Pattern Recognition Letters 33 (2012) 826–832

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

The dissimilarity space: Bridging structural and statistical pattern recognition

Robert P.W. Duin [a,*], Elżbieta Pękalska [b]

[a] Pattern Recognition Laboratory, Delft University of Technology, Delft, The Netherlands
[b] School of Computer Science, University of Manchester, United Kingdom

ARTICLE INFO

Article history:
Available online 1 May 2011

Keywords:
Dissimilarity representation
Representation set
Dissimilarity space
Vector space
Structural pattern recognition

ABSTRACT

Human experts constitute pattern classes of natural objects based on their observed appearance. Automatic systems for pattern recognition may be designed on a structural description derived from sensor observations. Alternatively, training sets of examples can be used in statistical learning procedures. They are most powerful for vectorial object representations. Unfortunately, structural descriptions do not match well with vectorial representations. Consequently it is difficult to combine the structural and statistical approaches to pattern recognition.

Structural descriptions may be used to compare objects. This leads to a set of pairwise dissimilarities from which vectors can be derived for the purpose of statistical learning. The resulting dissimilarity representation bridges thereby the structural and statistical approaches.

The dissimilarity space is one of the possible spaces resulting from this representation. It is very general and easy to implement. This paper gives a historical review and discusses the properties of the dissimilarity space approaches illustrated by a set of examples on real world datasets.

1. Introduction

Pattern recognition is an intrinsic human ability that starts in infancy. Already in early childhood we are able to recognize complex patterns such as smells, voices, faces and toys. It takes however a long time before we can accurately describe how we do this, and perhaps sometimes we are not able to outline it. It is not directly clear how we recognize a particular person as John or some toy as a car. This holds for children as well as for adults. Even experts such as medical doctors have difficulties in defining sharply how a particular heart disease is recognized from an ECG. If they specify their relevant observations to the designer of an expert system for coding an automatic recognition system, interestingly, this explicitation has to go through several iterations. In such a process the expert is being confronted with his mistakes in order to clarify the recognition steps.

The human expert is aware of different pattern classes but is in trouble to motivate them in terms of explicit observations. We often see that he is tempted to use the structure of the objects: the relations between internal parts when complicated objects need to be described. In such cases he is not able to stick to a straight set of directly measurable observations such as color, weight and size. Consequently, he phrases his arguments not only in measurable quantities, but also in a wordy description of the structure. If we want to incorporate this approach to building an automatic recognition system, both, a set of sensors as well as a structural model may be necessary.

It is difficult for an expert to define exactly how sensor outputs have to be combined into a decision function that accurately determines a pattern class. Such functions are usually optimized by learning from examples when using procedures developed in multi-variate statistics, statistical pattern recognition or machine learning. This works well for objects represented in vector spaces by measurements or by features derived from measurements. The lack of structural knowledge or the lack of its representation may hereby be partially compensated by statistical properties derived from a (large) set of examples. On the contrary, it is much more difficult to apply such procedures in order to optimize decision functions based on structural models. Often reasoning procedures are used to measure the similarity between the models of objects to be recognized and those of (examples of) objects of the pattern classes to be distinguished. The use of examples is thereby usually restricted to their storage in order to determine the most similar ones for new objects to be recognized. The generalization is thereby in the models and not derived from an analysis of the set of given examples.

In order to build automatic machines that mimic human recognition, the expert becomes gradually more and more aware of his own decision making, while he tries to make his recognition process explicit. Or, to phrase it somewhat differently, he is forced to do so. In this process he becomes more conscious of his own internal recognition procedure. The result is a description in terms

[*] Corresponding author. Tel.: +31 15 278 6143; fax: +31 15 278 1843.
E-mail addresses: r.duin@ieee.org, r.p.w.duin@tudelft.nl (R.P.W. Duin), pekalska@cs.man.ac.uk (E. Pękalska).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.04.019

of both observations and models. Initially, most people are not aware of these two aspects of the decision making. Only after being challenged to define it sharply they become conscious of how they do it.

To phrase it more generally: consciousness splits recognition in *empiricism* and *rationalism*. As a result, mechanical recognition devices can be built, as we have in place approximate procedures for both processes. In spite of Dennett's belief that consciousness is an illusion and does not play a role in the world (Dennett, 1991), the existence of automatic recognition systems proves the contrary. The expert becomes *conscious* of his recognition process when he is forced to clearly outline it such that it can be programmed in a computer.

The fact that experts experience the split of their knowledge into observations and structural models may lead to clear and computerizable representations, but has also severe drawbacks. Observations disposed from structural relations may be represented by vectors related to sensors or sensor samples. This representation is poor as dependencies are not included. They may be partially reconstructed from a statistical analysis of a large set of observations (learning from examples). Structural models, on the other hand, may preserve dependencies and relations, but it is difficult to enrich such a knowledge-based description by new observations. This is a fundamental problem in epistemology: the truth of a fact or statement may be proven either by showing an example or by reasoning. Consequently, research areas such as pattern recognition and artificial intelligence have been separated (around 1970) as they demanded different approaches and, consequently, attracted different types of researchers. Also within the pattern recognition domain the topics of statistical and structural recognition obtained their own sessions and tracks on conferences and were organized by different committees. The consciousness split in the expert's mind has thereby even a social impact.

