ELSEVIER

Pattern Recognition Letters 23 (2002) 943–956

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Dissimilarity representations allow for building good classifiers

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

Pattern Recognition Group, Laboratory of Applied Physics, Delft University of Technology, Lorentzweg 1,
2628 CJ Delft, The Netherlands

Received 19 January 2001; received in revised form 25 September 2001

Abstract

In this paper, a classification task on dissimilarity representations is considered. A traditional way to discriminate between objects represented by dissimilarities is the nearest neighbor method. It suffers, however, from a number of limitations, i.e., high computational complexity, a potential loss of accuracy when a small set of prototypes is used and sensitivity to noise. To overcome these shortcomings, we propose to use a normal density-based classifier constructed on the same representation. We show that such a classifier, based on a weighted combination of dissimilarities, can significantly improve the nearest neighbor rule with respect to the recognition accuracy and computational effort. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Similarity representations; Normal density-based classifiers

1. Introduction

The challenge of automatic pattern recognition is to develop computer methods which learn to distinguish among a number of classes represented by examples. First, an appropriate representation of objects should be found. Then, a decision rule can be constructed, which discriminates between different categories and which is able to generalize well (achieve a high accuracy when novel examples appear). One of the possible representations is based on similarity or dissimilarity relations between objects. When properly defined, it might be advantageous for solving class identification problems. Such a recommendation is supported by the fact that (dis)similarities can be considered as a connection between perception and higher-level knowledge, being a crucial factor in the process of human recognition and categorization (Goldstone, 1999; Edelman, 1999; Wharton et al., 1992).

In contrast to this observation, objects are conventionally represented by characteristic features (Duda et al., 2001). In some cases, however, a feasible feature-based description of objects might be difficult to obtain or inefficient for learning purposes, e.g., when experts cannot define features in a straightforward way, when data are high dimensional, or when features consist of both continuous and categorical variables. Then, the use of dissimilarities, built directly on

[*] Corresponding author. Tel.: +31-15-278-1845; fax: +31-15-278-6740.

E-mail address: ela@ph.tn.tudelft.nl (E. Pękalska).

0167-8655/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-8655(02)00024-7

measurements, e.g., based on template matching, is an appealing alternative. Also, in some applications, e.g., 2D shape recognition (Edelman, 1999), the use of dissimilarities makes the problem more viable.

The nearest neighbor method (NN) (Cover and Hart, 1967) is traditionally applied to dissimilarity representations. Although this decision rule is based on local neighborhoods, i.e., one or a few neighbors, it is still computationally expensive, since dissimilarities to all training examples have to be found. Another drawback is that it potentially decreases its performance when the training set is small. To overcome such limitations and improve the recognition accuracy, we propose to replace this method by a more global decision rule. Such a classifier is constructed from a training set represented by the dissimilarities to a set of prototypes, called the representation set. If this set is small, it has the advantage that only a small set of dissimilarities has to be computed for its evaluation, while it may still profit from the accuracy offered by a large training set.

Throughout this paper, all our investigations are devoted to dissimilarity representations, assuming that *no other* representations (e.g., features) are available for the researcher. The goal of this work is to propose a novel, advantageous approach to learn *only* from dissimilarity (distance) representations, dealing with classification problems in particular. Our experiments will demonstrate that the tradeoff between the recognition accuracy and the computational effort is significantly improved by using a normal density-based classifier built on dissimilarities instead of the NN rule. This paper is organized as follows. In Section 2, a more detailed description of dissimilarity representations and the decision rules considered are given. Section 3 describes the datasets used and the experiments conducted. The results are discussed in Section 4 and the conclusions are summarized in Section 5. The essential idea of this paper has been published in Electronic Letters (Pękalska, 2001). Some earlier elements of the presented research can be found in (Duin et al., 1999; Pękalska and Duin, 2000).

2. Learning from dissimilarities

To construct a classifier on dissimilarities, the training set T of size n (having n objects) and the representation set R (Duin, 2000) of size r will be used. R is a set of prototypes covering all classes present. R is chosen to be a subset of T ($R \subseteq T$), although, in general, R and T might be disjunct. In the learning process, a classifier is built on the $n \times r$ distance matrix $D(T,R)$, relating all training objects to all prototypes. The information on a set S of s new objects is provided in terms of their distances to R, i.e., as an $s \times r$ matrix $D(S,R)$.

