The Dissimilarity Representation for Structural Pattern Recognition

Robert P.W. Duin[1] and Elżbieta Pękalska[2]

[1] Pattern Recognition Laboratory,
Delft University of Technology, The Netherlands
r.duin@ieee.org
[2] School of Computer Science,
University of Manchester, United Kingdom
pekalska@cs.man.ac.uk

Abstract. The patterns in collections of real world objects are often not based on a limited set of isolated properties such as features. Instead, the totality of their appearance constitutes the basis of the human recognition of patterns. Structural pattern recognition aims to find explicit procedures that mimic the learning and classification made by human experts in well-defined and restricted areas of application. This is often done by defining dissimilarity measures between objects and measuring them between training examples and new objects to be recognized.

The dissimilarity representation offers the possibility to apply the tools developed in machine learning and statistical pattern recognition to learn from structural object representations such as graphs and strings. These procedures are also applicable to the recognition of histograms, spectra, images and time sequences taking into account the connectivity of samples (bins, wavelengths, pixels or time samples).

The topic of dissimilarity representation is related to the field of non-Mercer kernels in machine learning but it covers a wider set of classifiers and applications. Recently much progress has been made in this area and many interesting applications have been studied in medical diagnosis, seismic and hyperspectral imaging, chemometrics and computer vision. This review paper offers an introduction to this field and presents a number of real world applications[1].

1 Introduction

In the totality of the world around us we are able to recognize events or objects as separate items distinguished from their surroundings. We recognize the song of a bird in the noise of the wind, an individual tree in the wood, a cup on the table, a face in the crowd or a word in the newspaper. Two steps can now be distinguished. First, the objects are detected in their totality. Second, the isolated

[1] We acknowledge financial support from the FET program within the EU FP7, under the SIMBAD project (contract 213250) as well as the Engineering and Physical Sciences Research Council in the UK.

C. San Martin and S.-W. Kim (Eds.): CIARP 2011, LNCS 7042, pp. 1–24, 2011.

object is recognized. These two steps are strongly interconnected and verified by each other. Only after a satisfactory recognition the detection takes place. It may even be questioned whether it is not artificial to make a distinction of these processes in the human recognition of interesting items in the surrounding world.

It is common to separate the two processes in the design of artificial recognition systems. This is possible and fruitful as it is known what type of objects are considered in most applications. For example, we know that the system under construction has to recognize faces and is not intended to recognize characters or objects such as cups. The detection step is thereby simplified to selectively focus on faces only, on characters only or on cups only. The recognition step, however, may now lack important information from the context: the recognition of an isolated character is more difficult than its recognition given the entire word. Recognition systems that take the context into account can become more accurate, albeit at the price of a higher complexity.

On the level of the recognition of a single object a similar observation can made. In the traditional pattern recognition approaches this is mainly done by describing objects by isolated features. These are object properties that are appropriate locally, at some position on the object, e.g. the sharpness of a corner, or by global properties that describe just a single aspect such as the weight or size of the object. After these features are determined in a first step, the class or the name of the object is determined: it is a cup and not an ashtray, or it is the character 'C' out of the character set in the alphabet. Again it can be doubted whether these steps reflect the human recognition process.

Is it really true that we consciously observe a set of features before we come to a decision? Can we really name well-defined properties that distinguish a cup from an ashtray, or John from Peter? Some experts who have thought this over for their field of expertise may come a long way. Many people, however, can perfectly perform a recognition task, but can hardly name specific features that served the purpose. It is only under pressure when they mention some features.

In general, the process of human decision making may not be based on clear arguments but on an unconscious intuition, instead. Arguments or justifications may be generated afterwards. They may even be disputed and refuted without changing the decision. This points in the direction that human recognition and decision making are global processes. These processes take the entire object or situation into account and a specification into isolated observations and arguments becomes difficult.

The above reasoning raises the question whether it is possible to constitute an automatic pattern recognition procedure that is based on the totality of an object. In this paper some steps in this direction are formulated on the basis of the dissimilarity representation. A review will be given of the research that is done by the authors and their colleagues. Although many papers and experiments will be mentioned that describe their work, it should be emphasized that the context of the research has been of significant importance. The publications and remarks by researchers such as Goldfarb [29], Bunke [44], Hancock and Wilson [40,63], Buhman and Roth [36], Haasdonk [30], Mottle [41], Edelman [25] and

Vapnik [58] have been a significant source of inspiration on this topic. It is however the aim of this paper to sketch our own line of reasoning in such a way that it may serve as inspiration for newcomers in this area. We will therefore restrict this paper to an intuitive explanation illustrated by some examples. More details can be found in the references. Parts of this paper have been extracted from a recent journal paper [20] which deals with the same topic but which is more dedicated to research results and in which less effort has been made to introduce ideas and concepts carefully.

A global description of objects, which takes their totality into account, should be based on knowledge of how all aspects of the object contribute to the way it appears to the observer. To make this knowledge explicit some structural model may be used, e.g. based on graphs. This is not a simple task and usually demands much more background knowledge of the application area than the definition of some local properties such as features. The feature-based approach is mainly an effort in measuring the properties of interest in the observations as presented by the sensors. As features describe objects just partially, objects belonging to different classes may share the same feature vectors. This overlap has to be solved by a statistical analysis. The two approaches, mentioned above, are linked to the two subfields: structural and statistical pattern recognition.

The possibility to merge the two fields has intrigued various researchers over the decades. Thereby, it has been a research topic from the early days of pattern recognition. Originally, most attempts have been made by modifying the structural approach. Watanabe [59] and especially Fu [26] pointed to several possibilities using information theoretic considerations and stochastic syntactical descriptions. In spite of their inspiring research efforts, it hardly resulted in practical applications. Around 1985 Goldfarb [29] proposed to unify the two directions by replacing the feature-based representation of individual objects by distances between structural object models. As this proposal hardly requires a change of the existing structural recognition procedures, it may be considered as an attempt to bridge the gap between the two fields by approaching it from the statistical side. Existing statistical tools might thereby become available in the domain of structural pattern recognition. This idea did not attract much attention as it was hardly recognized as a profitable approach.

After 1995, the authors of this paper started to study this proposal further. They called it the *dissimilarity representation* as it allows various non-metric, indefinite or even asymmetric measures. The first experiences were published in a monograph in 2005, [49]. An inspiration for this approach was also the above explained observation that a human observer is primary triggered by object differences (and later similarities) and that the description by features and models comes second; see [25]. The analysis of dissimilarities, mainly for visualization, was already studied in the domain of psychonomy in the 1960s, e.g. by Shepard [54] and Kruskal [35]. The emphasis of the renewed interest in dissimilarities in pattern recognition, however, was in the construction of vector spaces that are suitable for training classifiers using the extensive toolboxes available in multivariate statistics, machine learning and pattern recognition. The significance for the accessibility of these tools in structural object recognition was recognized by Bunke [55,44] and others such as Hancock and Wilson [64] and Mottle [42,41].

Before introducing the further contents, let us first summarize key advantages and drawbacks of using the dissimilarity representation in statistical learning:

- Powerful statistical pattern recognition approaches become available for structural object descriptions.
- It enables the application expert to use model knowledge in a statistical setting.
- As dissimilarities can be computed on top of a feature-based description, the dissimilarity approach may also be used to design classifiers in a feature space. These classifiers perform very well in comparative studies [23].
- As a result, structural and feature-based information can be combined.
- Insufficient performance can often be improved by more observations without changing the dissimilarity measure.
- The computational complexity during execution, i.e. the time spent on the classification of new objects, is adjustable.
- The original representation can be large and computationally complex as dissimilarities between all object pairs may have to be computed. There are ways to reduce this problem [48,39,13].