The possibility of the merge of the two separated sources of knowledge has intrigued various researchers over the decades. Thereby, it has been a research topic in pattern recognition from its early days. Watanabe (1985) and especially Fu (1982) pointed to several possibilities of how to combine the approaches of statistical and structural pattern recognition based on information theoretic considerations and stochastic syntactical descriptions. In spite of their inspiring research efforts, it hardly resulted in practical applications. The 'gap' between the statistical and structural approaches continued, stimulated by the associated social gap. Around 1985, Goldfarb proposed to unify the two directions by replacing the feature-based representation of individual objects by distances between structural object models. 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 1990, Goldfarb himself focussed on a very fundamental, structural approach with a long-term perspective that the models might be learned from example – the Evolving Transformation System (Goldfarb et al., 2004). So, he shifted the focus of his research from using the structural models as they are in a statistical context to shifting the structural model building out of the expert's mind into a formalism by which a generative structural model will be directly learned from examples. This ambitious research line is still unfinished.

After 1995, the authors of this paper started to study the first proposal by Goldfarb to replace the traditional feature representation by a distance representation that could be applied to structural models. They called it the *dissimilarity representation* as it allows various non-metric, indefinite or even asymmetric proximity measures. An inspiration for this approach was also the observation that a human observer is primary triggered by object differences and that the description in terms of features and models comes second (see Edelman, 1999). We consciously observe differences, while similarity is a usually assumed context in which comparisons take place. The analysis of dissimilarities, mainly for visualization, was studied much earlier in the domain of psychonomy in the 1960s (e.g. by Shepard, 1962; Kruskal, 1964). 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 Spillmann et al. (2006) and others such as Wilson and Hancock (2010) and Mottl et al. (2001), Mottl et al. (2002). They realized that the dissimilarity representation might be profitable for the use of vector space learning procedures on top of matching algorithms used for graphs and strings. Traditionally, template matching (i.e. the nearest neighbor rule) was used as the main classification procedure. Since the dissimilarity representation puts just mild restrictions on the distance measure, it allows for the use of non-Euclidean distances as resulting from many procedures common in graph matching and shape recognition. Important application areas may be the classification of spectra, histograms, image recognition and, recently, multi-instance learning.

In addition, it appears that a dissimilarity representation defined on top of a traditional feature-based representation may lead to interesting classifiers with unique advantages (Pękalska and Duin, 2006; Pękalska and Duin, 2005). In general, such classifiers have a good performance and may be robust against variations in scale in the same feature space. If the nearest neighbor distances are small in some areas and large in other areas, such a dissimilarity-based classifier is less affected by that than a classifier in the original feature space.

There are two essential ways of constructing a vector space from a dissimilarity representation (Pękalska and Duin, 2005; Pękalska et al., 2008): (Pseudo-) Euclidean embedding and the so-called dissimilarity space. The first one, based on an (extension of) linear multi-dimensional scaling, has recently been studied extensively. Especially, an intriguing issue was the topic of embedding the given non-Euclidean dissimilarities into a vector space such that the obtained distances are sufficiently accurate in comparison to the original dissimilarities. Several aspects have beenresearched: why non-Euclidean dissimilarities arise, whether they are informative, and how to deal with them. These issues are challenging, both from theoretical and mathematical point of view.

The second way of handling the dissimilarity representation, the postulation of the dissimilarity space, raises less problems and is of high interest for practical applications. It can, without problems, be used for almost any kind of dissimilarity measure. Moreover, it has good asymptotic properties and offers the possibility of an adjustable computational complexity. It is the target of this paper to discuss and illustrate the use of the dissimilarity space and to elaborate on a number of interesting properties.

2. Road map

In this section we discuss shortly the various approaches for object representation. The road map of Fig. 1 shows their position between the real world objects and the generalization in either a vector space or by template matching.

Feature representation: the initial feature set consists of the object properties that are potentially judged as relevant for classification by the application expert or the system designer. It may sometimes be chosen as samples of the data, e.g. the pixels of an image.

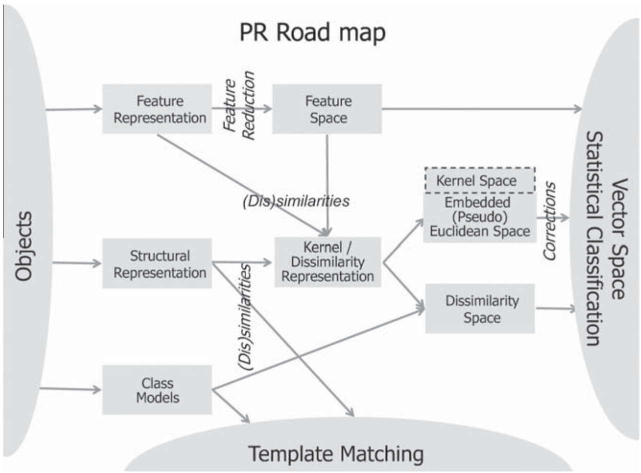

Fig. 1. Road map of various representations between objects and generalization as discussed in the paper.