2.1. Nearest neighbor method

A straightforward approach to dissimilarity representations leads to the nearest neighbor rule (Cover and Hart, 1967; Fukunaga, 1990) or more generally to instance-based learning (Aha et al., 1991). Such classifiers make use of distance information in a rank-based way. The NN rule, in its simplest form, i.e., 1-NN rule, assigns a new object to the class of its nearest neighbor from the representation set R by finding minima in the rows of $D(S,R)$. The k-NN decision rule is based on majority voting: an unknown object becomes a member of the class the most frequently occurring among the k-NN.

2.2. Normal density-based linear/quadratic classifiers

The novelty of our approach relies on interpreting distances as a representation of a dissimilarity space. In particular, $D(T,R)$ is treated as a description of a space where each dimension corresponds to the distance to a prototype. In general, $D(x,R)$ defines a vector consisting of r distances found between the object x and all the objects in the representation set R, i.e., if $R = \{p_1, \dots, p_r\}$, then $D(x,R) = [D(x,p_1), \dots, D(x,p_r)]^{\mathrm{T}}$. Therefore, $D(\cdot,R)$ is seen as a mapping onto an r-dimensional dissimilarity space. In this convention, neither x nor R refers to points in a feature space, instead they refer to the objects themselves. The advantage of such a representation is that any traditional classifier operating on feature spaces can be used.

Moreover, it can be optimized by using training sets larger than the given representation set. This does not complicate the decision rule, but does increase its accuracy.

The choice of Bayesian classifiers (Fukunaga, 1990) assuming normal distributions, is a natural consequence of the central limit theorem (CLT) applied to dissimilarities. It is supported by the observation that most of the commonly-used dissimilarity measures, e.g., Euclidean distance or Hamming distance, are based on sums of differences between measurements. The CLT states that the sum of random variables tends to be normally distributed in the limit, provided that none of the variances of the sum's components dominates. Therefore, summation-based distances (built from many components) tend to be approximately normally distributed, which suggests that Bayesian classifiers, i.e., the (R)LNC/(R)QNC, (Regularized) Linear/Quadratic Normal density-based Classifier (Ripley, 1996), should perform well in dissimilarity spaces. For a 2-class problem, the LNC based on the representation set R is given by

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

and the QNC is given by

$$f(D(x,R)) = \sum_{i=1}^{2} (-1)^{i} (D(x,R) - \boldsymbol{m}_{(i)})^{\mathrm{T}} C_{(i)}^{-1} (D(x,R) - \boldsymbol{m}_{(i)}) + 2\log\frac{P_{(1)}}{P_{(2)}} + \log\frac{|C_{(2)}|}{|C_{(1)}|}, \qquad (2)$$

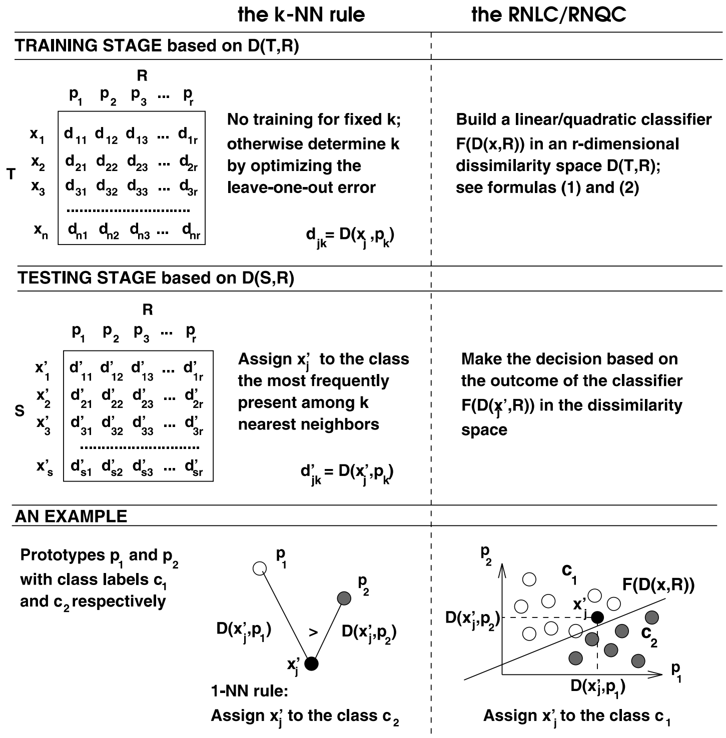
where C is the sample covariance matrix, $C_{(1)}$ and $C_{(2)}$ are the estimated class covariance matrices, and $\boldsymbol{m}_{(1)}$ and $\boldsymbol{m}_{(2)}$ are the mean vectors, all found in the dissimilarity space $D(T,R)$. $P_{(1)}$ and $P_{(2)}$ are the prior probabilities. When C becomes singular, its inverse cannot be computed, therefore the regularized version is used (Ripley, 1996), resulting in the RLNC. When the regularization is used in case of the QNC, the resulting classifier becomes the RQNC.