In this paper we present an intuitive introduction to dissimilarities (Sec. 2), ways to use them for representation (Sec. 3), the computation of classifiers (Sec. 4), the use of multiple dissimilarities (Sec. 5) and some applications (Sec. 6). The paper is concluded with a discussion of problems under research.

2 Dissimilarities

Suppose we are given an object to be recognized. That means: can we name it, or can we determine a class of objects of which it belongs to? Some representation is needed if we want to feed it to a computer for an automatic recognition. Recognition is based on a comparison with previous observations of objects like the one we have now. So, we have to search through some memory. It would be great if an identical object could be found there. Usually, an exact match is impossible. New objects or their observations are often at least slightly different from the ones previously seen. And this is the challenge of pattern recognition: can we recognize objects that are at most similar to the examples that we have been given before? This implies that we need at least the possibility to express the similarity between objects in a quantitative way. In addition, it is not always advantageous to look for an individual match. The generalization of classes of objects to a 'concept', or a distinction which can be expressed in a simple classification rule is often faster, demands less memory and/or can be more accurate.

It has been observed before [25], and it is in line with the above discussions, that in human recognition processes it is more natural to rely on similarities or dissimilarities between objects than to find explicit features of the objects that are used in the recognition. This points to a representation of objects based on a pairwise comparison of the new examples with examples that are already collected. This differs from the feature-based representations that constitute the basis of the traditional approaches to pattern recognition described in the well-known textbooks by Fukunaga [27], Duda, Hart and Stork [17], Devijver and Kittler [16], Ripley [53], Bishop [7],Webb [60] and Theodorides [56]. We want to point out that although the pairwise dissimilarity representation presented here is different in its foundation from the feature-based representation, many procedures described in the textbooks can be applied in a fruitful way.

We will now assume that a human recognizer, preferably an expert w.r.t. the objects of interest, is able to formulate a dissimilarity measure between objects that reflects his own perception of object differences (for now we will stick to dissimilarity measures). A dissimilarity measure $d(o_i, o_j)$ between two objects o_i and o_j out of a training set of n objects may have one or more of the following properties for all $i, j, k \leq n$.

- **Non-negativity:** $d(o_i, o_j) \geq 0$.
- **Identity of indiscernibles:** $d(o_i, o_j) = 0$ if and only if $o_i \equiv o_j$.
- **Symmetry:** $d(o_i, o_j) = d(o_j, o_i)$.
- **Triangle inequality:** $d(o_i, o_j) + d(o_j, o_k) \geq d(o_i, o_k)$.
- **Euclidean:** An $n \times n$ dissimilarity matrix D is Euclidean if there exists an isometric Euclidean embedding into a Euclidean space. In other words, a Euclidean space with n vectors can be found such that the pairwise Euclidean distances between these vectors are equal to the original distances in D.
- **Compactness:** A dissimilarity measure is defined here as compact if a sufficiently small perturbation of an object (from a set of allowed transformations) leads to an arbitrary small dissimilarity value between the disturbed object and its original. We call such a measure compact because it results in compact class descriptions for which sufficiently small disturbances will not change the class membership of objects. Note that this definition is different than compactness discussed in topological spaces.

The first two properties together produce positive definite dissimilarity measures. The first four properties coincide with the mathematical definition of a metric distance measure.

Non-negativity and symmetry seem to be obvious properties, but sometimes dissimilarity measures are defined otherwise. E.g. if we define the distance to a city as the distance to the border of that city, then a car that reaches the border from outside has a distance zero. When the car drives further into the city the distance may be counted as negative in order to keep consistency. An example of an asymmetric distance measure is the directed Hausdorff distance between two sets of points A and B: $d_H(A, B) = max_x\{min_y\{d(x, y), x \in A, y \in B\}\}$.

An important consequence of using positive definite dissimilarity measures is that classes are separable for unambiguously labeled objects (identical objects belong to the same class). This directly follows from the fact that if two objects have a distance zero they should be identical and as a consequence they belong to the same class. For such classes a zero-error classifier exists (but may still be difficult to find). See [49]. This is only true if the dissimilarity measure reflects all object differences. Dissimilarity measures based on graphs, histograms, features or other derived measurements may not be positive definite as different objects may still be described by the same graph (or histogram, or sequence) and thereby have a zero dissimilarity.

The main property is the Euclidean property. A metric distance measure in fact states that the Euclidean property holds for every set of three points while the first two properties (positive definiteness) state that the dissimilarity of every pair of points is Euclidean.

We may distinguish the properties of the dissimilarity measure itself and the way it works out for a set of objects. The first should be analyzed mathematically from the definition of the measure and the known properties of the objects. The second can be checked numerically from a given $n \times n$ dissimilarity matrix D. There might be a discrepancy between what is observed in a finite data matrix and the definition of the measure. It may occur for instance that the matrix D for a given training set of objects is perfectly Euclidean but that the dissimilarities for new objects behave differently.

The concept of compactness is important for pattern recognition. It was first used in the Russian literature around 1965, e.g. see Arkedev and Braverman [3], and also [18]. We define here that a compact class of objects consists of a finite number of subsets, such that in each subset every object can be continuously transformed (morphed) into every other object of that subset without passing through objects outside the subset. This property of compactness is slightly different from the original concept defined in [3] where it is used as a hypothesis on which classifiers are defined. It is related to the compactness used in topology. Compactness is a basis for generalization from examples. Without proof we state here that for compact classes the consequence of the no-free-lunch theorem [65] (every classifier is as good as any other classifier unless we use additional knowledge) is avoided: compactness pays the lunch. The prospect is that for the case of positive definite dissimilarity measures and unambiguously labeled objects, the classes can be separated perfectly by classifiers of a finite complexity.

3 Representation

A representation of real world objects is needed in order to be able to relate to them. It prepares the generalization step by which new, unseen objects are classified. So, the better the representation, the more accurate classifiers can be trained. The traditional representation is defined by numerical features. The use of dissimilarities is an attractive alternative, for which arguments were given in Introduction. This section provides more details by focussing on the object structure.

3.1 Structural Representations

The concept of structure is ill defined. It is related to the global connectivity of all parts of the object. An image can be described by a set of pixels organized in a square grid. This grid may be considered as the structure of the image. It is, however, independent of the content of the image. If this is taken into account then the connectivity between the pixels may be captured by weights, e.g. related to the intensity values of the neighboring pixels. We may also forget that there are pixels and determine regions in the image by a segmentation procedure. The structure may then be represented by a graph in which every node is related to an image segment and the graph edges correspond to neighboring segments. Nodes and edges may have attributes that describe properties of the segments and the borders between them.

A simpler example of a structure is the contour of an image segment or a blob: its shape. The concept of shape leads to a structure, but shapes are also characterized by features, e.g. the number of extremes or a set of moments. A structural representation of a shape is a string. This is a sequence of symbols representing small shape elements, such as straight lines (in some direction) or curves with predefined curvatures. Shapes are also found in spectra, histograms and time signals. The movement of an object or a human body may be described as a set of coordinates in a high-dimensional space as a function of time. This multi-dimensional trajectory has a shape and may be considered as a structure.

The above examples indicate that structures also have some (local) properties that are needed for their characterization. Examples of pure structures without attributes can hardly be found. Certainly, if we want to represent them in a way that facilitates comparisons, we will use attributes and relations (connections). The structural representations used here will be restricted to attributed graphs and sequences.