Feature space: this is the most relevant subset of the original feature set or their combination obtained by an analysis of the training set. This feature space can directly be used for the training of classifiers. It may also become an input for the computation of kernels or dissimilarities (see below).

Structural representation: a structural representation includes relations between individual measurements. It is thereby richer than a feature representation. Also if these features cover relations between separate object measurements, a structural representation may include the relation between several such features.

Dissimilarity representation: pairwise dissimilarities are computed between the training examples and objects from the representation set. The measure is defined on top of a structural representation or in the feature space. A representation set is a set of either chosen or generated prototypes, in particular, it may also be the entire training set.

Kernel representation: similarities may be used, instead of dissimilarities. In case of structurally represented objects there is little difference. For feature spaces kernels are usually based on inner products. It often holds for the commonly used dissimilarity measures that they are translation independent. Kernels in feature spaces use similarity measure which are usually rotation independent.

Pseudo-Euclidean space: given a dissimilarity representation or a kernel matrix we may find whether the Euclidean distances or the Euclidean inner products perfectly reproduce the original dissimilarities or kernels (embedding). This is only possible if the original dissimilarity measure is Euclidean or if the kernel satisfies the Mercer conditions (Pękalska and Duin, 2005). In other situations, the so-called pseudo-Euclidean embedding can be found for a symmetric definite measure for which a different distance measure and a different inner product are defined. Such a pseudo-Euclidean space can be transformed into one or more Euclidean spaces by various correction procedures. The standard statistical classification algorithms may be applied in these spaces.

Kernel space: existing classification procedures are written in terms of inner products. By the use of the so-called kernel trick, the inner products are substituted by nonlinear kernel values. As a result, the classifiers implicitly operate in a (reproducing kernel) Hilbert space for which the inner product corresponds to the kernel measure. For kernel matrices, this is the isometric embedded Euclidean space mentioned above. The embedding itself, however, is now not needed. This may also hold for classifiers that can operate in pseudo-Euclidean spaces, i.e. they can handle the indefinite inner product definition (Ong et al., 2004; Canu et al., 2003; Pękalska and Haasdonk, 2009). This is not true for the standard Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000), but may still be applied with some restriction (Haasdonk, 2005).

Dissimilarity space: a straightforward way of handling the dissimilarity representation is by interpreting the dissimilarity vectors (defined by dissimilarities between objects and prototypes) as features. This is in fact a data-dependent mapping to a vector space which we equip with traditional inner product. This can be done without almost any restriction as it can be applied to an indefinite, asymmetric distance measure. The dissimilarity space is thereby a simple straightforward way to map a structural representation into a vector space in which further traditional statistical training procedures may be applied.

Class models: class models may be constructed in various ways. We do not indicate them here to maintain clarity. However, both statistical (e.g. pdf's, HMMs and one-class classifiers) and structural (e.g. by the Evolving Transformation System) class models can be built. They can be used directly for classification by template matching as well as for building a dissimilarity space.

3. The dissimilarity space

The dissimilarity space is a vector space in which the dimensions are defined by dissimilarity vectors measuring pairwise dissimilarities between examples and individual objects from the so-called representation set R. Hence, a dissimilarity representation $D(X,R)$ is addressed as a data-dependent mapping $D(\cdot,R) : X \to \mathbb{R}^n$ from an initial set of objects X to a dissimilarity space, equipped with the traditional inner product and Euclidean metric. The representation set can be chosen as the complete training set T, a set of carefully selected or constructed prototypes or an arbitrary set of labeled or unlabeled objects (even objects from a test set S can be considered). Here, we choose R to be either the training set T or its subset. All training and test objects are now represented in this space by dissimilarity vectors whose elements express degrees of differences to individual objects from R.

In order to show the possibilities of the dissimilarity space we use the Pendigits dataset (Alimoglu and Alpaydin, 1997). It consists of a training set T of 7494 handwritten digits 0–9, obtained from 30 writers, and an independent testset S of 3498 digits, obtained from another set of 14 writers. All digits were transformed into string sequences of vectors of constant length and compared by using a vector cost function (Spillmann et al., 2006). In the below experiments we thereby use a 7494 × 7494 training set $D_T = D(T,T)$ with pairwise dissimilarities between all training objects and a 3498 × 7494 test set $D_S = D(S,T)$ with pairwise dissimilarities between the 3498 test objects and the 7494 training objects. Both, training set and test set, can thereby be represented in a 7494-dimensional vector space, the dissimilarity space.