2.3. Nearest neighbor versus normal density-based classifiers

The NN rule can learn complex boundaries and generalizes well for large training sets. Asymptotically, its recognition error is bounded from above by twice the Bayes error (the smallest error possible) (Cover and Hart, 1967). In practice, however, it is often difficult to get a sufficiently large set R to reach such an accuracy. Another drawback of the NN method is that the classification results may be affected by the presence of noisy prototypes. Therefore, other discrimination functions, such as the RLNC/RQNC, might be more advantageous on dissimilarity representations (Pękalska and Duin, 2000; Duin et al., 1999), especially when the number of prototypes is small. They may perform much better since they become less local in their decisions by operating on larger training sets. By using weighted combinations of dissimilarities, they suppress the influence of noisy prototypes as well.

Since training can be done off-line, here we are only concerned with the computational effort needed for an evaluation of a novel object. Given r prototypes in the representation set, the complexity of the RLNC is $\mathrm{O}(r)$ (products and sums), while the RQNC is $\mathrm{O}(r^2)$. The 1-NN rule requires $\mathrm{O}(r)$ comparisons while of the k-NN rule needs at least $\mathrm{O}(r)$ and at most $\mathrm{O}(r\log(r))$ comparisons. Thereby, the k-NN rule might seem to be preferable. However, our point in this paper is that the k-NN method requires more prototypes than the RLNC/RQNC to reach the same accuracy (see Section 4). Since the cost of computing dissimilarities is very high (dissimilarities are beneficial for data described by a large amount of measurements, i.e., images, signals, spectra), we consider the number of prototypes being crucial for judging the computational complexity. Therefore, we claim that the RLNC can improve the k-NN rule with respect to the recognition accuracy and computational effort. The same holds for RQNC if the representation set is not very large, depending on the effort needed for computing dissimilarities.

The difference between the two approaches can be briefly summarized as follows: the k-NN



No training for fixed k; otherwise determine k by optimizing the leave-one-out error

Build a linear/quadratic classifier F(D(x,R)) in an r-dimensional dissimilarity space D(T,R); see formulas (1) and (2)

Assign x'j to the class the most frequently present among k nearest neighbors

Make the decision based on the outcome of the classifier F(D(x'j,R)) in the dissimilarity space

Prototypes p1 and p2 with class labels c1 and c2 respectively

1-NN rule: Assign x'j to the class c2

Assign x'j to the class c1

Fig. 1. The *k*-NN rule and the RLNC/RQNC on a dissimilarity representation.

rule (as used here) operates *directly* on the individual dissimilarities, while the RLNC/RQNC is defined for the representation spaces $D(T, R)$, treating dissimilarities as input features (see Fig. 1).

3. Datasets and the experimental set-up

A number of experiments is conducted to compare the results of the *k*-NN rule and the RLNC/RQNC built on dissimilarities. They are designed to observe and analyze the behavior of these classifiers in relation to different sizes of the representation and training sets. Smaller representation sets are of interest, because of lower complexity for representation and evaluation of new objects. This is important for the storage purposes, as well as for the computational aspect. Our concern is then how much can be gained by using a smaller representation set and a larger training set.