3.2 The Dissimilarity Representation

Dissimilarities themselves have been discussed in Sec. 2. Three sets of objects may be distinguished for constructing a representation:

- **A representation set** $R = \{r_1, \ldots, r_k\}$. These are the objects we refer to. The dissimilarities to the representation set have to be computed for training objects as well as for test objects used for evaluation, or any objects to be classified later. Sometimes the objects in the set R are called prototypes. This word may suggest that these objects are in some way typical examples of the classes. That can be the case but it is not necessary. So prototypes may be used for representation, but the representation set may also consist of other objects.
- **A training set** $T = \{o_1, \ldots, o_n\}$. These are the objects that are used to train classifiers. In many applications we use $T := R$, but R may also be just a (small) subset of T, or be entirely different from T.
- **A test set** S. These are the objects that are used to evaluate classification procedure. They should be representative for the target objects for which the classification procedure is built.

After determining these three sets of objects the dissimilarity matrices $D(T, R)$ and $D(S, R)$ have to be computed. Sometimes also $D(T, T)$ is needed, e.g. when the representation set $R \subset T$ has to be determined by a specific algorithm. The next problem is how to use these two or three matrices for training and testing. Three procedures are usually considered:

- **The k-nearest neighbor classifier.** This is the traditional way to classify new objects in the field of structural pattern recognition: assign new objects to the (majority) class of its (k) nearest neighbor(s). This procedure can directly by applied to $D(S,T)$. The dissimilarities inside the training set, $D(T,T)$ or $D(T,R)$ are not used.
- **Embedded space.** Here a vector space and a metric (usually Euclidean) are determined from $D(T,T)$ containing $n = |T|$ vectors, such that the distances between these vectors are equal to the given dissimilarities. See Sec. 3.3 for more details.
- **The dissimilarity space.** This space is postulated as a Euclidean vector space defined by the dissimilarity vectors $d(\cdot,R) = [d(\cdot,r_1),\ldots,d(\cdot,r_k)]^T$ computed to the representation set R as dimensions. Effectively, the dissimilarity vectors are used as numerical features. See Sec. 3.4.

3.3 Embedding of Dissimilarities

The topic of embedding dissimilarity matrices has been studied for a long time. As mentioned in the introduction (Sec. 1), it was originally used for visualizing the results of psychonomic experiments and other experiments representing data in pairwise object comparisons [54,35]. In such visualization tasks, a reliable, usually 2D map of the data structure is of primary importance. Various nonlinear procedures have been developed over the years under the name of multi-dimensional scaling (MDS) [9].

It is difficult to reliably project new data to an existing embedded space resulting from a nonlinear embedding. Therefore, such embeddings are unsuitable for pattern recognition purposes in which a classifier trained in the embedded space needs to be applied to new objects. A second, more important drawback of the use of nonlinear MDS for embedding is that the resulting space does not reflect the original distances anymore. It usually focusses either on local or global object relations to force a 2D (or other low-dimensional) result.

For the purpose of generalization a restriction to low-dimensional spaces is not needed. Moreover, for the purpose of the projection of new objects linear procedures are preferred. Therefore, the linear MDS embedding has been studied, also known as classical scaling [9]. As the resulting Euclidean space is by its very nature not able to perfectly represent non-Euclidean dissimilarity data, see Sec. 2, a compromise has to be made. The linear Euclidean embedding procedure is based on an eigenvalue decomposition of the Gram matrix derived from the given $n \times n$ dissimilarity matrix D, see [29,49], in which some eigenvalues become negative for non-Euclidean dissimilarities. This conflicts with the construction of a Euclidean space as these eigenvalues are related to variances of the extracted features, which should be positive. This is solved in classical scaling by neglecting all 'negative' directions. The distances in this embedded space may thereby be entirely different from the original dissimilarity matrix D.

The approach followed by the pseudo-Euclidean embedding is to construct a vector space [49] in which the metric is adjusted such that the squared distance contributions of the 'negative' eigenvectors are counted as negative. The resulting pseudo-Euclidean space thereby consists out of two orthogonal Euclidean spaces of which the distances are not added (in the squared sense) but subtracted. Distances computed in this way are exactly equal to the original dissimilarities, provided that they are symmetric and self-dissimilarity is zero. Such an embedding is therefor an isometric mapping of the original D into a suitable pseudo-Euclidean space, which is an inner product space with an indefinite inner product.

The perfect representation of D in a pseudo-Euclidean embedded space is an interesting proposal, but it is not free from some disadvantages:

- Embedding relies on a square dissimilarity matrix, usually $D(T,T)$. The dissimilarities between all pairs of training objects should be taken into account. The computation of this matrix as well as the embedding procedure itself may thereby be time and memory demanding operations.
- Classifiers that obey the specific metric of the Pseudo-Euclidean space are difficult to construct or not yet well defined. Some have been studied [32,50,21], but many problems remain. For instance, it is not clear how to define a normal distribution in a pseudo-Euclidean space. Also the computation of SVM may be in trouble as the related kernel is indefinite, in general [31]. Solutions are available for specific cases. See also Sec. 4.
- There is a difficulty in a meaningful projection of new objects to an existing pseudo-Euclidean embedded space. The straightforward projection operations are simple and linear, but they may yield solutions with negative distances to other objects even though the original distances are non-negative. This usually happens when a test object is either an outlier or not well represented in the training set T (which served to define the embedded space). A possible solution is to include such objects in the embedding procedure and retrain the classifier for the new objects. For test objects this implies that they will participate in the representation. Classification may thereby improve at the cost of the retraining. This approach is also known as transductive learning [58].
- The fact that embedding strictly obeys the given dissimilarities is not always an advantage. All types of noise and approximations related to the computation of dissimilarities are expressed in the result. It may thereby be questioned whether all non-Euclidean aspects of the data are informative. In [19] it is shown that there are problems for which this is really the case.

In order to define a proper topology and metric, mathematical texts, ȩ.g. [8], propose to work with the associated Euclidean space instead of the pseudo-Euclidean space. In this approach all 'negative' directions are treated as 'positive' ones. As a result, one can freely use all traditional classifiers in such a space. The information extracted from the dissimilarity matrix is used but the original distance information is not preserved and may even be significantly distorted. Whether this is beneficial for statistical learning depends on the problem.

3.4 The Dissimilarity Space

The dissimilarity space [46,49] postulates a Euclidean vector space defined by the dissimilarity vectors. The elements of these vectors are dissimilarities from a given object to the objects in the representation set R. The dissimilarity vectors serve as features for the objects in the training set. Consequently, such a space overcomes all problems that usually arise with the non-Euclidean dissimilarity measures, simply by neglecting the character of the dissimilarity. This approach is at least locally consistent for metric distance measures: distances in the dissimilarity space between pairs of objects characterized by small dissimilarities $d(o_i, o_j)$ will also have a small distance as their dissimilarity vectors $d(o_i, R) = [d(o_i, r_1), \ldots, d(o_i, r_k)]^T$ and $d(o_j, R) = [d(o_j, r_1), \ldots, d(o_j, r_k)]^T$ will be about equal. This may serve as a proof that the topology of a set of objects with given dissimilarities $\{d(o_i, o_j)\}_{i,j=1:n}$ is identical to the topology of this set of objects in the dissimilarity space $\{d_E(d(o_i, R), d(o_j, R))\}_{i,j=1:n}$provided that R is sufficiently large (to avoid that different objects have, by accident, a zero distance in the dissimilarity space).

