineda et al. (2002).

2.4 Nearest feature classifiers

The nearest feature classifiers are geometrical extensions of the nearest neighbor rule. They are based on a measure of distance between the query point and a function calculated from the prototypes, such as a line, a plane or a space. In this work, we review three different nearest feature rules: the nearest feature line or NFL, the nearest feature plane or NFP and the nearest feature space or NFS. Their natural extensions by majority voting are the k nearest feature line rule, or k-NFL, and the k nearest feature plane rule, or k-NFP (Orozco-Alzate & Castellanos-Domínguez, 2006). Two recent improvements of NFL and NFP are also discussed here: the rectified nearest feature line segment (RNFLS) and the genetic nearest feature plane (G-NFP), respectively.

Nearest Feature Line

The *k nearest feature line* rule, or k-NFL (Li & Lu, 1999), is an extension of the k-NN classifier. This method generalizes each pair of prototype feature points belonging to the same class: $\{x_{ci}, x_{cj}\}$ by a linear function L_{ij}^{c}, which is called *feature line* (see Fig. 2). The line is expressed by the span $L_{ij}^{c} = sp(x_{ci}, x_{cj})$. The query x is projected onto L_{ij}^{c} as a point p_{ij}^{c}. This projection is computed as

$$p_{ij}^{c} = x_{ci} + \tau(x_{cj} - x_{ci}), \tag{4}$$

where $\tau=(x-x_{ci})(x_{cj}-x_{ci})/\|x_{cj}-x_{ci}\|^2$. Parameter τ is called the position parameter. When $0<\tau<1$, p_{ij}^{c} is in the interpolating part of the feature line; when $\tau>1$, p_{ij}^{c} is in the forward extrapolating side and; when $\tau<0$, p_{ij}^{c} is in the backward extrapolating part. The two special cases when the query point is exactly projected on top of one of the points generating the feature line correspond to $\tau= 0$ and $\tau=1$. In such cases, $p_{ij}^{c} = x_{ci}$ and $p_{ij}^{c} = x_{cj}$, respectively. The classification of x is done by assigning it the class label ĉ most frequently represented among the k nearest feature lines, for k=1 that means:

$$d\left(x, L_{\hat{i}\hat{j}}^{\hat{c}}\right) = \min_{1 \le c \le C;\ 1 \le i,j \le n_c;\ i \ne j} d\left(x, L_{ij}^{c}\right), \tag{5}$$

where $d\left(x, L_{ij}^{c}\right) = \left\|x - p_{ij}^{c}\right\|$. In this case, the number of distance calculations is:

$$n_L = \sum_{c=1}^{C} n_c(n_c - 1)/2 \tag{6}$$

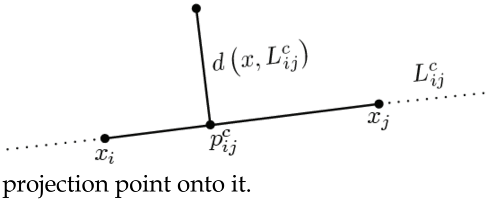


Fig. 2. Feature line and projection point onto it.

The nearest feature line classifier is supposed to deal with variations such as changes in viewpoint, illumination and face expression (Zhou et al., 2000). Such variations correspond to new conditions which were possibly not represented in the available prototypes. Consequently, the k-NFL classifier expands the representational capacity of the originally available feature points. Some typical variations in image faces taken from the *Biometric System Lab* data set (Cappelli et al., 2002) are shown in Fig. 3.

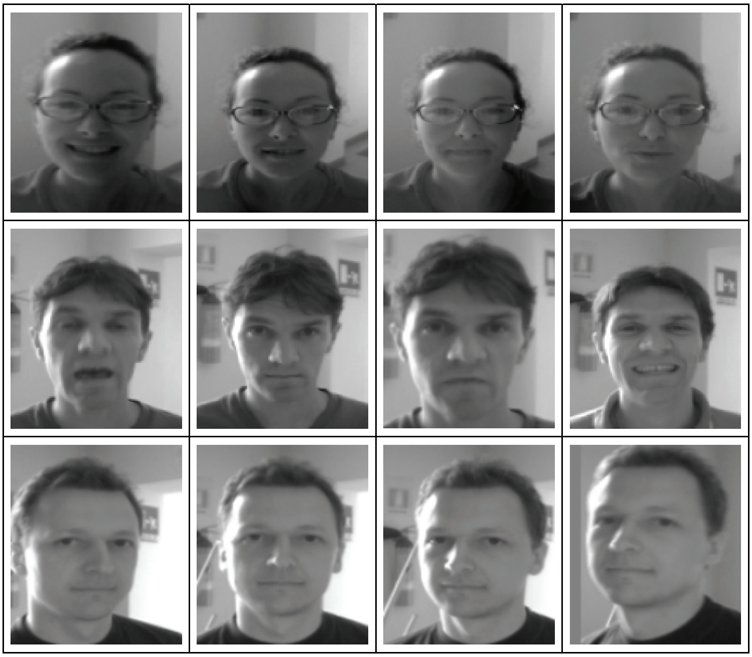


Fig. 3. Samples from the *Biometric System Lab* face dataset. Typical variations in face images are illustrated: illumination (first row), expression (second row) and pose (third row).

Nearest Feature Plane

The k nearest feature plane rule (Chien & Wu, 2002), or k-NFP, is an extension of the k-NFL classifier. This classifier assumes that at least three linearly independent prototype points are available for each class. It generalizes three feature points $\{x_{ci},x_{cj},x_{cm}\}$ of the same class by a feature plane F_{ijm}^{c} (see Fig. 4); which is expressed by the span $F_{ijm}^{c} = sp\left(x_{ci}, x_{cj}, x_{cm}\right)$. The query x is projected onto F_{ijm}^{c} as a point p_{ijm}^{c}. See Fig. 4. The projection point can be calculated as follows:

$$p_{ijm}^{c} = X_{ijm}^{c}\left({X_{ijm}^{c}}^{T} X_{ijm}^{c}\right)^{-1} {X_{ijm}^{c}}^{T} x, \tag{7}$$

where $X_{ijm}^{c} = \left[x_{ci}\ x_{cj}\ x_{cm}\right]$. Considering k=1, the query point x is classified by assigning it the class label ĉ, according to

$$d\left(x, F_{\hat{i}\hat{j}\hat{m}}^{\hat{c}}\right) = \min_{1 \le c \le C;\ 1 \le i,j,m \le n_c;\ i \ne j \ne m} d\left(x, F_{ijm}^{c}\right), \tag{8}$$

where $d\left(x, F_{ijm}^{c}\right) = \left\| x - p_{ijm}^{c} \right\|$. In this case, the number of distance calculations is:

$$n_F = \sum\nolimits_{c=1}^{C} n_c (n_c - 1)(n_c - 2)/6 \tag{9}$$

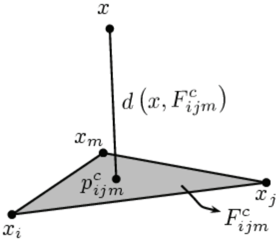


Fig. 4. Feature plane and projection point onto it.