In traditional approaches to structural pattern recognition test objects are classified by the 1-NN rule determining the class of the nearest neighbor in the training set. In this example test objects have to be compared to all 7494 objects in the training set, leading to a classification error of 0.0374 for the 1-NN rule. For $k > 1$ no better results are found by the k-NN rule. In this approach, however, the given training set $D(T,T)$ is not used for the learning process, except for the storage of examples. Hence, there is a possible room for improvement.

The dissimilarity space can be used for object representation and construction of classifiers. Fig. 2 shows a 2D subspace of the dissimilarity space built by all training objects as found by LDA. The projected objects come from 500 randomly selected test

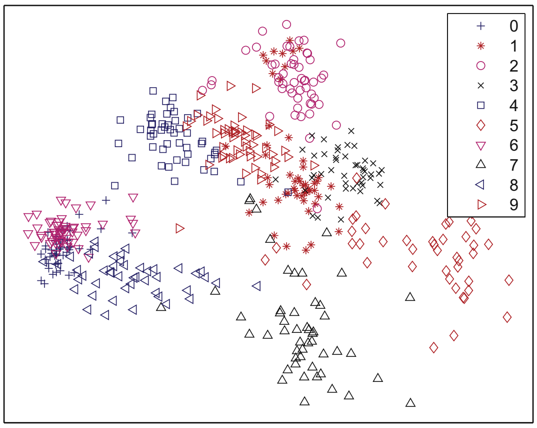

Fig. 2. A random subset of the test set of the Pendigits dissimilarity data projected on a 2D LDA space computed from the training set.

objects, not used for the definition of this space. Clearly, a class structure can be observed, which is promising for building classifiers.

To illustrate the possibilities of this representation we first compute learning curves for growing subsets, randomly drawn from the training set T. Since the representation set R is chosen as T, $R = T$, the dimension of the dissimilarity space equals the size of T. In this space we build the 1-NN classifier and a linear SVM, using the LIBSVM package (Chang and Lin, 2001). The results are compared with the 1-NN rule directly applied to the original dissimilarities; see Fig. 3.

We observe that the direct use of the dissimilarities leads to better results than the other two approaches for small training sets, while the dissimilarity approach is more beneficial for large training sets. Relating test objects to all training objects, followed by computing distances over dissimilarity vectors gradually takes over in the latter case. This can be understood by realizing that the original dissimilarities, as obtained from string matching, contain some noise, e.g. caused by the arbitrary chosen starting points of the vectors of the string vector description used for describing the digit structure. By taking into account all dissimilarities to all objects in the training set this noise is reduced w.r.t. the result obtained by just a single pairwise comparison. The figure also shows clearly that a global classifier such as the linear SVM finally outperforms the local sensitive 1-NN rule. The fact that this classifier uses all dissimilarities in the training set D_T pays off. This is in contrast to the direct 1-NN rule applied to the given dissimilarities which does not make use of D_T.

Another interesting aspect of the learning curves in Fig. 3 is that they are not saturated. It is to be expected that a significantly better performance may be obtained by enlarging the training set by, e.g. by a factor of 10. A severe drawback of these approaches is that dissimilarity representations based on a large training set can be computationally demanding, especially if they are based on string matching or graph matching procedures. For that reason it is worthwhile to study the pruning of the representation set R in order to reduce it from the complete training set T to a significantly smaller set. The representation of the training set thereby becomes a rectangular matrix $D'_T = D(T, R)$. As a result, test objects need a simpler representation. The test set is now $D'_S = D(S, R)$.

Several procedures have been studied for the reduction of the representation set R (Pękalska et al., 2006; Lozano et al., 2006; Calana et al., 2010; Riesen et al., 2007): random selection, cluster analysis, geometric distribution, feature selection, nearest neighbor based prototype selection and genetic algorithms. A simple and fast procedure, suitable for selecting a small R out of a large training set, is a forward search based on the 1-NN performance of the reduced dissimilarity matrix D'_T (Pękalska et al., 2006; Calana et al., 2010). It is fast as it is based on the given dissimilarities. Fig. 4 compares performance of linear SVM in dissimilarity spaces built by either systematic or random selection of R. The dashed line shows the asymptotic performance of the direct 1-NN rule by using all training objects. It is clear that systematic selection is better than random selection. Already for R consisting of 30 objects, a better classifier is found than by using the 1-NN rule on all 7494 training objects. The best performance, an error of 0.0189, is found for an R of 500 selected objects. This is, given the size of 3498 objects of the test set, significantly better than the error of 0.0244 found for the classifier on the original representation set $R = T$.

Learning curves may be computed for representation sets of various sizes; see Fig. 5. It shows that an increased performance may be expected for all sizes of the representation set if large training sets become available. Moreover, it can be observed that training sets can be not only larger, but also smaller than the representation set. An interesting application of this observation is that an unlabeled test set may be added to the representation set and used for an improved performance (in transductive learning). This will demand an increased computational effort, of course, in which also the dissimilarities between the test objects have to be computed.