If all training objects are used for representation, the dimension of the dissimilarity space is equal to $|T|$. Although, in principle, any classifier defined for a feature space may be applied to the dissimilarity space, some of them will be ill-defined or overtrained for such a large representation set. Dimension reduction, e.g. by prototype selection may thereby be an important issue in this approach [48,39,13]. Fortunately, these studies show that if the reduction is not put to the extreme, a randomly selected representation set may do well. Here the dissimilarity space is essentially different from a traditional feature space: features may be entirely different in their nature. A random selection of R may exclude a few significant examples. The objects in a training set, however, will in expectation include many similar ones. So, a random selection is expected to sample all possible aspects of the training set, provided that the training set T as well as the selected R are sufficiently large.

If a representation set R is a subset of T and we use the complete set T in training, the resulting representation $D(T, R)$ contains some zero dissimilarities to objects in R. This is not expected to be the case for new test objects. In that sense the training objects that participate in the representation set are not representative for test objects. It might be better to exclude them. In all our experiments however we found just minor differences in the results if we used $D(T \backslash R, R)$ instead of $D(T, R)$.

Although any feature-based classifier can be used in a dissimilarity space, some fit more naturally than others. For that reason we report a number of experiments and their findings in Sec. 4.

4 Classifiers

We will discuss here a few well-known classifiers and their behavior in various spaces. This is a summary of our experiences in many studies and applications. See [49] and its references.

In making a choice between embedding and the dissimilarity space for training a classifier one should take into account the essential differences between these spaces. As already stated, embedding strictly obeys the distance characteristics of the given dissimilarities, while the dissimilarity spaces neglects this. In addition, there is a nonlinear transformation between these spaces: by computing the distances to the representation objects in the embedded space the dissimilarity space can be defined. As a consequence, a linear classifier in the embedded space is a nonlinear classifier in the dissimilarity space, and the other way around. Comparing linear classifiers computed in these spaces is thereby comparing linear and nonlinear classifiers.

It is outside the scope of this paper, but the following observation might be helpful for some readers. If the dissimilarities are not constructed by a procedure on a structural representation of objects, but are derived as Euclidean distances in a feature space, then the pseudo-Euclidean embedding effectively reconstructs the original Euclidean feature space (except for orthonormal transformations). So in that case a linear classifier in the dissimilarity space is a nonlinear classifier in the embedded space, which is the same nonlinear classifier in the feature space. Such a classifier, computed in a dissimilarity space, can perform very well [23].

4.1 Nearest Neighbor Classifier

The k-nearest neighbor (k-NN) classifier in an embedded (pseudo-)Euclidean space is based on the distances computed in this space. By definition these are the original dissimilarities (provided that the test examples are embedded together with the training objects). So without the process of embedding this classifier, can directly be applied to a given dissimilarity matrix. This is the classifier traditionally used by many researchers in the area of structural pattern recognition. The study of the dissimilarity representation arose because this classifier does not make use of the given dissimilarities in the training set. Classification is entirely based on the dissimilarities of a test object to the objects in the training (or representation) set only.

The k-NN rule computed in the dissimilarity space relies on a Euclidean distance between the dissimilarity vectors, hence the nearest neighbors are determined by using all dissimilarities of a given object to the representation objects. As explained in Sec. 3.4 for the metric case and for large sets it is expected that the distances between similar objects are small for the two spaces. So, it is expected that learning curves are asymptotically identical, but for small training sets the dissimilarity space works better as it uses more information.

4.2 Parzen Density Classifiers

The class densities computed by the Parzen kernel density procedure are based on pairwise distance computations between objects. The applicability of this classifier as well as its performance is thereby related to those of the k-NN rule. Differences are that this classifier is more smooth, depending on the choice of the smoothing parameter (kernel) and that its optimization involves the entire training set.

4.3 Normal Density Bayes Classifiers

Bayes classifiers assume that classes can be described by probability density functions. Using class priors and Bayes' rule the expected classification error is minimized. In case of normal density function either a linear classifier (Linear Discriminant Analysis, LDA) arises on the basis of equal class covariances, or a quadratic classifier is obtained for the general case (Quadratic Discriminant Analysis, QDA). These two classifiers belong to best possible in case of (close to) normal class distributions and a sufficiently large training set. As mean vectors and covariance matrices can be computed in a pseudo-Euclidean space, see [29,49], these classifiers exist there as well if we forget the starting point of normal distributions. The reason is that normal distributions are not well defined in pseudo-Euclidean spaces; it is not clear what a normal distribution is unless we refer to associated Euclidean spaces.

In a dissimilarity space the assumption of normal distributions works often very well. This is due to the fact that many cases dissimilarity measures are based on, or related to sums of numerical differences. Under certain conditions large sums of random variables tend to be normally distributed. It is not perfectly true for distances as we often get Weibull [12] or χ^2 distributions, but the approximations are sufficient for a good performance of LDA and QDA. The effect is emphasized if the classification procedure involves the computation of linear subspaces, e.g. by PCA. Thanks to projections normality is emphasized even more.

4.4 Fisher's Linear Discriminant

In a Euclidean space the Fisher linear discriminant (FLD) is defined as the linear classifier that maximizes the Fisher criterion, i.e. the ratio of the between-class variance to the within-class variance. For a two-class problem, the solution is equivalent to LDA (up to an added constant), even though no assumption is made about normal distributions. Since variance and covariance matrices are well defined in pseudo-Euclidean spaces, the Fisher criterion can be used to derive the FLD classifier there. Interestingly, FLD in a pseudo-Euclidean space coincides with FLD in the associated Euclidean space. FLD is a linear classifier in a pseudo-Euclidean space, but can be rewritten to FLD in the associated space; see also [50,32].

In a dissimilarity space, which is Euclidean by definition, FLD coincides with LDA for a two-class problem. The performance of these classifiers may differ for multi-class problems as the implementations of FLD and LDA will usually vary then. Nevertheless, FLD performs very well. Due to the nonlinearity of the dissimilarity measure, FLD in a dissimilarity space corresponds to a nonlinear classifier in the embedded pseudo-Euclidean space.

4.5 Logistic Classifier

The logistic classifier is based on a model of the class posterior probabilities as a function of the distance to the classifier [1]. The distance between a vector and a linear hyperplane in a pseudo-Euclidean space however is an unsuitable concept for classification as it can have any value $(-\infty, \infty)$ for vectors on the same side of this hyperplane. We are not aware of a definition and an implementation of the logistic classifier for pseudo-Euclidean spaces. Alternatively, the logistic classifier can be constructed in the associated Euclidean space.

In a dissimilarity space, the logistic classifier performs well, although normal density based classifiers work often better. It relaxes the demands for normality as made by LDA. It is also more robust in case of high-dimensional spaces.

4.6 Support Vector Machine (SVM)

The linear kernel in a pseudo-Euclidean space is indefinite (non-Mercer). The quadratic optimization procedure used to optimize a linear SVM may thereby fail [30]. SVM can however be constructed if the contribution of the positive subspace of the Euclidean space is much stronger than that of the negative subspace. Mathematically, it means that the measure is slightly deviating from the Euclidean behavior and the solution of SVM optimization is found in the positive definite neighborhood. Various researchers have reported good results in applying this classifier, e.g. see [11]. Although the solution is not guaranteed and the algorithm (in this case LIBSVM, [14]) does not stop in a global optimum, a good classifier can be obtained.

In case of a dissimilarity space the (linear) SVM is particularly useful for computing classifiers in the complete space for which $R := T$. The given training set defines therefore a separable problem. The SVM does not or just hardly overtrain in this case. The advantage of this procedure is that it does not demand a reduction of the representation set. A linear SVM is well defined. By normalizing the dissimilarity matrix (such that the average dissimilarity is one) we found stable and good results in many applications by setting the trade-off parameter C in the SVM procedure [15] to $C = 100$. Hereby, additional cross-validation loops are avoided to optimize this parameter. As a result, in an application one can focus on optimizing the dissimilarity measure.