Nearest Feature Space

The *nearest feature space* rule (Chien & Wu, 2002), or NFS, extends the geometrical concept of k-NFP classifier. It generalizes the independent prototypes belonging to the same class by a feature space $S^c = sp\left(x_{c1}, x_{c2}, \ldots, x_{cn_c}\right)$. The query point x is projected onto the C spaces as follows

$$p^c = X^c \left(X^{c^T} X^c\right)^{-1} X^{c^T} x, \tag{10}$$

where $X^c = \left[x_{c1}\ x_{c2} \cdots x_{cn_c}\right]$. The query point x is classified by assigning it the class label ĉ, according to

$$d\left(x, S^{\hat{c}}\right) = \min_{1 \le c \le C} d\left(x, S^c\right) = \min_{1 \le c \le C} \left\| x - p^c \right\| \tag{11}$$

Always, C distance calculations are required. It was geometrically shown in (Chien & Wu, 2002) that the distance of x to F_{ijm}^{c} is smaller than that to the feature line. Moreover, the distance to the feature line is nearer compared with the distance to two prototype feature points. This relation can be written as follows:

$$d\left(x, F_{ijm}^{c}\right) \le \min\left(d\left(x, L_{ij}^{c}\right), d\left(x, L_{jm}^{c}\right), d\left(x, L_{mi}^{c}\right)\right) \le \min\left(d\left(x, x_{ci}\right), d\left(x, x_{cj}\right), d\left(x, x_{cm}\right)\right) \tag{12}$$

In addition,

$$d\left(x, S^c\right) = \min_{1 \le c \le C} d\left(x, F_{ijm}^{c}\right) \tag{13}$$

In consequence, k-NFL classifier is supposed to capture more variations than k-NN, k-NFP should handle more variations of each class than k-NFL and NFS should capture more variations than k-NFP. So, it is expected that k-NFL performs better than k-NN, k-NFP is more accurate than k-NFL and NFS outperforms k-NFP.

Rectified Nearest Feature Line Segment

Recently, two main drawbacks of the NFL classifier have been pointed out: extrapolation and interpolation inaccuracies. The first one was discussed by (Zheng et al., 2004), who also proposed a solution termed as nearest neighbor line (NNL). The second one —interpolation inaccuracy— was considered in (Du & Chen, 2007). They proposed an elegant solution called *rectified nearest feature line segment* (RNFLS). Their idea is aimed to overcome not just the interpolation problems but also the detrimental effects produced by the extrapolation inaccuracy. In the subsequent paragraphs, we will discuss both inaccuracies and the RNFLS classifier.

Extrapolation inaccuracy. It is a major shortcoming in low dimensional feature spaces. Nonetheless, its harm is limited in higher dimensional ones such those generated by pixel-based representations for face recognition. Indeed, several studies related to NFL applied to high dimensional feature spaces have reported improvements in classification performance; see for instance (Li, 2000; Li et al., 2000; Orozco-Alzate & Castellanos-Domínguez, 2006; Orozco-Alzate & Castellanos-Domínguez, 2007). In brief, the extrapolation inaccuracy occurs when the query point is far from the two points generating the feature line L but, at the same time, the query is close to the extrapolating part of L. In such a case, classification is very likely to be erroneous. Du & Chen (2007) mathematically proved for a two-class problem that the probability that a feature line L^1 (first class) trespasses the region R_2 (second class) asymptotically approaches to 0 as the dimension becomes large. See (Du & Chen, 2007) for further details.

Interpolation inaccuracy. This drawback arises in multi-modal classification problems. That is, when one class c_i has more than one cluster and the territory between two of them belongs to another class c_j, $i \neq j$. In such a case, a feature line linking two points of the multi-modal class will trespass the territory of another class. Consequently, a query point located near to the interpolating part of the feature line might be erroneously assigned to the class of the feature line.

As we stated before, the two above-mentioned drawbacks are overcome by the so-called rectified nearest feature line segment. It consists in a two step correction procedure for the original k-NFL classifier. Such steps are a segmentation followed by a rectification. Segmentation consists in cutting off the feature line in order to preserve only the interpolating part which is called a feature line segment $\tilde{L}^c_{ij}$. Segmentation is aimed to avoid the extrapolation inaccuracy. When the orthogonal projection of a query point onto L^c_{ij} is in the interpolating part; that is, in $\tilde{L}^c_{ij}$, the distance of such query point to $\tilde{L}^c_{ij}$ is computed in the same way that the distance to L^c_{ij}, i.e. according to Eqs. (4) and (5). In contrast, when the projection point p_{ij}^c is in the extrapolating part, the distance to $\tilde{L}^c_{ij}$ is forced to be equal to the distance of the query point to one of the extreme point of $\tilde{L}^c_{ij}$: x_{ci} if p_{ij}^c is in the backward extrapolating part and x_{cj} if p_{ij}^c is the forward extrapolating part, respectively. See Fig. 5.

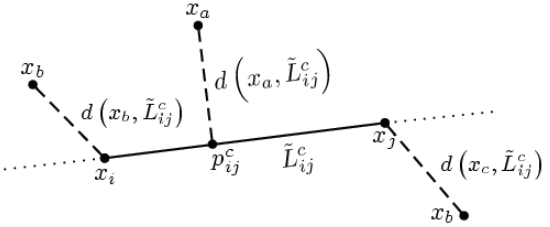

x_a
x_b
$d\left(x_a, \tilde{L}^c_{ij}\right)$
$d\left(x_b, \tilde{L}^c_{ij}\right)$
x_j
x_i
p^c_{ij}
$\tilde{L}^c_{ij}$
$d\left(x_c, \tilde{L}^c_{ij}\right)$
x_b

Fig. 5. Feature line segment and distances to it for three cases: projection point in the interpolating part, projection point in the backward extrapolating part and projection point in the forward extrapolating part.

Afterwards, the rectification procedure is achieved in order to avoid the effect of the interpolation inaccuracy. It consists in removing feature line segments trespassing the territory of another class. To do so, the concept of territory must be defined. Indeed, Du & Chen (2007) define two types of territory. The first one is called *sample territory* which, for a particular feature point x, stands for a ball centered at x with a radius equals to the distance from x to its nearest neighbor belonging to a different class. The second one —the *class territory*— is defined as the union of all sample territories of feature points belonging to the same class. Then, for each feature line segment, we check if it trespass the class territory of another class or not. In the affirmative case, we proceed to remove that feature line segment from the representation set. Finally, classification is performed in a similar way to Eq. (5) but replacing feature lines L_{ij}^{c} by those feature line segments $\widetilde{L}_{ij}^{c}$ which were not removed during the rectification.