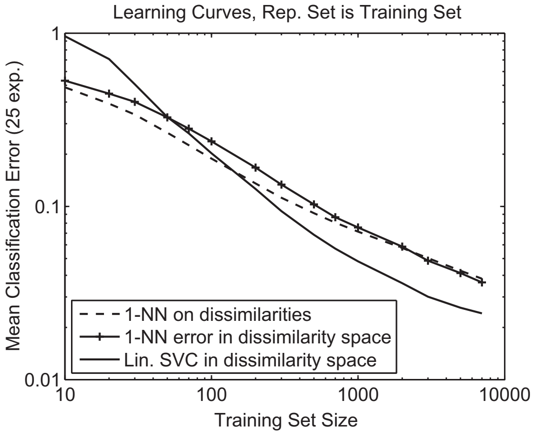

Fig. 3. Learning curves for the Pendigits dataset for the direct 1-NN rule on the given dissimilarity data and two classifiers in the dissimilarity space defined by $R = T$.

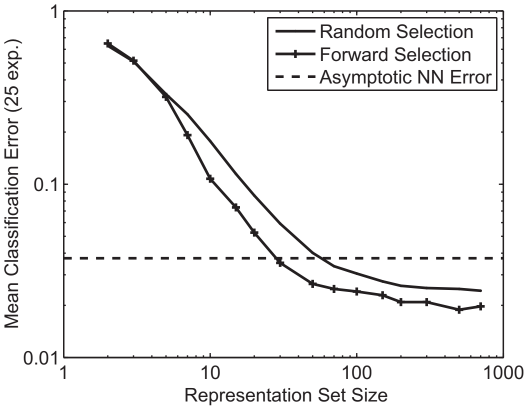

Fig. 4. Performance of SVM-1 in dissimilarity spaces of the Pendigits dataset using either random or optimized representation sets. The entire training set is used for training the classifier. The dashed line shows the asymptotic performance of the 1-NN classifier.

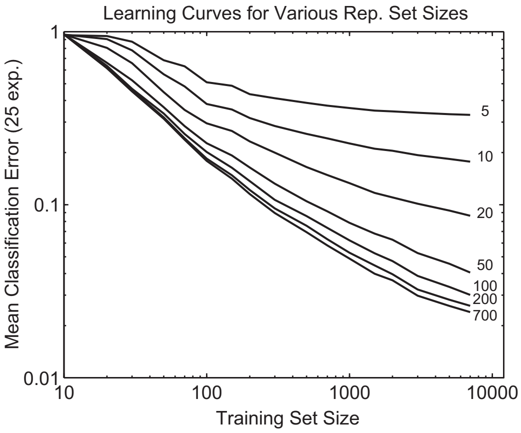

Fig. 5. Learning curves for the Pendigits dataset for SVM-1 in dissimilarity spaces defined over various fixed-size representation sets.

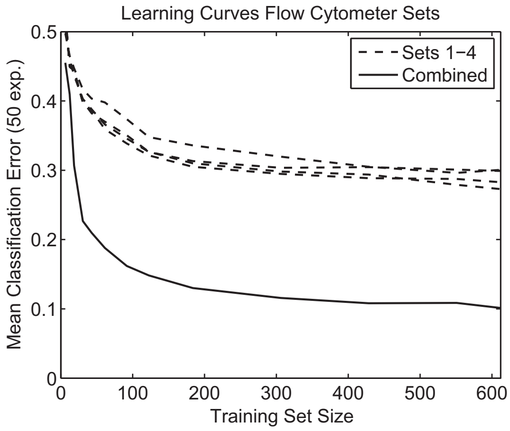

Fig. 6. Learning curves for the four Flow Cytometer datasets for SVM-1 in the dissimilarity space defined for $R = T$. The bottom curve applies to the average combination of the four datasets.

An interesting phenomenon, which attracts more and more attention, is that multiple dissimilarity matrices, available for the same data, may often be combined in a profitable way. There are several options (Pękalska and Duin, 2001), e.g. concatenation of the dissimilarity spaces, combining classifiers trained on the individual dissimilarity spaces and a (weighted) average of the dissimilarity matrices, resulting in a new matrix. Especially, the last option is attractive as the resulting representation has the same shape and size as the original ones. Comparison is thereby straightforward and reliable. A question is about weights to be used for the input matrices. This issue is very similar to the problem studied in kernel metric learning (Lanckriet et al., 2004). In case the classification performances of the individual dissimilarities are about equal, equal weights are appropriate. We illustrate this by the following example.

The dataset we used is based on 612 FL3-A DNA flow cytometer histograms from breast cancer tissues in 256 resolution. 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 in 3 classes: aneuploid (335 patients), diploid (131) and tetraploid (146). Every tube contained about 20,000 cells per patient. We removed the first and the last bin of every histogram as here outliers are collected, thereby obtaining 254 bins per histogram. As the histograms may suffer from an incorrect calibration in the horizontal direction (DNA content) we computed for every pairwise dissimilarity between two histograms the multiplicative correction factor for the bin positions that minimizes their dissimilarity using the $\ell 1$ norm. This representation makes use of the shape structure of the histograms and removes an invariant (the varying original calibration).