4.7 Combining Classifiers

In the area of dissimilarity representations many approaches can be considered. Various strategies can be applied for the choice of the representation set, either embedded or dissimilarity spaces can be used, and various modifications can be considered, e.g. refinements or correction procedures for these spaces; see [24,21]. Instead of selecting one of the approaches, classifier combining may provide an additional value. However, as all these classifiers are based on the same dissimilarities they do not provide any additional or valuable information. Effectively, just additional procedures are considered that encode different nonlinearities. As the given square dissimilarity matrix D describes an already linearly separable set of objects (under the assumption of the positive definite dissimilarity) we do not expect that in general much can be gained by combining, although an occasional success is possible in particular problems.

5 Multiple Dissimilarities

Instead of generating sets of classifiers defined on the same dissimilarity representation also modifications of the dissimilarity measure may be considered. Another measure can emphasize other aspects of the objects. The resulting dissimilarity matrices cannot be derived from each other, in general. Consequently, they are chosen to encode different information. Combining various dissimilarity representations or classifiers derived from them is now much more of interest. These types of studies are closely related to the studies on kernel metric learning [61,68,66]. An important difference is that the study of kernels is often focussed on the use of SVM for classification, and consequently positive definite kernels obeying the Mercer conditions are the key. As the dissimilarity representation permits many classifiers this point is not relevant for dissimilarity measures. On the contrary, the unrestricted use of dissimilarity definitions is of particular significance for structural pattern recognition as there non-Euclidean measures naturally arise. See also [22].

There are a number of reasons why a set of different dissimilarities between objects arises. A few examples are:

- The same set of objects is observed multiple times under different conditions.
- The dissimilarities are computed on different samplings from the original signals (multi-scale approach).
- Different dissimilarity measures are used on the same signals.

A very interesting observation that can be made from various studies such as [33,57] is that a simple element-wise averaging of dissimilarity matrices defined by different measures often leads to a significant improvement of the classification error over the best individual measure. Attempts to improve this further by a weighted averaging is sometimes successful but often appears not to be useful. The precise value of the weights does not seem to be very significant, either.

6 Application Examples

In this section we will discuss a few examples that are typical for the possibilities of the use of dissimilarities in structural pattern recognition problems. Some have been published by us before [22] for another readership. They are repeated here as they may serve well as an illustration in this paper.

6.1 Shapes

A simple and clear example of a structural pattern recognition problem is the recognition of blobs: 2D binary structures. An example is given in Fig. 1. It is an object out of the five-class chickenpieces dataset consisting of 445 images [2]. One of the best structural recognition procedure uses a string representation of the contour described by a set of segments of the same length [10]. The string elements are the consecutive angles of these segments. The weighted edit

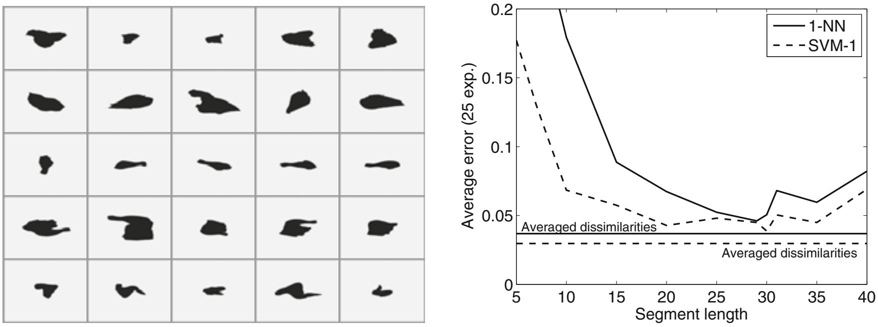

Fig. 1. Left: some examples of the chickenpieces dataset. Right: the error curves as a function of the segment length L.

distances between all pairs of contours are used to compute dissimilarities. This measure is non-Euclidean. A $(\gamma, \text{Ł})$ family of problems is considered depending on the specific choice for the cost of one editing operation γ as well as for the segment's length L used in the contour description. As a result, the classification performance depends on the parameters used, as shown in Fig 1, right. 10-fold cross-validation errors are shown there for the 1-NN rule directly applied on the dissimilarities as well as the results for the linear SVM computed by LIBSVM, [14], in the dissimilarity space. In addition the results are shown for the average of the 11 dissimilarity matrices. It is clearly observed that the linear classifier in the dissimilarity space (SVM-1) improves the traditional 1-NN results and that combining the dissimilarities improves the results further on.

6.2 Histograms and Spectra

Histograms and spectra offer very simple examples of data representations that are judged by human experts on their shape. In addition, also the sampling of the bins or wavelengths may serve as a useful vector representation for an automatic analysis. This is thanks to the fact that the domain is bounded and that spectra are often aligned. Below we give an example in which the dissimilarity representation outperforms the straightforward vector representation based on sampling because the first can correct for a wrong calibration (resulting in an imperfect alignment) in a pairwise fashion. Another reason to prefer dissimilarities for histograms and spectra over sampled vectorial data is that a dissimilarity measure encodes shape information. For examples see the papers by Porro [52,51].

We will consider now a dataset of 612 FL3-A DNA flow cytometer histograms from breast cancer tissues in a resolution of 256 bins. The initial data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000-2004, using the four tubes 3-6 of a DACO Galaxy flow cytometer. Histograms are labeled into three classes: aneuploid (335 patients), diploid (131) and tetraploid (146). We averaged the histograms of the four tubes thereby covering the DNA contents of about 80000 cells per patient.

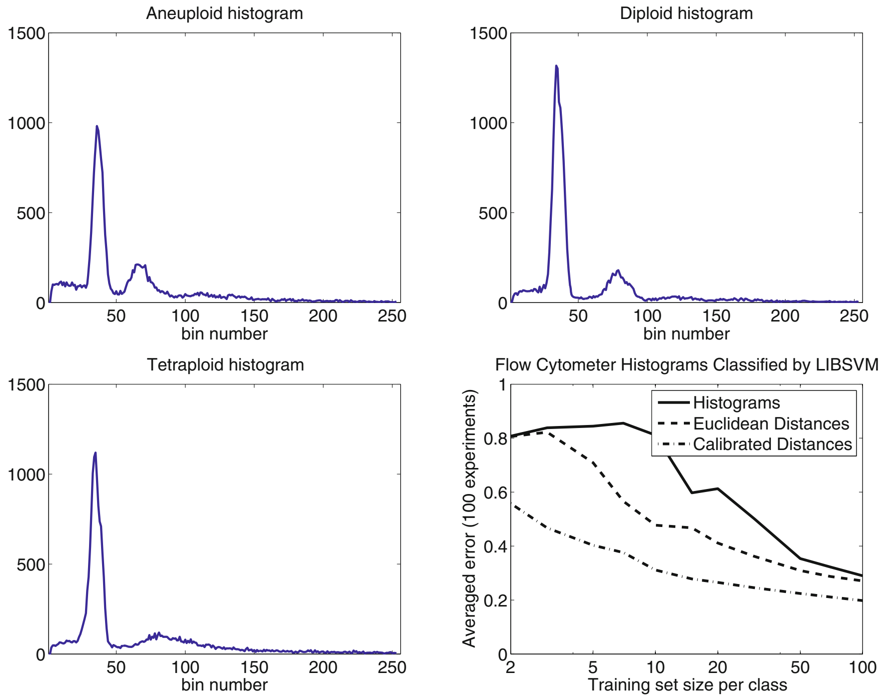

Fig. 2. Examples of some flow cytometer histograms: aneuploid, diploid and tetraploid. Bottom right shows the learning curves.