Center-based Nearest Neighbor Classifier and Genetic Nearest Feature Plane

The k-NFL and k-NFP classifiers tend to be computationally unfeasible as the number of training objects per class grows. Such a situation is caused by to the combinatorial increase of combinations of two and three feature points, see Eqs. (6) and (9). Some alternatives to overcome this drawback have been recently published. Particularly, the center-based nearest neighbor (CNN) classifier (Gao & Wang, 2007) and the genetic nearest feature plane (G-NFP) (Nanni & Lumini, 2007). The first one is aimed at reducing the computation cost of the k-NFL classifier by using only those feature lines linking two feature points of the same class and, simultaneously, passing by the centroid of that class. In such a way, only a few feature lines are kept (authors call them center-based lines) and computation time is therefore much lower. The G-NFP classifier is a hybrid method to reduce the computational complexity of the k-NFP classifier by using a genetic algorithm (GA). It consists in a GA-based prototype selection procedure followed by the conventional method to generate feature lines. Selected prototypes are centroids of a number of intra-class clusters found by the GA.

2.5 Prototype selection

Prototype selection methods aim at the reduction of the initial set of prototypes while maintaining an acceptable classification accuracy or even increasing it. There are two groups of prototypes selection methods as shown in Fig. 1. See also (Wilson & Martínez, 2000) and (Lozano et al., 2006). Editing methods (Wilson, 1972; Devijver & Kittler, 1982; Aha et al., 1991) remove noisy and/or close border prototypes in order to avoid overlapping and smoothing the resulting decision boundaries. In other words, they are intended to produce a subset of prototypes forming homogeneous clusters in the feature space. Condensing algorithms try to select a small subset of prototypes while preserving classification performance as good as possible. Condensing may involve just a pure selection of prototypes (Hart, 1968; Tomek, 1976; Toussaint et al., 1985; Dasarathy, 1990; Dasarathy, 1994) or include a modification of them (Chang, 1974; Chen & Józwik, 1996; Ainslie & Sánchez, 2002; Lozano et al., 2004a; Lozano et al., 2004b).

2.6 HMM-based sequence classification

There are two standard ways for classifying sequences using HMMs. The first one is referred to as ML_{OPC} (Maximum-likelihood, one per class HMM-based classification).

Assume that a particular object x, a face in our case of interest, is represented by a sequence O and that C HMMs, $\{\lambda^{(1)}, \lambda^{(2)}, \ldots,\lambda^{(C)}\}$, has been trained; i.e. there is one trained HMM per class. Thus, the sequence O is assigned to the class showing the highest likelihood:

$$\text{Class}(O)=\arg\max_c P(O\,|\,\lambda^{(c)}) \tag{14}$$

The likelihood $P(O, \lambda^{(c)})$ is the probability that the sequence O was generated by the model $\lambda^{(c)}$. The likelihood can be estimated by different methods such as the Baum-Welch estimation procedure (Baum et al., 1970) and the forward-backward procedure (Baum, 1970). This approach can be considered analogous to the nearest feature space classification (see §2.4), with the proximity measure defined by the likelihood function.

The second method, named as ML_{OPS} (Maximum-likelihood, one per sequence HMM-based classification) consists in training one HMM per each training sequence $O_i^{(c)}$, where c denotes the class label. Similarly to (14), it can be written as:

$$\text{Class}(O)=\arg\max_c(\arg\max_i P(O\,|\,\lambda_i^{(c)})\,) \tag{15}$$

This method is analogous to the nearest neighbor classifier. Compare (1) and (15).

3. Dissimilarity representations

In (Orozco-Alzate & Castellanos-Domínguez, 2007) we introduced the concepts of dissimilarity-based face recognition and their relationship with the nearest feature classifiers. For convenience of the reader and for the sake of self–containedness, we repeat here a part of our previously published discourse on this matter.

A dissimilarity representation of objects is based on their pairwise comparisons. Consider a representation set $R:=\{p_1,p_2,\ldots,p_n\}$ and a dissimilarity measure d. An object x is represented as a vector of the dissimilarities computed between x and the prototypes from R, i.e. $D(x,R)=[d(x,p_1),d(x,p_2),\ldots,d(x,p_n)]$. For a set T of N objects, it extends to an N×n dissimilarity matrix (Pękalska & Duin, 2005a):

$$D(T,R)=\begin{bmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & d_{nn} \end{bmatrix}, \tag{16}$$

where $d_{jk}=D(x_j,p_k)$.

For dissimilarities, the geometry is contained in the definition, giving the possibility to include physical background knowledge; in contrast, feature-based representations usually suppose a Euclidean geometry. Important properties of dissimilarity matrices, such as metric nature, tests for Euclidean behavior, transformations and corrections of non-Euclidean dissimilarities and embeddings, are discussed in (Pękalska & Duin, 2005b).

When the entire T is used as R, the dissimilarity representation is expressed as an N×N dissimilarity matrix D(T,T). Nonetheless, R may be properly chosen by prototype selection procedures. See §2.5 and (Pękalska et al., 2006).

3.1 Classifiers in dissimilarity spaces

Building a classifier in a dissimilarity space consists in applying a traditional classification rule, considering dissimilarities as features; it means, in practice, that a dissimilarity-based classification problem is addressed as a traditional feature-based one. Even though the nearest neighbor rule is the reference method to discriminate between objects represented by dissimilarities, it suffers from a number of limitations. Previous studies (Pękalska et al., 2001; Pękalska & Duin, 2002; Paclík & Duin, 2003; Pękalska et al., 2004; Orozco-Alzate et al., 2006) have shown that Bayesian (normal density based) classifiers, particularly the linear (LDC) and quadratic (QDC) normal based classifiers, perform well in dissimilarity spaces and, sometimes, offer a more accurate solution. For a 2-class problem, the LDC based on the representation set R is given by

$$\mathrm{f}(\mathrm{D}(\mathrm{x},\mathrm{R}))=\left[\mathrm{D}(\mathrm{x},\mathrm{R})-\frac{1}{2}\left(\mathbf{m}_{(1)}+\mathbf{m}_{(2)}\right)\right]^{\mathrm{T}}\times \mathrm{C}^{-1}\left(\mathbf{m}_{(1)}-\mathbf{m}_{(2)}\right)+\log\frac{\mathrm{P}_{(1)}}{\mathrm{P}_{(2)}} \tag{17}$$