Three datasets, described in Table 1, are studied: two versions of the NIST digit sets (Wilson

Table 1
Datasets used in experiments

	Pixel-based digit	Contour digit	Chromosome band
Dimensionality	16 × 16 normalized	64 × 64 resampled	20–100
# Classes	10	10	24
# Objects per class	200	200	60
# Objects of a design set L	1000	1000	720
# Objects of a testing set S	1000	1000	720
Dissimilarity measure	Euclidean	Modified Hausdorff	DNA-difference

and Garris, 1992), 10 classes each, and the small part of the chromosome band profiles (Houtepen, 1994; Houtepen and Duin, 1994; Houtepen and Vossepoel, 1994), consisting of 24 classes. We used the following summation-based dissimilarity measures. The Euclidean distance on a blurred version of the digit dataset, the modified-Hausdorff distance (Dubuisson and Jain, 1994) on a contour representation of the same set and a DNA difference measure on the chromosome banding profiles. Since, for digits and chromosome bands, no natural features arise from the application, constructing dissimilarities is an interesting possibility to deal with such recognition problems. The modified-Hausdorff distance measures the difference between two sets (here two contours) $A = \{a_1, \dots, a_g\}$ and $B = \{b_1, \dots, b_h\}$ and is defined as $\max(h(A,B), h(B,A))$, where $h(A,B) = (1/g) \times \sum_{a \in A} \min_{b \in B} \|a - b\|$. All chromosome band profiles, given in 20–100 samples per profile, are first normalized by scaling the integrated density to a constant value relating to the DNA-content. Then, for each profile, the intensity values corresponding to the fixed percentages of the DNA-content create a vector. The Euclidean distance between such vectors describes the dissimilarity measure used, called here as the DNA-difference (see Fig. 2).

The experiments are performed 25 times for randomly chosen training and testing sets for each dataset under investigation. In a single experiment, each dataset is randomly split into two equal-sized sets: the design set L and the testing set S. L serves for obtaining both the representation set R and the training set T. After R is chosen, a number of training sets of different sizes is then considered. First, T is set to be equal to R and then it is gradually enlarged by adding random objects until it becomes L.

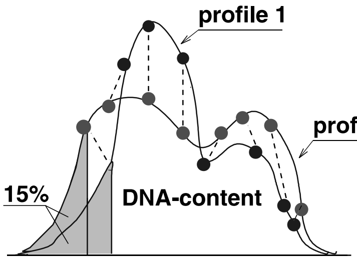


Fig. 2. The DNA-difference distance. First, chromosome band profiles are scaled such that the integrated density is constant. Then, a number of fixed percentages of the DNA-content (e.g., 15%) is considered and the corresponding intensity values (marked by circles) create vectors, for which Euclidean distance is computed. The dashed lines indicate the correspondence between intensity values of two profiles.

3.1. Selection of the representation set

There exists a number of ways to select the representation set R from the design set L. Here, we do not study the best possible set R for the given problem. Instead, we focus on illustrating the usefulness of our approach. Therefore, only three criteria are considered, referred to as: random, most-dissimilar (MD) and condensed nearest neighbor (CNN). The first two methods work on each class separately. The random method is based on a random selection of objects. The MD criterion selects objects which differ the most from each other. It starts from a randomly chosen object and then keeps adding new ones which are the most dissimilar to all objects already chosen. Summarizing, in a single experiment, initially, a subset of the design set L is used for representation. Then, it is increased gradually by adding new objects according to the given criterion, until it is equivalent to the complete set L. In this way a number of representation sets of different sizes can be studied.

The CNN criterion is based on the condensed nearest neighbor method (Hart, 1968; Devijver and Kittler, 1982) developed to reduce the computational effort of the 1-NN rule. The CNN method finds a subset of the training set so that the 1-NN rule gives a zero error when tested on the remaining objects. Here, the representation set R becomes the condensed set found on the design set L. In contrast to the other selections, the size of R is fixed in a single experiment and determined by the method itself. However, since the training sets differ in all experiments, the number of prototypes may vary. Therefore, the size of R is averaged over all runs when reported in Tables 2–4.