We removed the first and the last bin of every histogram as here outliers are collected, thereby obtaining 254 bins per histogram. Examples of histograms are shown in Fig. 2. The following representations are used:

Histograms. Objects (patients) are represented by the normalized values of the histograms (summed to one) described by a 254-dimensional vector. This representation is similar to the pixel representation used for images as it is based on just a sampling of the measurements.

Euclidean distances. These dissimilarities are computed as the Euclidean distances in the vector space mentioned above. Every object is represented by by a vector of distances to the objects in the training set.

Calibrated distances. As the histograms may suffer from an incorrect calibration in the horizontal direction (DNA content) for every pairwise dissimilarity we compute the multiplicative correction factor for the bin positions that minimizes their dissimilarity. Here we used the ℓ_1 distance. This representation makes use of the shape structure of the histograms and removes an invariant (the wrong original calibration).

A linear SVM with a fixed trade-off parameter C is used in learning. The learning curves for the three representations are shown in the bottom right of Fig. 2. They clearly illustrate how for this classifier the dissimilarity representation leads to

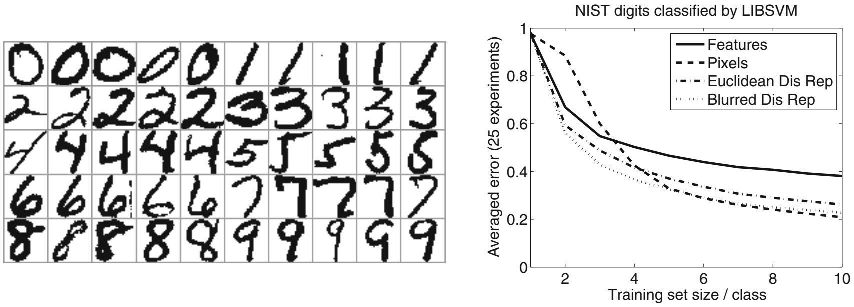

Fig. 3. Left: examples of the images used for the digit recognition experiment. Right: the learning curves.

better results than the vector representation based on the histogram sampling. The use of the background knowledge in the definition of the dissimilarity measure improves the results further.

6.3 Images

The recognition of objects on the basis of the entire image can only be done if these images are aligned. Otherwise, earlier pre-procession or segmentation is necessary. This problem is thereby a 2-dimensional extension of the histogram and spectra recognition task. We will show an example of digit recognition by using a part of the classic NIST database of handwritten numbers [62] on the basis of random subsets of 500 digits for the ten classes 0-9. The images were resampled to 32×32 pixels in such a way that the digits fit either horizontally or vertically. Fig. 3 shows a few examples: black is '1' and white is '0'. The dataset is repeatedly split into training and test sets and hold-out classification is applied. In every split the ten classes are evenly represented.
The following representations are used:

Features. We used 10 moments: the seven rotations invariant moments and the moments $[00], [01], [10]$, measuring the total number of black pixels and the centers of gravity in the horizontal and vertical directions.

Pixels. Every digit is represented by a vector of the intensity values in $32 * 32 = 1024$ dimensional vector space.

Dissimilarities to the training object. Every object is represented by the Euclidean distances to all objects in the training set.

Dissimilarities to blurred digits in the training set. As the pixels in the digit images are spatially connected blurring may emphasize this. In this way the distances between slightly rotated, shifted or locally transformed but otherwise identical digits becomes small.

The results are shown in Fig. 3 on the right. They show that the pixel representation is superior for large training sets. This is to be expected as this representation stores asymptotically the universe of possible digits. For small training sets a suitable set of features may perform better. The moments we use here are very general features. Better ones can be found for digit description. As explained before a feature-based description reduces the (information on the) object: it may be insensitive for some object modifications. For sufficiently large representation sets the dissimilarity representation may see all object differences and may thereby perform better.

6.4 Sequences

The recognition of sequences of observations is in particular difficult if the sequences of a given class vary in length, but capture the same 'story' (information) from the beginning to the end. Some may run faster, or even run faster over just a part of the story and slow down elsewhere. A possible solution is to rely on Dynamic Time Warping (DTW) that relates the sequences in a nonlinear way, yet obeys the order of the events. Once two sequences are optimally aligned, the distance between them may be computed.

An example in which the above has been applied successfully is the recognition of 3-dimensional gestures from the sign language [38] based on an statistically optimized DTW procedure [4]. We took a part of a dataset of this study: the 20 classes (signs) that were most frequently available. Each of these classes has 75 examples. The entire dataset thereby consists of a 1500×1500 matrix of DTW-based dissimilarities. The leave-one-out 1-NN error for this dataset is 0.041, which is based on the computation of 1499 DTW dissimilarities per test object. In Fig. 4, left, a scatterplot is shown of the first two PCA components showing that some classes can already be distinguished with these two features (linear combinations of dissimilarities).

We studied dissimilarity representations consisting of just one randomly drawn example per class. The resulting dissimilarity space has thereby 20 dimensions. New objects have to be compared with just these 20 objects. This space is now filled with randomly selected training sets of containing between 2 and 50 objects per class. Remaining objects are used for testing. Two classifiers are studied, the linear SVM (using the LIBSVM package [14]) with a fixed trade-off parameter $C = 100$ (we used normalized dissimilarity matrices with average dissimilarities of 100) and LDA. The experiment was repeated 25 times and the results averaged out. The learning curves in Fig. 4, right, show the constant value of the 1-NN classifier performance using the dissimilarities to the single training examples per class only, and the increasing performances of the two classifiers for a growing number of training objects. Their average errors for 50 training objects per class is 0.07. Recall that this is still based on the computation of just 20 DTW dissimilarities per object as we work in the related 20-dimensional dissimilarity space. Our experiments show that LDA reaches an error of 0.035 for a representation set of three objects per class, i.e. 60 objects in total. Again, the training set size is 50 examples per class, i.e. 1000 examples in total. For testing new objects one needs to compute a weighted sum (linear combination) of 60 dissimilarity values giving the error of 0.035 instead of computing and ordering 1500 dissimilarities to all training objects for the 1-NN classifier leading to an error of 0.041.

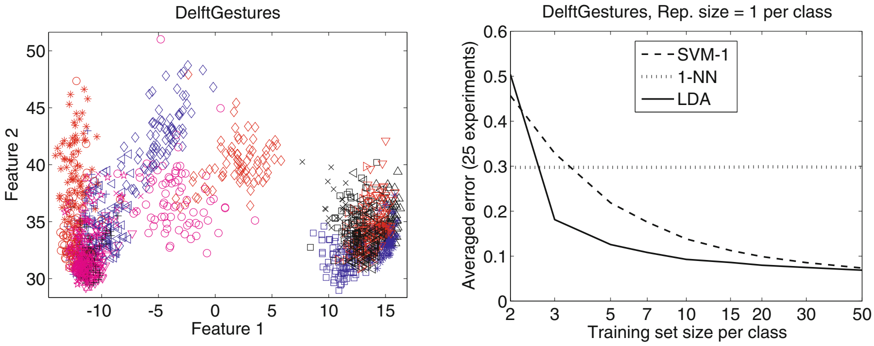

Fig. 4. PCA and learning curves for the 20-class Delft Gesture Dataset

6.5 Graphs

Graphs[2] are the main representation for describing structure in observed objects. In order to classify new objects, the pairwise differences between graphs have to be computed by using a graph matching technique. The resulting dissimilarities are usually related to the cost of matching and may be used to define a dissimilarity representation. We present here classification results obtained with a simple set of graphs describing four objects in the Coil database [43] described by 72 images for every object. The graphs are the Delaunay triangulations derived from corner points found in these images; see [67]. They are unattributed. Hence, the graphs describe the structure only. We used three dissimilarity measures:

CoilDelftSame. Dissimilarities are found in a 5D space of eigenvectors derived from the two graphs by the JoEig approach; see [37]

CoilDelftDiff. Graphs are compared in the eigenspace with a dimensionality determined by the smallest graph in every pairwise comparison by the JoEig approach; see [37]

CoilYork. Dissimilarities are found by graph matching, using the algorithm of Gold and Ranguranjan; [28]

All dissimilarity matrices are normalized such that the average dissimilarity is 1. In addition to the three dissimilarity datasets we used also their averaged dissimilarity matrix.