and the QDC is derived as

$$\mathrm{f}(\mathrm{D}(\mathrm{x},\mathrm{R}))=\sum_{\mathrm{i}=1}^{2}(-1)^{\mathrm{i}}\left(\mathrm{D}(\mathrm{x},\mathrm{R})-\mathbf{m}_{(\mathrm{i})}\right)^{\mathrm{T}}\times \mathrm{C}_{(\mathrm{i})}^{-1}\left(\mathrm{D}(\mathrm{x},\mathrm{R})-\mathbf{m}_{(\mathrm{i})}\right)+2\log\frac{\mathrm{p}_{(1)}}{\mathrm{p}_{(2)}}+\log\frac{\left|\mathrm{C}_{(1)}\right|}{\left|\mathrm{C}_{2}\right|} \tag{18}$$

where C is the sample covariance matrix, $\mathrm{C}_{(1)}$ and $\mathrm{C}_{(2)}$ are the estimated class covariance matrices, and $\mathbf{m}_{(1)}$ and $\mathbf{m}_{(2)}$ are the mean vectors, computed in the dissimilarity space D(T,R). $\mathrm{P}_{(1)}$ and $\mathrm{P}_{(2)}$ are the class prior probabilities. If C is singular, a regularized version must be used. In practice, the following regularization is suggested for r=0.01 (Pękalska et al., 2006):

$$\mathrm{C}_{\mathrm{reg}}^{\mathrm{r}}=(1-\mathrm{r})\mathrm{C}+\mathrm{r}\,\mathrm{diag}(\mathrm{C}) \tag{19}$$

Nonetheless, regularization parameter should be optimized in order to obtain the best possible results for the normal density based classifiers.

Other classifiers can be implemented in dissimilarity spaces, usually by a straightforward implementation. Nearest mean linear classifiers, Fisher linear discriminants, support vector machines (SVMs), among others are particularly interesting for being used in generalized dissimilarity spaces. In addition, traditional as well as specially derived clustering techniques can be implemented for dissimilarity representations, see (Pękalska & Duin, 2005c) for a detailed discussion on clustering techniques in dissimilarity representations.

3.2 Generalization of dissimilarity representations

Dissimilarity representations were originally formulated as pairwise constructs derived by object to object comparisons. Nonetheless, it is also possible to define them in a wider form, e.g. defining representations based on dissimilarities with functions of (or models built by) objects. In the general case, representation objects used for building those functions or models do not need labels, allowing for semi-supervised approaches in which the unlabeled objects are used for the representation and not directly for the classifier, or might even be artificially created, selected by an expert or belong to different classes than the ones under consideration (Duin, 2008).

We phrase such a wider formulation as a *generalized dissimilarity representation*. In spite of the potential to omit labels, to the best of the authors' knowledge, all the current generalization procedures —including ours— make use of labels. At least three different approaches for generalizing dissimilarity representations have been proposed and developed independently: generalization by using hidden Markov models (Bicego et al., 2004), generalization by pre-clustering (Kim, 2006) and our own proposal of generalizing dissimilarity representations by using feature lines and feature planes. In this subsection, we discuss the first two approaches, focusing particularly in their motivations and methodological principles. The last one is discuss in §3.3.

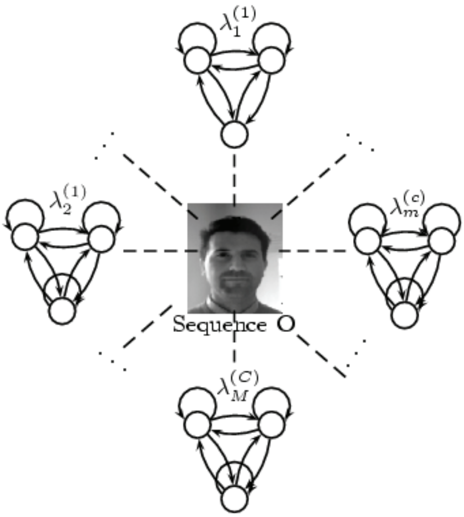
Dissimilarity-based Classification Using HMMs

It can be easily seen that likelihoods $P(O|\lambda^{(c)})$ and/or $P(O|\lambda_i^{(c)})$ can be interpreted as similarities; e.g. Bicego et al. (2004) propose to use the following similarity measure between two sequences O_i and O_j:

$$d_{ij} = d(O_i,O_j)=\log P(O_i|\lambda_j)/T_i \qquad (20)$$

where T_i is the length of the sequence O_i, introduced as a normalization factor to make a fair comparison between sequences of different length. Notice that, even though we use d to denote a dissimilarity measure, in (20) we are in fact referring to a similarity. Nonetheless, the two concepts are closely related and even used indistinctively as in (Bicego et al., 2004). In addition, there exist some ways of changing a similarity value into a dissimilarity value and vice versa (Pękalska & Duin, 2005a).

A HMM-based dissimilarity might be derived by measuring the likelihood between all pairs of sequences and HMMs. Consider a representation set $R=\{O_1^{(1)},O_2^{(1)},\ldots,O_M^{(C)}\}$. A dissimilarity representation for a new sequence O is given in terms of the likelihood of having generated O with the associated HMMs for each sequence in R. Those HMMs are grouped in the representation set $R_\lambda=\{\lambda_1^{(1)},\lambda_2^{(1)},\ldots,\lambda_M^{(C)}\}$. In summary, the sequence O is represented by the following vector: $D(O,R_\lambda)=[d_1 \;\; d_2 \cdots d_M]$. For a training set $T=\{O_1,O_2,\ldots, O_N\}$, it extends to a matrix $D(T,R_\lambda)$ as shown in Fig. 6.



$$D(T,R_\lambda) = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \\ \vdots \\ O_N \end{array} \begin{array}{c} \begin{array}{ccccccc} \lambda_1^{(1)} & \lambda_2^{(1)} & \cdots & \lambda_m^{(c)} & \cdots & \lambda_M^{(C)} \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} & \cdots & d_{1M} \\ d_{21} & d_{12} & \cdots & d_{2m} & \cdots & d_{2M} \\ d_{31} & d_{32} & \cdots & d_{3m} & \cdots & d_{3M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{Nm} & \cdots & d_{NM} \end{bmatrix} \end{array}$$

$$D(O,R_\lambda)=\begin{bmatrix} d_1 & d_2 & \cdots & d_m & \cdots & d_M \end{bmatrix}$$

Fig. 6. Generalization of a dissimilarity representation by using HMMs.

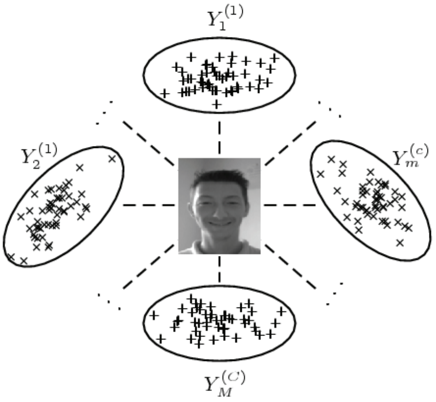
On top of the generalized dissimilarity representation D(T,R_λ), a dissimilarity-based classifier can be built.