Table 2
Averaged generalization error (with its standard deviation) of the *k*-NN rule and the RLNC/RQNC for the pixel-based digit dataset and three selection methods of *R*

r_c	*k*-NN	RLNC		RQNC	
Random					
10	17.5 (0.4)	15.6 (0.3)	8.6 (0.1)	19.0 (0.5)	4.4 (0.1)
20	12.5 (0.3)	10.2 (0.1)	7.1 (0.1)	10.3 (0.2)	4.6 (0.1)
50	8.3 (0.2)	6.6 (0.2)	5.5 (0.1)	5.6 (0.2)	4.7 (0.1)
70	7.1 (0.2)	5.8 (0.1)	5.1 (0.1)	5.0 (0.1)	4.6 (0.1)
90	6.4 (0.1)	5.1 (0.1)	5.0 (0.1)	4.6 (0.1)	4.6 (0.1)
Most-dissimilar					
10	34.2 (0.5)	21.3 (0.6)	7.9 (0.2)	24.3 (0.6)	4.7 (0.1)
20	25.9 (0.4)	12.1 (0.4)	6.2 (0.1)	12.3 (0.4)	5.3 (0.1)
50	12.4 (0.2)	6.0 (0.1)	5.1 (0.1)	5.3 (0.1)	4.5 (0.2)
70	8.4 (0.2)	5.2 (0.1)	4.9 (0.1)	4.9 (0.1)	4.3 (0.1)
90	6.3 (0.1)	4.9 (0.1)	4.9 (0.1)	4.6 (0.1)	4.4 (0.1)
r_c	1-NN	RLNC		RQNC	
CNN					
20	10.6 (0.2)	8.5 (0.2)	5.7 (0.1)	8.7 (0.4)	4.6 (0.1)

The RLNC/RQNC errors presents the worst (left column) and the best (right column) results, depending on the training size, achieved for the fixed r_c.

Table 3
Averaged generalization error (with its standard deviation) of the *k*-NN rule and the RLNC/RQNC for the contour digit dataset and three selection methods of *R*

r_c	*k*-NN	RLNC		RQNC	
Random					
10	24.4 (0.4)	21.3 (0.3)	11.1 (0.2)	34.9 (0.8)	8.0 (0.2)
20	17.1 (0.2)	15.6 (0.3)	9.8 (0.2)	21.2 (0.5)	7.4 (0.2)
50	10.6 (0.2)	10.3 (0.2)	9.0 (0.2)	9.9 (0.2)	7.2 (0.2)
70	8.9 (0.1)	9.2 (0.2)	8.7 (0.2)	8.2 (0.2)	7.2 (0.2)
90	7.9 (0.2)	8.5 (0.2)	8.2 (0.2)	8.2 (0.2)	8.3 (0.2)
Most-dissimilar					
10	40.0 (0.4)	26.8 (0.6)	11.3 (0.2)	29.0 (0.8)	7.5 (0.2)
20	31.4 (0.5)	20.0 (0.4)	9.8 (0.2)	16.4 (0.3)	7.0 (0.2)
50	16.4 (0.3)	10.5 (0.2)	8.9 (0.1)	9.1 (0.2)	7.0 (0.2)
70	10.8 (0.2)	8.5 (0.1)	8.9 (0.2)	8.2 (0.2)	7.0 (0.2)
90	8.2 (0.2)	8.2 (0.2)	8.3 (0.2)	8.2 (0.2)	8.3 (0.2)
r_c	1-NN	RLNC		RQNC	
CNN					
24	12.8 (0.2)	12.6 (0.3)	9.5 (0.2)	12.3 (0.4)	6.8 (0.2)

The RLNC/RQNC errors presents the worst (left column) and the best (right column) results, depending on the training size, achieved for the fixed r_c.

3.2. Regularization of the normal density-based classifiers

The RLNC/RQNC (Ripley, 1996), assuming normal distributions with equal or different class covariance matrices, respectively, is built for different training sets. The regularized versions are used to prevent the estimated covariance matrices from being singular (when, e.g., n, the size of T, approaches r, the size of R, for the RLNC).