In a 10-fold cross-validation experiment, with $R := T$, we use four classifiers: the 1-NN rule on the given dissimilarities and the 1-NN rule in the dissimilarity space (listed as 1-NND in Table 6.5), LDA on a PCA-derived subspace covering 99% of the variance and the linear SVM with a fixed trade-off parameter $C = 1$. All experiments are repeated 25 times. Table 6.5 reports the mean classification errors and the standard deviations of these means in between brackets. Some interesting observations are:

[2] Results presented in this section are based on joint research with Prof. Richard Wilson, University of York, UK, and Dr. Wan-Jui Lee, Delft University of Technology, The Netherlands.

Table 1. 10-fold cross-validation errors averaged over 25 repetitions

dataset	1-NN	1-NND	PCA-LDA	SVM-1
CoilDelftDiff	0.477 (0.002)	0.441 (0.003)	0.403 (0.003)	0.395 (0.003)
CoilDelftSame	0.646 (0.002)	0.406 (0.003)	0.423 (0.003)	0.387 (0.003)
CoilYork	0.252 (0.003)	0.368 (0.004)	0.310 (0.004)	0.326 (0.003)
Averaged	0.373 (0.002)	0.217 (0.003)	0.264 (0.003)	0.238 (0.002)

- The CoilYork dissimilarity measure is apparently much better than the two CoilDelft measures.
- The classifiers in the dissimilarity space however are not useful for the CoilYork measure, but they are for the CoilDelft measures. Apparently these two ways of computing dissimilarities are essentially different.
- Averaging all three measures significantly improves the classifier performance in the resulting dissimilarity space, even outperforming the original best CoilYork result. It is striking that this does not hold for the 1-NN rule applied to the original dissimilarities.

7 Discussion

In this paper we have given a review of the arguments why the dissimilarity representation is useful for applications in structural pattern recognition. This has been illustrated by a set of examples on real world data. This all shows that using the collective information from all other objects and relating them to each other on the top of the given pairwise dissimilarities (either in the dissimilarity or embedded space), reveals an additional source of information that is otherwise unexplored.

The dissimilarity representation makes the statistical pattern recognition tools available for structural data. In addition, features are given the use of combiners may be considered or the features may be included in the dissimilarity measure. If either the chosen or optimized dissimilarity measure covers all relevant aspects of the data, then a zero dissimilarity arises if and only if the objects are identical. In that case the classes are separable in a sufficiently large dissimilarity space. Traditional statistical classification tools are designed for overlapping classes. They may still be applied, but the topic of designing proper generalization tools may be reconsidered for the case of high-dimensional separable classes. For instance, the demand that a training set should be representative for the future data to be classified in the statistical sense (i.e. they are generated from the same distributions) is not necessary anymore. These sets should just cover the same domain.

A result, not emphasized in this paper, is that for positive definite dissimilarity measures, see Sec. 2, and sufficiently complex classifiers, any measure asymptotically (for increasing training and representation sets) reaches a zero-error classifier. So, a poorly discriminative dissimilarity measure can be compensated by a large training set as long as the measure is positive definite.

An interesting experimental observation is that if several of these measures are given the average of the dissimilarity matrix offers a better representation than any of them separable. Apparently, the asymptotic convergence speeds (almost) always contribute in combinations and do not disturb each other.

One may wonder whether the dissimilarity measures used in Sec. 6 are all have the positive definite property. However, entirely different objects may be described by identical histograms or graphs. So, the users should analyze, if they need this property and whether an expert is able to label the objects unambiguously on the basis of histograms or graphs only. If not, as a way to attain a better generalization, he may try to extend the distance measure with some features, or simply add another, possibly bad measure, which is positive definite.

They area of dissimilarity representations is conceptually closely related to kernel design and kernel classifiers. It is, however, more general as it allows for indefinite measures and makes no restrictions w.r.t. the classifier [47,50]. The dissimilarity representation is essentially different from kernel design in the sense that the dissimilarity matrix is not necessarily square. This has not only strong computational advantages, but also paves the way to the use of various classifiers. As pointed out in Sec. 3.4, systematic prototype selection is mainly relevant to obtain low-dimensional dissimilarity spaces defined by a small set of prototypes. Another way to reach this goal, not discussed here due to space limit, is the use of out-of-the training set prototypes or the so-called generalized dissimilarity representation. Here prototypes are replaced by sets of prototypes, by models based on such sets, or by artificially constructed prototypes; see [5,6,45,34].

For future research in this field we recommend the study of dissimilarity measures for sets of applications such as spectra, images, etcetera. In every individual application measures may be optimized for the specific usage, but the availability of sets of measures for a broader field of structural applications may, according to our intuition, be most profitable for the field of structural pattern recognition.

References

1. Anderson, J.A.: Logistic discrimination. In: Krishnaiah, P.R., Kanal, L.N. (eds.) Handbook of Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality, pp. 169–191. North Holland, Amsterdam (1982)
2. Andreu, G., Crespo, A., Valiente, J.M.: Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In: Proceedings of ICNN 1997, International Conference on Neural Networks, vol. II, pp. 1341–1346. IEEE Service Center, Piscataway (1997)
3. Arkedev, A.G., Braverman, E.M.: Computers and Pattern Recognition. Thompson, Washington, D.C (1966)
4. Bahlmann, C., Burkhardt, H.: The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. IEEE Trans. Pattern Anal. Mach. Intell. 26(3), 299–310 (2004)
5. Bicego, M., Cristani, M., Murino, V., Pękalska, E., Duin, R.P.W.: Clustering-based construction of hidden markov models for generative kernels. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 466–479. Springer, Heidelberg (2009)