Generalization by Clustering Prototypes

Kim (2006) proposed a methodology to overcome the SSS problem in face recognition applications. In summary, the proposed approach consists in:

1. Select a representation set R from the training set T.
2. Compute a dissimilarity representation D(T,R) by using some suitable dissimilarity measure.
3. For each class, perform a clustering of R into a few subsets $Y_m^{(c)}$, c=1,…,C and i= m,…, M; that is, M clusters of objects belonging to the same class. Any clustering method can be used; afterwards, the M mean vectors $\hat{Y}_i^{(c)}$, are computed by averaging each cluster.
4. A dissimilarity based classifier is built in D(T,R_Y). Moreover, a fusion technique may be used in order to increase the classification accuracy.

Kim's attempt to reduce dimensionality by choosing means of clusters as representatives can be interpreted as a generalization procedure. Similarly to the case of generalization by HMMs, this generalization procedure by clustering prototypes is schematically shown in Fig. 7.



$$D(T,R_\lambda) = \begin{array}{c} \\ O_1 \\ O_2 \\ O_3 \\ \vdots \\ O_N \end{array} \begin{array}{c} \begin{array}{ccccc} \lambda_1^{(1)} & \lambda_2^{(1)} & \cdots & \lambda_m^{(c)} & \cdots & \lambda_M^{(C)} \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} & \cdots & d_{1M} \\ d_{21} & d_{12} & \cdots & d_{2m} & \cdots & d_{2M} \\ d_{31} & d_{32} & \cdots & d_{3m} & \cdots & d_{3M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & d_{Nm} & \cdots & d_{NM} \end{bmatrix} \end{array}$$

$$D(O,R_\lambda) = \begin{bmatrix} d_1 & d_2 & \cdots & d_m & \cdots & d_M \end{bmatrix}$$

Fig. 7. Generalization of dissimilarity representations by clustering prototypes.

As we explained above, our generalization method by feature lines and feature planes can be included in the family of model- or function-based generalization procedures. It will be presented in detail in the subsequent section. For the sake of comparison, here we point to a few relevant coincidences and differences between the generalizations by HMMs and clustering and our proposed approach by feature lines and feature planes. See Fig. 8 and compare it against Figs. 6 and 7. Firstly, notice that the representation role is played in our case by a function generated by two representative objects, e.g. the so-called feature lines: $\{L_m^c\}$. A given object x is now represented in terms of its dissimilarities to a set R_L of representative feature lines. It also extends to D(T,R_L) for an entire training set. One remarkable difference is that our approach, in principle, leads to a higher dimensional space; i.e. M > N. In contrast, HMM- and cluster-based approaches leads in general to low dimensional spaces: M < N.

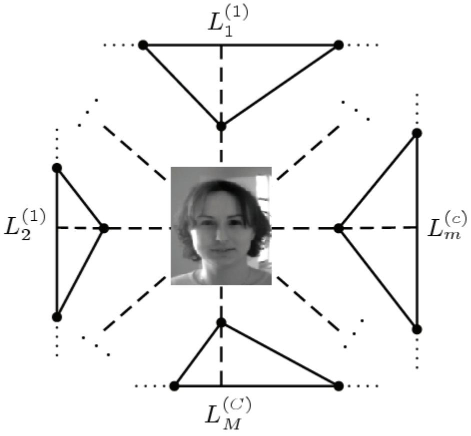


Fig. 8. Generalization of dissimilarity representations by feature lines.

3.3 Generalization by feature lines and feature planes

Our generalization consists in creating matrices $D_L(T,R_L)$ and $D_F(T,R_F)$ by using the information available at the original representation D(T,R), where subindexes L and F stand for feature lines and feature planes respectively. This generalization procedure was proposed in (Orozco-Alzate & Castellanos-Domínguez, 2007) and (Orozco-Alzate et al., 2007a). In this section, we review our method as it was reported in the above-mentioned references but also including some results and remarks resulted from our most recent discussions and experiments.

$D_L(T,R_L)$ and $D_F(T,R_F)$ are called *generalized dissimilarity representations* and their structures are:

$$D_L\left(T,R_L\right)=\begin{array}{c|ccccc|}
 & L_1 & L_2 & L_3 & \cdots & L_n \\
x_1 & d_{11} & d_{12} & d_{13} & \cdots & d_{1n_L} \\
x_2 & d_{21} & d_{22} & d_{23} & \cdots & d_{2n_L} \\
x_3 & d_{31} & d_{32} & d_{33} & \cdots & d_{3n_L} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_N & d_{N1} & d_{N2} & d_{N3} & \cdots & d_{Nn_L}
\end{array}, \tag{21}$$

where $d_{jk}=D_L(x_j,L_k)$; and

$$D_F\left(T,R_F\right)=\begin{array}{c|ccccc|}
 & F_1 & F_2 & F_3 & \cdots & F_n \\
x_1 & d_{11} & d_{12} & d_{13} & \cdots & d_{1n_F} \\
x_2 & d_{21} & d_{22} & d_{23} & \cdots & d_{2n_F} \\
x_3 & d_{31} & d_{32} & d_{33} & \cdots & d_{3n_F} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_N & d_{N1} & d_{N2} & d_{N3} & \cdots & d_{Nn_F}
\end{array}, \tag{22}$$

where $d_{jk}=D_F(x_j, F_k)$.

D(T, R_L) and D(T, R_F) are high dimensional matrices because the original representation set R is generalized by combining all the pairs (R_L) and all the triplets (R_F) of prototypes of the same class. Consequently, a proper procedure for prototype selection (dimensionality reduction) is needed in order to avoid the curse of the dimensionality. Another option is to use a strong regularization procedure. In general, all the possible dissimilarities between objects are available but the original feature points are not. Nonetheless, it is possible to compute the distances to feature lines from the dissimilarities. The problem consists in computing the height of a scalene triangle as shown in Figure 9.

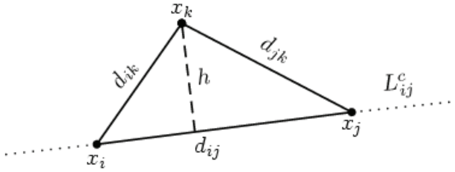


Fig. 9. Height of a scalene triangle corresponding to the distance to a feature line.