The four dissimilarity matrices based on the four tubes have about the same performance, as shown by the top curves in Fig. 6 based on the linear SVM in a hold-out evaluation experiment. Representation set equals the training set, while the test set is the hold-out set. We normalized the four sets such that the average dissimilarities between different objects is one. The learning curve for the averaged dissimilarity matrix, which is the lower curve in the figure, is strikingly better than the ones based on the individual

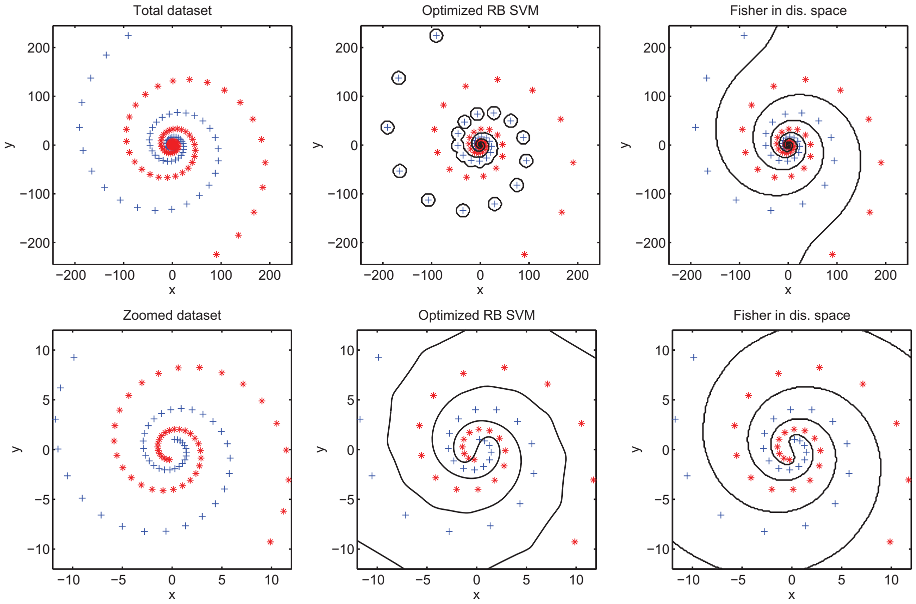

Fig. 7. A spiral example with 100 objects per class. Top row shows complete data sets, while bottom row presents the zoom of the spiral center. Training set consists of 50 objects per class, systematically sampled. The middle column shows the training set and SVM with an optimized radial basis function; 17 out of 100 test objects are erroneously classified. The right column shows the Fisher linear discriminant (no regularization) computed in the dissimilarity space derived from the Euclidean distances. All test objects are correctly classified.

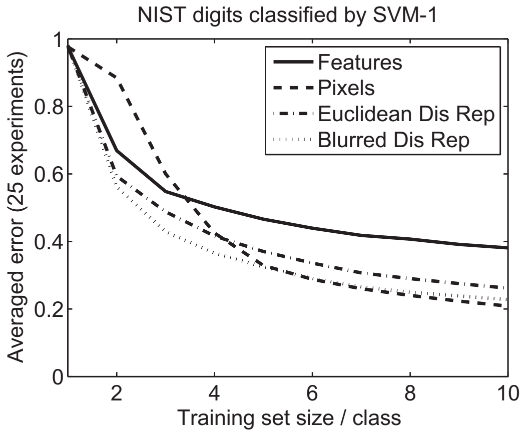

Fig. 8. Learning curves of SVM-1 compared for three representations of the same digit recognition problem: features, pixels and dissimilarities.

sets. An explanation is that the original dissimilarities suffer from noise as 20,000 counts for a 256 bin histogram results in many noisy bin counts.

An often returning question about the dissimilarity representation is whether it is any better than a feature representation. This cannot be answered. The performances depend on the choices of the dissimilarity measure or features and thereby on the ability of the analyst or application expert to express his knowledge on the problem in a particular way. Thereby, the preference for one representation or the other depends on the application as well as the expert.

A direct comparison of the two approaches can be made in case of given feature representations and dissimilarities computed by the Euclidean distances in the feature space. This is certainly not the target situation for the dissimilarity approach as much better dissimilarity measures might be defined on the original data even before features are extracted. Nevertheless, such comparisons can be made. In (Duin et al., 2010) it is shown in a contest on 300 datasets that linear classifiers in a Feature based Dissimilarity Space (FDS) perform very well. Fig. 7 illustrates such a comparison for a 2D spiral problem. It shows that a linear classifier in the FDS has a type of neighborhood adaptive kernel characteristic and, unlike the radial basis SVM, is insensitive for scale variations.[1]

For a comparison of various representations based on the same set of objects we summarize here the results of an experiment reported more extensively elsewhere (Duin et al., 2010). It is based on four representations of a subset of a NIST digit database (Wilson and Garris, 1992) with 32×32 resampled images: features (10 moments), pixels, a dissimilarity space built by distances between the original images and a dissimilarity space built by distances between blurred images. The latter is less sensitive for small variations in the way digits are positioned (shifts and rotations) and written. The learning curves in Fig. 8 show that this is better than the direct use of the image distances. The features are not sufficiently well defined to be of use anywhere. The pixel representation needs a sufficiently large training set and then finally outperforms the other approaches. The fact that pixel representations finally win is to be expected as they just store the universe of objects.