6. Bicego, M., Pękalska, E., Tax, D.M.J., Duin, R.P.W.: Component-based discriminative classification for hidden markov models. Pattern Recognition 42(11), 2637–2648 (2009)
7. Bishop, C.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
8. Bognár, J.: Indefinite Inner Product Spaces. Springer, Heidelberg (1974)
9. Borg, I., Groenen, P.: Modern Multidimensional Scaling. Springer, New York (1997)
10. Bunke, H., Buhler, U.: Applications of approximate string matching to 2D shape recognition. Pattern Recognition 26(12), 1797–1812 (1993)
11. Bunke, H., Riesen, K.: Graph classification on dissimilarity space embedding. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, p. 2. Springer, Heidelberg (2008)
12. Burghouts, G.J., Smeulders, A.W.M., Geusebroek, J.M.: The distribution family of similarity distances. In: Advances in Neural Information Processing Systems, vol. 20 (2007)
13. Calana, Y.P., Reyes, E.B.G., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: ICPR 2010, pp. 177–180 (2010)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), software `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
15. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
16. Devijver, P.A., Kittler, J.V.: Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs (1982)
17. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, Inc., Chichester (2001)
18. Duin, R.P.W.: Compactness and complexity of pattern recognition problems. In: Perneel, C. (ed.) Proc. Int. Symposium on Pattern Recognition 'In Memoriam Pierre Devijver', pp. 124–128. Royal Military Academy, Brussels (1999)
19. Duin, R.P.W., Pękalska, E.: Non-euclidean dissimilarities: Causes and informativeness. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 324–333. Springer, Heidelberg (2010)
20. Duin, R.P.W., Pękalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recognition Letters (in press, 2011)
21. Duin, R.P.W., Pękalska, E., Harol, A., Lee, W.-J., Bunke, H.: On euclidean corrections for non-euclidean dissimilarities. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 551–561. Springer, Heidelberg (2008)
22. Duin, R.P.W.: Non-euclidean problems in pattern recognition related to human expert knowledge. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2010. LNBIP, vol. 73, pp. 15–28. Springer, Heidelberg (2011)
23. Duin, R.P.W., Loog, M., Pękalska, E., Tax, D.M.J.: Feature-based dissimilarity space classification. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 46–55. Springer, Heidelberg (2010)
24. Duin, R., Pękalska, E.: On refining dissimilarity matrices for an improved nn learning. In: ICPR, pp. 1–4 (2008)
25. Edelman, S.: Representation and Recognition in Vision. MIT Press, Cambridge (1999)
26. Fu, K.: Syntactic Pattern Recognition and Applications. Prentice-Hall (1982)

27. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press (1990)
28. Gold, S., Rangarajan, A.: A graduated assignment algorithm for graph matching. IEEE Trans. Pattern Anal. Mach. Intell. 18(4), 377–388 (1996)
29. Goldfarb, L.: A new approach to pattern recognition. In: Kanal, L., Rosenfeld, A. (eds.) Progress in Pattern Recognition, vol. 2, pp. 241–402. Elsevier (1985)
30. Haasdonk, B.: Feature space interpretation of SVMs with indefinite kernels. IEEE TPAMI 25(5), 482–492 (2005)
31. Haasdonk, B., Burkhardt, H.: Invariant kernel functions for pattern analysis and machine learning. Machine Learning 68(1), 35–61 (2007)
32. Haasdonk, B., Pękalska, E.: Indefinite kernel fisher discriminant. In: ICPR, pp. 1–4 (2008)
33. Ibba, A., Duin, R.P.W., Lee, W.J.: A study on combining sets of differently measured dissimilarities. In: ICPR, pp. 3360–3363. IEEE (2010)
34. Kim, S.W., Duin, R.P.W.: On improving dissimilarity-based classifications using a statistical similarity measure. In: Bloch, I., Cesar Jr., R.M. (eds.) CIARP 2010. LNCS, vol. 6419, pp. 418–425. Springer, Heidelberg (2010)
35. Kruskal, J.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27 (1964)
36. Laub, J., Roth, V., Buhmann, J.M., Müller, K.R.: On the information and representation of non-euclidean pairwise data. Pattern Recognition 39(10), 1815–1826 (2006)
37. Lee, W.J., Duin, R.P.W.: An inexact graph comparison approach in joint eigenspace. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 35–44. Springer, Heidelberg (2008)
38. Lichtenauer, J.F., Hendriks, E.A., Reinders, M.J.T.: Sign language recognition by combining statistical DTW and independent classification. IEEE Trans. Pattern Analysis and Machine Intelligence 30(11), 2040–2046 (2008)
39. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pękalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recognition 39(10), 1827–1838 (2006)
40. Luo, B., Wilson, R.C., Hancock, E.R.: Spectral embedding of graphs. Pattern Recognition 36(10), 2213–2230 (2003)
41. Mottl, V., Seredin, O., Dvoenko, S., Kulikowski, C.A., Muchnik, I.B.: Featureless pattern recognition in an imaginary hilbert space. In: ICPR, vol. 2, pp. 88–91 (2002)
42. Mottl, V., Dvoenko, S., Seredin, O., Kulikowski, C., Muchnik, I.: Featureless pattern recognition in an imaginary Hilbert space and its application to protein fold classification. In: Perner, P. (ed.) MLDM 2001. LNCS (LNAI), vol. 2123, pp. 322–336. Springer, Heidelberg (2001)
43. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library (COIL-100), Columbia University (1996)
44. Neuhaus, M., Bunke, H.: Bridging the Gap Between Graph Edit Distance and Kernel Machines. World Scientific (2007)
45. Orozco-Alzate, M., Duin, R.P.W., Castellanos-Domínguez, G.: A generalization of dissimilarity representations using feature lines and feature planes. Pattern Recognition Letters 30(3), 242–254 (2009)
46. Pękalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recognition Letters 23(8), 943–956 (2002)

47. Pękalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(6), 729–744 (2008)
48. Pękalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39(2), 189–208 (2006)
49. Pękalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. World Scientific, Singapore (2005)
50. Pekalska, E., Haasdonk, B.: Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Trans. Pattern Anal. Mach. Intell. 31(6), 1017–1032 (2009)
51. Porro-Muñoz, D., Duin, R.P.W., Talavera-Bustamante, I., Orozco-Alzate, M.: Classification of three-way data by the dissimilarity representation. Signal Processing 91(11), 2520–2529 (2011)
52. Porro-Muñoz, D., Talavera, I., Duin, R.P.W., Hernández, N., Orozco-Alzate, M.: Dissimilarity representation on functional spectral data for classification. Journal of Chemometrics, n/a–n/a (2011)
53. Ripley, B.D.: An introduction to statistical pattern recognition. Cambridge University Press, Cambridge (1996)
54. Shepard, R.: The analysis of proximities: Multidimensional scaling with an unknown distance function. i. Psychometrika 27, 125–140 (1962)
55. Spillmann, B., Neuhaus, M., Bunke, H., Pękalska, E.z., Duin, R.P.W.: Transforming strings to vector spaces using prototype selection. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 287–296. Springer, Heidelberg (2006)
56. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press (2008)
57. Ulas, A., Duin, R.P., Castellani, U., Loog, M., Mirtuono, P., Bicego, M., Murino, V., Bellani, M., Cerruti, S., Tansella, M., Brambilla, P.: Dissimilarity-based detection of schizophrenia. International Journal of Imaging Systems and Technology 21(2), 179–192 (2011)
58. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc. (1998)
59. Watanabe, S.: Pattern Recognition: Human and Mechanical. Wiley (1985)
60. Webb, A.: Statistical pattern recognition. Wiley (2002)
61. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research 10, 207–244 (2009)
62. Wilson, C., Garris, M.: Handprinted character database 3. Tech. rep., National Institute of Standards and Technology (February 1992)
63. Wilson, R., Luo, B., Hancock, E.: Pattern vectors from algebraic graph theory. IEEE Trans. on PAMI 27, 1112–1124 (2005)
64. Wilson, R.C., Hancock, E.R.: Spherical embedding and classification. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 589–599. Springer, Heidelberg (2010)
65. Wolpert, D.H. (ed.): The Mathematics of Generalization. Addison-Wesley, Reading (1995)
66. Woznica, A., Kalousis, A., Hilario, M.: Learning to combine distances for complex representations. In: Ghahramani, Z. (ed.) ICML. ACM International Conference Proceeding Series, pp. 1031–1038. ACM (2007)
67. Xiao, B., Hancock, E.R.: Geometric characterisation of graphs. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 471–478. Springer, Heidelberg (2005)
68. Yang, L., Jin, R., Sukthankar, R., Liu, Y.: An efficient algorithm for local distance metric learning. In: AAAI. AAAI Press (2006)