Let us define s=(d_{jk}+d_{ij}+d_{ik})/2. Then, the area of the triangle is given by:

$$A = \sqrt{s(s-d_{jk})(s-d_{ij})(s-d_{ik})}; \tag{23}$$

but it is also known that area, assuming d_{ij} as base, is:

$$A = \frac{d_{ij}h}{2} \tag{24}$$

So, we can solve (23) and (24) for h, which is the distance to the feature line. The generalized dissimilarity representation in (21) is constructed by replacing each entry of D(T,R_L) by the corresponding value of h. The distance d_{ij} in Fig. 9 must be an intraclass one.

Computing the distances to the feature planes in terms of dissimilarities consists in calculating the height of an irregular (scalene) tetrahedron as shown in Fig. 10.

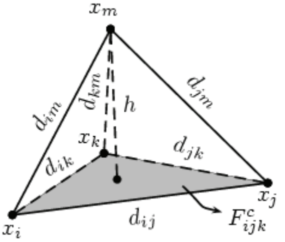


Fig. 10. Height of an irregular tetrahedron corresponding to the distance to a feature plane.

Let us define s=(d_{jk}+d_{ij}+d_{ik})/2. Then, the volume of a tetrahedron is given by:

$$V = \frac{h\sqrt{s(s-d_{jk})(s-d_{ij})(s-d_{ik})}}{3}; \tag{25}$$

but volume is also (Uspensky, 1948):

$$V^2 = \frac{1}{288}\begin{vmatrix} 0 & d_{ij}^2 & d_{ik}^2 & d_{im}^2 & 1 \\ d_{ij}^2 & 0 & d_{jk}^2 & d_{jm}^2 & 1 \\ d_{ik}^2 & d_{jk}^2 & 0 & d_{km}^2 & 1 \\ d_{im}^2 & d_{jm}^2 & d_{km}^2 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{vmatrix} \tag{26}$$

So, we can solve (25) and (26) for h, which is the distance to the feature plane. The generalized dissimilarity representation in (22) is constructed by replacing each entry of D(T,R$_F$) by the corresponding value of h. Distances d_{ij}, d_{ik} and d_{jk} in Fig. 10 must be intraclass.

Experiments (Orozco-Alzate and Castellanos-Domínguez, 2007; Orozco-Alzate et al., 2007a) showed that nearest feature rules are especially profitable when variations and conditions are not fully represented by the original prototypes; for example the case of small or non-representative training sets. The improvement in such a case respect to the reference method —the k-NN rule— is due to the feature lines/planes' ability to expand the representational capacity of the available points, accounting for new conditions not fully represented by the original set. Those are precisely the conditions in face recognition problems, where the number of prototypes is typically limited to few images per class and the number of classes is high: tens or even one hundred persons. As a result, the effectiveness of the nearest feature rules is remarkable for this problem.

Generalized dissimilarity representations by feature lines and feature planes are not square, having two or three zeros per column for feature lines and feature planes respectively. Firstly, we have considered generalizations of metric representations because the generalization procedure requires constructing triangles and tetrahedrons and, as a result, generalizing non-metric dissimilarity representations might produce complex numbers when solving equations for heights. An apparently promising alternative to take into account is the use of refining methods for non-metric dissimilarities; see (Duin & Pękalska, 2008). Such refining methods lead to pseudo-Euclidean embedded spaces or to the computation of distances on distances which is equivalent to apply the k-NN rule in the dissimilarity space.

In order to construct classifiers based on generalized dissimilarity representations, we should proceed similarly as dissimilarity-based classifiers are built; see §3.1. That is, using a training set T and a representation set R containing prototype examples from T. Prototype lines or planes considered must be selected by some prototype selection procedure; classifiers should be built on D(T,R$_L$) and D(T,R$_F$). Different sizes for the representation set R must be considered. In (Orozco-Alzate et al., 2007b), we proved an approach for selecting middle-length feature lines. Experiments showed that they are appropriate to represent moderately curved subspaces.

Enriching the dissimilarity representations implies a considerable number of calculations. The number of feature lines and planes grows rapidly as the number of prototypes per class increases; in consequence, computational effort may become high, especially if a generalized representation is computed for an entire set. When applying traditional statistical classifiers to dissimilarity representations, dissimilarities to prototypes may be treated as features. As a result, classifiers built in enriched dissimilarity spaces are also subject to the curse of dimensionality phenomenon. In general, for generalized dissimilarity representations

$D_g(T,R_g)$, the number of training objects is small relative to the number of prototype lines or planes.

According to the two reasons above, it is important to use dimensionality reduction techniques —feature extraction and feature selection methods— before building classifiers in generalized dissimilarity representations. Systematic approaches for prototype selection such as exhaustive search and the forward selection process lead to an optimal representation set; however, they require a considerable number of calculations. Consequently, due to the increased dimensionality of the enriched representations, the application of a systematic prototype selection method will be computationally expensive. Nonetheless, it has been shown that non-optimal and computationally simple procedures such as *Random* and *RandomC* may work well (Pękalska et al., 2006) and that simple geometrical criteria such the length of the lines (Orozco-Alzate et al., 2007b) may be wise for selecting representation subsets.

4. Conclusion

We started this chapter with a critic about the overwhelming research efforts in face recognition, discussing benefits and drawbacks of such a situation. The particular case of studies related to linear dimensionality reduction is a *per antonomasia* example of an enormous concentration of attention in a small and already mature area. We agree that research topics regarding preprocessing, classification and even nonlinear dimensionality reduction are much more promising and susceptible of significant contributions than further studies in LDA. Afterwards, we reviewed the state of the art in prototype-based classification for face recognition. Several techniques and variants have been proposed since the early days of the nearest neighbor classifier. Indeed, an entire family of prototype-based methods arose; some of the family members are entirely new ideas, others are modifications or hybrid methods. In brief, three approaches in prototype-based classification can be distinguished: modifications of the distance measure, prototype generation and prototype selection methods. The last two approaches present dichotomies as shown in Fig. 1. We focus our discussion on nearest feature classifiers and their improved versions as well as on their use in dissimilarity-based classification.

The main advantage of the RNFLS classifier is its property of generating feature line segments that are more concentrated in distribution than the original feature points. In addition, RNFLS corrects the interpolation and extrapolation inaccuracies of the k-NFL classifier, allowing us to use feature lines (in fact, feature line segments) in low dimensional classification problems. k-NFL was originally proposed and successfully used just in high dimensional representations such pixel-based representations for face recognition; however, thanks to the improvement provided by RNFLS, feature line-based approaches are also applicable now to feature-based face recognition. The k-NFP classifier is computationally very expensive. G-NFP can reduce the effort to a manageable amount, just by using a simple two-stage process: a GA-based prototype selection followed by the original k-NFP algorithm.