4. Conclusions

This paper elaborates on the advantages of a dissimilarity representation as the one that fills the gap between statistical learning and structural models. The elements of this representation encode degrees of (dis)similarity between pairs of examples or between examples and optimized (selected or generated) prototypes or between examples and class models. It is a powerful representation, especially when one defines the dissimilarity measure based on structural approaches and builds statistical classifiers in the corresponding (dis)similarity space. Such a dissimilarity space can also result from an integration or combination of various dissimilarity representations. In this way, general proximity measures, including non-Euclidean or non-metric dissimilarities and indefinite similarities, either symmetric or asymmetric can be used with success and elegantly handled for the pattern recognition tasks. Such measures are important to study as they result from various applications, especially when invariance or robustness is incorporated (Haasdonk and Burkhardt, 2007; Jacobs et al., 2000). Learning in dissimilarity spaces extends the kernel methods into indefinite kernel approaches and beyond (Pękalska and Haasdonk, 2009; Hochreiter and Obermayer, 2006).

In summary, the clear benefits of the dissimilarity space approach are:

- the straightforward use of arbitrary, yet application meaningful, dissimilarity measure, defined either on vectorial or structural representations,
- handling of difficult problems (Pękalska et al., 2008); an alternative to the NN rule for small and moderate training sets,
- an adjustable computational complexity which may be significantly smaller than the NN rule,
- a simple but still powerful combining rule in case multiple dissimilarity measures are available.

The development of procedures based on the dissimilarity representation is still going on. It is thereby not yet possible to formulate definite recipes for its use. A few general guidelines and insights can be presented here w.r.t. the choice of the dissimilarity measure, the selection of prototypes and the classifier.

The formulation of the dissimilarity measure is an opportunity to include specific application knowledge. Especially knowledge of invariants should be used as much as possible. As dissimilarities are computed pairwise it is more easy to incorporate such knowledge than to remove invariants globally, for all objects simultaneously. For instance, in a pairwise comparison the alignment of objects by rotation, translation or deformation is better possible than by a global normalization of orientation, position or shape. This has to be exploited. Any dissimilarity measure that is just a straightforward summation of local differences without such operations indicates room for improvement.

Prototype selection is primarily important for computational reasons. It will speed up the classification of new objects as less dissimilarities have to be computed. It may also be desired to reduce the dimensionality of the dissimilarity space for classifiers based on density estimation as they suffer from overtraining. In our experience the linear SVM in the full dissimilarity space which has as many dimensions as training objects, is very good. For large training sets this classifier might be too complex for computational reasons, both for training as well as for testing. Random selection of prototypes may do reasonably well, unless a very small set (e.g. less than 20) of prototypes has to be found. Systematic procedures based on feature selection may then yield better results.

Although any classifier designed for vector spaces can be used in the dissimilarity space, some may be preferred. In case no prototype selection is applied and especially for larger training set sizes, there is a high correlation to be expected between dimensions related to similar objects. The larger the training set, the more similar objects are to be expected. So a classifier is needed that can handle this problem. When the total training set is used

[1] This example has been published before in (Duin et al., 2010).

for representation and the numbers of dimensions and training objects are the same, a linear classifier is sufficient. As stated above, the linear SVM may be a good choice. For reduced sizes of the representation set density based classifiers may be appropriate. We had good results with LDA and QDA based on the assumptions of normal distributions with equal or different covariances. Depending on the dissimilarity measure this assumption may be applicable.

In future research we will study the design of optimal representation sets, in particular the use of out-off-training-set objects, resulting in transductive learning. Another interesting topic is the design of dissimilarity measures for various applications, e.g. the variants of the multiple-instance-learning problem. In addition, we will further investigate relations between kernel methods and methods in (dis)similarity spaces in which the (dis)similarity characteristic of the representation is explicitly taken into account.

Acknowledgments

This research is financially supported by the FET programme within the EU FP7, under the SIMBAD project (contract 213250) and the British Engineering and Physical Science Research Council (EPSRC), project number EP/D066883/1. The authors wish to express their gratitude to Professor Horst Bunke who intensively followed and encouraged their research for more than 10 years and who recognized at a very early moment its possible significance for integrating structural and statistical approaches to pattern recognition.

References

Alimoglu, F., Alpaydin, E., 1997. Combining multiple representations and classifiers for pen-based handwritten digit recognitio. In: ICDAR. pp. 637–640.

Calana, Y.P., Reyes, E.B.G., Orozco-Alzate, M., Duin, R.P.W., 2010. Prototype selection for dissimilarity representation by a genetic algorithm. In: ICPR 2010. pp. 177–180.