We presented and compare three different but related approaches to generalize dissimilarity representations by using HMMs, clustering techniques and feature lines/planes respectively. The first two are model-based extensions for a given dissimilarity matrix and lead, in general, to lower dimensional dissimilarity spaces. In contrast, our methodology produces high dimensional dissimilarity spaces and, consequently, proper prototype selection methods must be applied. Generalized dissimilarity representations by feature lines and feature planes are in fact enriched instead of condensed representations. Consequently, they have the property of accounting for entirely new information not originally available in the given dissimilarity matrix. Potential open issues for further research are alternative methods for feature line/plane selection such as sparse classifiers and linear programming-based approaches. In order to extend methods based on feature lines and feature planes to the case of indefinite representations, correction techniques for non-Euclidean data are also of interest.

5. References

Aha, D. W., Kibler, D. F., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1):37–66.

Ainslie, M. C. & Sánchez, J. S. (2002). Space partitioning for instance reduction in lazy learning algorithms. In *2nd International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, pages 13–18, Helsinki (Finland).

Avesani, P., Blanzieri, E., & Ricci, F. (1999). Advanced metrics for class-driven similarity search. In *DEXA '99: Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, page 223, Washington, DC, USA. IEEE Computer Society.

Baum, L. E. (1970). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequality*, 3:1–8.

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Bicego, M., Murino, V., & Figueiredo, M. A. T. (2004). Similarity-based classification of sequences using hidden Markov models. *Pattern Recognition*, 37(12):2281–2291.

Bunke, H. & Sanfeliu, A., editors (1990). *Syntactic and structural pattern recognition –Theory and applications*, volume 7 of *World Scientific Series in Computer Science*. World Scientific.

Cappelli, R., Maio, D., & Maltoni, D. (2002). Subspace classification for face recognition. In *Proceedings of the International Workshop on Biometric Authentication, (ECCV 2002 - Copenhagen)*, volume 2359/2002 of *Lecture Notes in Computer Science*, pages 133–142, London, UK. Springer-Verlag.

Chang, C.-L. (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Comput.*, 23(11):1179–1184.

Chen, C. H. & Józwik, A. (1996). A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters*, 17(8):819–823(5).

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

Cover, T. M. & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory*, IT-13(1):21–27.

Dasarathy, B. V. (1990). *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA.

Dasarathy, B. V. (1994). Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Trans. Systems, Man, and Cybernetics*, 24(3):511–517.

Delac, K. & Grgic, M., editors (2007). *Face Recognition*. Artificial Intelligence. I-TECH Education and Publishing, Vienna, Austria.

Devijver, P. A. & Kittler, J. (1982). *Pattern Recognition: a Statistical Approach*. Prentice Hall International, London.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, Berlin.

Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1281–1285.

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

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. Wiley-Interscience, New York, 2 edition.

Duin, R. P. W. (2008). Personal communication.

Duin, R. P. W., Loog, M., & Ho, T. K. (2006). Recent submissions in linear dimensionality reduction and face recognition. *Pattern Recognition Letters*, 27(7):707–708.

Duin, R. P. W. & Pękalska, E. (2008). On refining dissimilarity matrices for an improved NN learning. In *19th International Conference on Pattern Recognition ICPR 2008. Tampa, USA*. Accepted to be presented.

Friedman, J. H. (1994). Flexible metric nearest neighbor classification. Technical report, Department of Statistics, Stanford University.

Gao, Q.-B. & Wang, Z.-Z. (2007). Center-based nearest neighbor classifier. *Pattern Recognition*, 40(1):346–349.

Goldberger, J., Roweis, S., Hinton, G., & Salakhutdinov, R. (2004). Neighborhood component analysis. In Saul, L. K., Weiss, Y., & Bottou, L., editors, *Advances in Neural Information Processing Systems. Proceedings of the 17th Eighteenth Annual Conference on Neural Information Processing Systems NIPS2004*, volume 17, pages 513–520, Cambridge, MA. Neural Information Processing Systems Foundation, MIT Press.

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

Hastie, T. & Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification and regression. In Touretzky, D. S., Mozer, M. C., & Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 409–415. The MIT Press.

Holmström, L., Koistinen, P., Laaksonen, J., & Oja, E. (1996). Comparison of neural and statistical classifiers — theory and practice. Technical report a13, Rolf Nevanlinna Institute, Helsinki.

Holmström, L., Koistinen, P., Laaksonen, J., & Oja, E. (1997). Neural and statistical classifiers-taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8(1):5–17.

Huang, Y. S., Chiang, C. C., Shieh, J. W., & Grimson, W. E. L. (2002). Prototype optimization for nearest-neighbor classification. *Pattern Recognition*, 35(6):1237–1245.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(1):4–37.

Kim, S.-W. (2006). On using a dissimilarity representation method to solve the small sample size problem for face recognition. In Blanc-Talon, J., Philips, W., Popescu, D., & Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems, Proceedings of the 8th International Conference, ACIVS 2006*, volume 4179 of *Lecture Notes in Computer Science*, pages 1174–1185, Antwerp, Belgium. Springer.

Kohonen, T. (1995). *Self-Organizing Maps*. Number 30 in Information Sciences. Springer, Germany.

Laaksonen, J. (1997). *Subspace Classifiers in Recognition of Handwritten Digits*. PhD thesis, Helsinki University of Technology. http://lib.tkk.fi/Diss/199X/isbn9512254794/.

Laaksonen, J. & Oja, E. (1996). Classification with learning k-nearest neighbors. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN'96). Washington, DC, USA*, volume 3, pages 1480–1483. IEEE.

Li, S. Z. (2000). Content-based classification and retrieval of audio using the nearest feature line method. *IEEE Trans. Speech and Audio Process*, 8(5):619–625.

Li, S. Z. (2008). Nearest feature line. *Scholarpedia*, 3(3):4357. Available at http://www.scholarpedia.org/article/Nearest_feature_line.

Li, S. Z., Chan, K. L., & Wang, C. (2000). Performance evaluation of the nearest feature line method in image classification and retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(11):1335–1349.

Li, S. Z. & Lu, J. (1998). Generalizing capacity of face database for face recognition. In *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 402–407, Nara, Japan. IEEE Computer Society.

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

Liu, J., Chen, S., Tan, X., & Zhang, D. (2007). Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(8):1265–1278.

Lozano, M., Sánchez, J. S., & Pla, F. (2004a). Using the geometrical distribution of prototypes for training set condensing. In *Current Topics in Artificial Intelligence, Lecture Notes in Artificial Intelligence 3040*, pages 618–627. Springer-Verlag.

Lozano, M., Sotoca, J. M., Sánchez, J. S., & Pla, F. (2004b). An adaptive condensing algorithm based on mixtures of gaussians. In *7é Congrés Català d'Intel.ligència Artificial, Frontiers in Artificial Intelligence and Applications 113*, pages 225–232, Barcelona (Spain). IOS Press.

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

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. & Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Mollineda, R. A., Ferri, F. J., & Vidal, E. (2002). An efficient prototype merging strategy for the condensed 1-nn rule through class-conditional hierarchical clustering. *Pattern Recognition*, 35(12):2771–2782.

Nanni, L. & Lumini, A. (2007). Genetic nearest feature plane. *Expert Systems with Applications*. In Press, Corrected Proof.

Nhat, V. D. M. & Lee, S. (2007). Image-based subspace analysis for face recognition. In Delac, K. & Grgic, M., editors, *Face Recognition*, Artificial Intelligence, chapter 17, pages 321–336. I-TECH Education and Publishing, Vienna, Austria.

Orozco-Alzate, M. & Castellanos-Domínguez, C. G. (2006). Comparison of the nearest feature classifiers for face recognition. *Machine Vision and Applications (Springer)*, 17(5):279–285.

Orozco-Alzate, M. & Castellanos-Domínguez, C. G. (2007). Nearest feature rules and dissimilarity representations for face recognition problems. In Delac, K. & Grgic, M., editors, *Face Recognition*, Artificial Intelligence, chapter 18, pages 337–356. I-TECH Education and Publishing, Vienna, Austria.

Orozco-Alzate, M., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2007a). Generalizing dissimilarity representations using feature lines. In Rueda, L., Mery, D., & Kittler, J., editors, *Progress in Pattern Recognition, Image Analysis and Applications. Proceedings of the 12th Iberoamerican Congress on Pattern Recognition CIARP 2007*, volume 4756 of *Lecture Notes in Computer Science*, pages 370–379, Viña del Mar-Valparaiso, Chile. IAPR, Springer.

Orozco-Alzate, M., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2007b). 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.

Orozco-Alzate, M., García-Ocampo, M. E., Duin, R. P. W., & Castellanos-Domínguez, C. G. (2006). Dissimilarity-based classification of seismic volcanic signals at Nevado del Ruiz volcano. *Earth Sciences Research Journal*, 10(2):57–65.

Paclík, P. & Duin, R. P. W. (2003). Dissimilarity-based classification of spectra: computational issues. *Real Time Imaging*, 9:237–244.

Paclík, P., Novovicova, J., & Duin, R. P. W. (2006a). Building road sign classifiers using trainable similarity measure. *IEEE Trans. Intelligent Transportation Systems*, 7(3):293–308.

Paclík, P., Novovicova, J., & Duin, R. P. W. (2006b). A trainable similarity measure for image classification. In Tang, Y. Y., Wang, S. P., Lorette, G., Yeung, D. S., & Yan, H., editors, *Proc. of the 18th Int. Conf. on Pattern Recognition (ICPR2006, Hong Kong, China, August 2006)*, volume 3, pages 391–394, Los Alamitos. IEEE Computer Society Press.

Paredes, R. & Vidal, E. (2000). A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recogn. Lett.*, 21(12):1027–1036.

Pękalska, E. & Duin, R. P. W. (2002). Dissimilarity representations allow for building good classifiers. *Pattern Recognition Lett.*, 23:943–956.

Pękalska, E. & Duin, R. P. W. (2005a). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, volume 64 of *Machine Perception and Artificial Intelligence*. World Scientific, Singapore.

Pękalska, E. & Duin, R. P. W. (2005b). *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, volume 64 of *Machine Perception and Artificial Intelligence*, chapter 3, pages 89–145. World Scientific, Singapore.

Pękalska, E. & Duin, R. P. W. (2005c). Further data exploration. In *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*, chapter 7, pages 289–332. World Scientific.

Pękalska, E., Duin, R. P. W., Günter, S., & Bunke, H. (2004). On not making dissimilarities Euclidean. In *Proceedings of Structural and Statistical Pattern Recognition*, pages 1143–1151, Lisbon, Portugal.

Pękalska, E., Duin, R. P. W., & Paclík, P. (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208.

Pękalska, E., Paclík, P., & Duin, R. P. W. (2001). A generalized kernel approach to dissimilarity based classification. *J. Mach. Learn. Res.*, 2(2):175–211.

Ruiz-del Solar, J. & Navarrete, P. (2005). Eigenspace-based face recognition: a comparative study of different approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(3):315–325.

Sánchez, J. S., Pla, F., & Ferri, F. J. (1998). Improving the k-NCN classification rule through heuristic modifications. *Pattern Recognition Letters*, 19(13):1165–1170.

Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:679–772.

Toussaint, G. T., Bhattacharya, B. K., & Poulsen, R. S. (1985). The application of Voronoi diagrams to nonparametric decision rules. In Billard, L., editor, *Computer Science and Statistics: The Interface*. Elsevier, Amsterdam.

Uspensky, J. V. (1948). *Theory of Equations*. McGraw-Hill.

Wang, J., Neskovic, P., & Cooper, L. N. (2005). An adaptive nearest neighbor rule for classifications. In Yeung, D. S., Liu, Z. Q., Wang, X. Z., & Yan, H., editors, *Advances in Machine Learning and Cybernetics. Proceedings of the 4th International Conference on Machine Learning and Cybernetics ICMLC2005. Guangzhou, China*, volume 3930 of *Lecture Notes in Computer Science. Sublibrary: LNAI*, pages 548–557, Berlin. Springer.

Wang, J., Neskovic, P., & Cooper, L. N. (2006). Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern Recognition*, 39(3):417–423.

Wang, J., Neskovic, P., & Cooper, L. N. (2007). Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recogn. Lett.*, 28(2):207–213.

Weinberger, K., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In Weiss, Y., Schölkopf, B., & Platt, J., editors, *Advances in Neural Information Processing Systems. Proceedings of the 18th Annual Conference on Neural Information Processing Systems NIPS2005*, volume 18, pages 1473–1480, Cambridge, MA. Neural Information Processing Systems Foundation, MIT Press.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 2(3):408–421.

Wilson, D. R. & Martínez, T. R. (1997). Improved heterogeneous distance functions. *J. Artif. Intell. Res. (JAIR)*, 6:1–34.

Wilson, D. R. & Martínez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Mach. Learn.*, 38(3):257–286.

Zhao, H. T. & Yuen, P. C. (2007). Incremental linear discriminant analysis for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(1):210–221.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458.

Zheng, W., Zhao, L., & Zou, C. (2004). Locally nearest neighbor classifiers for pattern classification. *Pattern Recognition*, 37:1307–1309.

Zhou, Z., Li, S. Z., & Chan, K. L. (2000). A theoretical justification of nearest feature line method. In *Proceedings of the 15th International Conference on Pattern Recognition ICPR2000, Barcelona, Spain*, volume II, pages 2759–2762. IAPR, IEEE Computer Society.