Canu, S., Mary, X., Rakotomamonjy, A., 2003. Functional learning through kernel. In: Advances in Learning Theory: Methods. Models and Applications, NATO Science Series III: Computer and Systems Sciences, vol. 190. IOS Press, Amsterdam, pp. 89–110.

Chang, C.-C., Lin, C.-J., 2001. LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines. Cambridge University Press, UK.

Dennett, D., 1991. Consciousness Explained. Penguin.

Duin, R.P.W., Loog, M., Pe kalska, E., Tax, D.M.J., 2010. Feature-based dissimilarity space classification. In: Unay, D., Cataltepe, Z., Aksoy, S. (Eds.), Recognizing Patterns in Signals, Speech, Images, and Videos, ICPR 2010, vol. 6388, Springer, pp. 46–55.

Edelman, S., 1999. Representation and Recognition in Vision. MIT Press, Cambridge.

Fu, K., 1982. Syntactic Pattern Recognition and Applications. Pretice-Hall.

Goldfarb, L., 1985. A new approach to pattern recognition. In: Kanal, L., Rosenfeld, A. (Eds.), Progress in Pattern Recognition, vol. 2. Elsevier, pp. 241–402.

Goldfarb, L., Gay, D., Golubitsky, O., Korkin, D., 2004. What is a structural representation? Tech. rep. URL <http://www.cs.unb.ca/tech-reports/files/tr04-165.pdf>.

Haasdonk, B., 2005. Feature space interpretation of SVMs with indefinite kernels. IEEE TPAMI 25 (5), 482–492.

Haasdonk, B., Burkhardt, H., 2007. Invariant kernel functions for pattern analysis and machine learning. Machine Learning 68 (1), 35–61.

Hochreiter, S., Obermayer, K., 2006. Support vector machines for dyadic data. Neural Computation 18 (6), 1472–1510.

Jacobs, D., Weinshall, D., Gdalyahu, Y., 2000. Classification with non-metric distances: Image retrieval and class representation. IEEE TPAMI 22 (6), 583–600.

Kruskal, J., 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29, 1–27.

Lanckriet, G.R.G., Cristianini, N., Bartlett, P.L., Ghaoui, L.E., Jordan, M.I., 2004. Learning the kernel matrix with semidefinite programming. J. Mach. Learn. Res. 5, 27–72.

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

Mottl, V., Dvoenko, S., Seredin, O., Kulikowski, C., Muchnik, I., 2001. Featureless pattern recognition in an imaginary Hilbert space and its application to protein fold classification. In: Perner, P. (Ed.), Machine Learning and Data Mining in Pattern Recognition, LNCS, vol. 2123. Springer, Berlin/Heidelberg, pp. 322–336.

Mottl, V., Seredin, O., Dvoenko, S., Kulikowski, C., Muchnik, I., 2002. Featureless pattern recognition in an imaginary Hilbert space. In: ICPR 2002. pp. II: 88–91.

Ong, C., Mary, X., Canu, S., Smola, S.A.J., 2004. Learning with non-positive kernels. In: International Conference on Machine Learning. Brisbane, Australia, pp. 639–646.

Pękalska, E., Duin, R.P.W., 2001. On combining dissimilarity representations. In: Kittler, J., Roli, F. (Eds.), Multiple Classifier Systems, LNCS, vol. 2096. Springer-Verlag, pp. 359–368.

Pękalska, E., Duin, R.P.W., 2005. The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore.

Pękalska, E., Duin, R.P.W., 2006. Dissimilarity-based classification for vectorial representations. In: ICPR2006. pp. III: 137–140.

Pękalska, E., Duin, R.P.W., 2008. Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Trans. Syst. Man. Cybernet., Part C: Appl. Rev. 38 (6), 729–744.

Pękalska, E., Duin, R.P.W., Paclı́k, P., 2006. Prototype selection for dissimilarity-based classifiers. Pattern Recognit. 39 (2), 189–208.

Pękalska, E., Haasdonk, B., 2009. Kernel discriminant analysis for positive definite and indefinite kernels. IEEE TPAMI 31 (6), 1017–1032.

Riesen, K., Neuhaus, M., Bunke, H., 2007. Graph embedding in vector spaces by means of prototype selection. In: GbRPR. Lecture Notes in Computer Science, vol. 4538. Springer, pp. 383–393.

Shepard, R., 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. Psychometrika 27, 125–140.

Spillmann, B., Neuhaus, M., Bunke, H., Pekalska, E., Duin, R.P.W., 2006. Transforming strings to vector spaces using prototype selection. In: SSPR/SPR. LNCS, vol. 4109. Springer, pp. 287–296.

Watanabe, S., 1985. Pattern Recognition: Human and Mechanical. Wiley.

Wilson, C., Garris, M., 1992. Handprinted character database 3. Tech. rep., National Institute of Standards and Technology.

Wilson, R.C., Hancock, E.R., 2010. Spherical embedding and classification. In: SSPR/SPR. LNCS, vol. 6218. Springer, pp. 589–599.