Table 4
Averaged generalization error (with its standard deviation) of the *k*-NN rule and the RLNC for the chromosome band dataset and three selection methods of *R*

r_c	*k*-NN	RLNC	
Random			
10	30.4 (0.4)	25.3 (0.3)	21.0 (0.3)
15	27.1 (0.3)	23.3 (0.3)	20.8 (0.3)
20	25.6 (0.3)	21.7 (0.2)	20.7 (0.2)
25	24.6 (0.3)	21.0 (0.2)	20.5 (0.2)
Most dissimliar			
10	47.9 (0.9)	28.0 (0.3)	20.4 (0.2)
15	36.0 (0.6)	22.8 (0.2)	20.1 (0.2)
20	28.8 (0.4)	20.8 (0.2)	20.2 (0.2)
25	25.2 (0.3)	20.3 (0.3)	20.2 (0.2)
r_c	1-NN	RLNC	
CNN			
15	30.9 (0.5)	22.2 (0.3)	20.5 (0.3)

The RLNC/RQNC errors presents the worst (left column) and the best (right column) results, depending on the training size, achieved for the fixed r_c.

Regularization takes care that the inverse operation is possible (necessary to build the classifiers) by emphasizing the variances of the sample covariance matrix with respect to the off-diagonal elements.

When T is about the size of R, the estimation of the covariance matrices is poor and a relatively large regularization should be used. In such cases, different regularizations may significantly influence the performance of the RLNC/RQNC. For sufficiently large training sets, these matrices are well defined and no regularization is needed. In our experiments, the regularization parameters are chosen to be fixed values. Since they are not optimized, the presented results are not the best possible. For instance, C is regularized as $C_{\text{reg}} = (1 - \lambda)C + \lambda\ \text{diag}(C)$, where $\text{diag}(C)$ is a diagonal matrix consisting of the main diagonal of C and λ is at most 0.01 for training sets slightly larger than representation sets and becomes zero for larger training sets.

3.3. The algorithm

The algorithm in a single experiment with the random or most-dissimilar selection of the representation set is schematically presented below:

```
define the design set L
define the testing set S
define a vector r_R of the sizes for
the representation set R
for i = 1 to length(r_R) do
  select the set R of size r_R(i) ac-
  cording to a selection method
  determine k for the k-NN method
  error_k-NN(i) = TEST        (k-NN,
  D(S,R))
  for z = i to length(r_R) do
    choose the training set T of size
    r_R(z) such that
    T = R + objects randomly selected
    (per class) from L-R
    TRAIN (RLNC/RQNC, D(T,R))
    error_RLNC/RQNC(i,z) = TEST
    (RLNC/RQNC, D(S,R))
  end
end
```

In case of the *k*-NN rule, we studied the following fixed choices of *k*: 1, 3, 5, 7 or 9. Additionally, we tried to optimize *k* via the leave-one-out procedure on the $D(T,T)$. However, the *k* determined was always found to be one of the fixed, odd *k* mentioned above. For both digit sets, the best *k*-NN test results are found for $k = 1$ or 3, while for the chromosome data, *k* equals 7 or 9. An example of the behavior of the generalization error as a function of *k* is given in Fig. 6. Due to the small sample size of the training set, large neighborhood sizes perform bad as they average out significant details. In the experiments below we will report only the best test results for the studied values of *k*. In case of the RLNC/RQNC, the training stage is used for determining the sample covariance matrices and the mean vectors in the dissimilarity space $D(T,R)$, as presented in formulas (1) and (2).

Since for the CNN criterion the size of the set *R* is determined by the procedure, the outer loop in the above given scheme is superfluous. Another difference relates to the choice of training sets. Here, the classes are likely to be unequally present in such a set *R*, therefore the training set is constructed from *R* by adding objects, but now randomly selected from *all* the remaining examples in *L*. The generalization errors for each selection method are averaged over the repeated experiments and serve to create all figures presented in this paper.

4. Results

The generalization error rates of the k-NN rule and the RLNC/RQNC for three datasets are presented in Figs. 3–5. The k-NN results, marked by stars '$*$', are presented on the $r_c = n_c$ line. The results depend either on the random selection of the representation set (left subplots) or on the MD criterion (right subplots). Since, the k-NN results are worse in case of the MD selection, the k-NN results always refer to the random selection (also in right subplots). The RLNC's (RQNC's) curves are lines of constant classification error relating the sizes of the representation and training sets. Since all classes are equally sized (i.e., the priors $P_{(i)} = 1/c$ if c is the number of classes in formula (1)) for both the representation set and the training

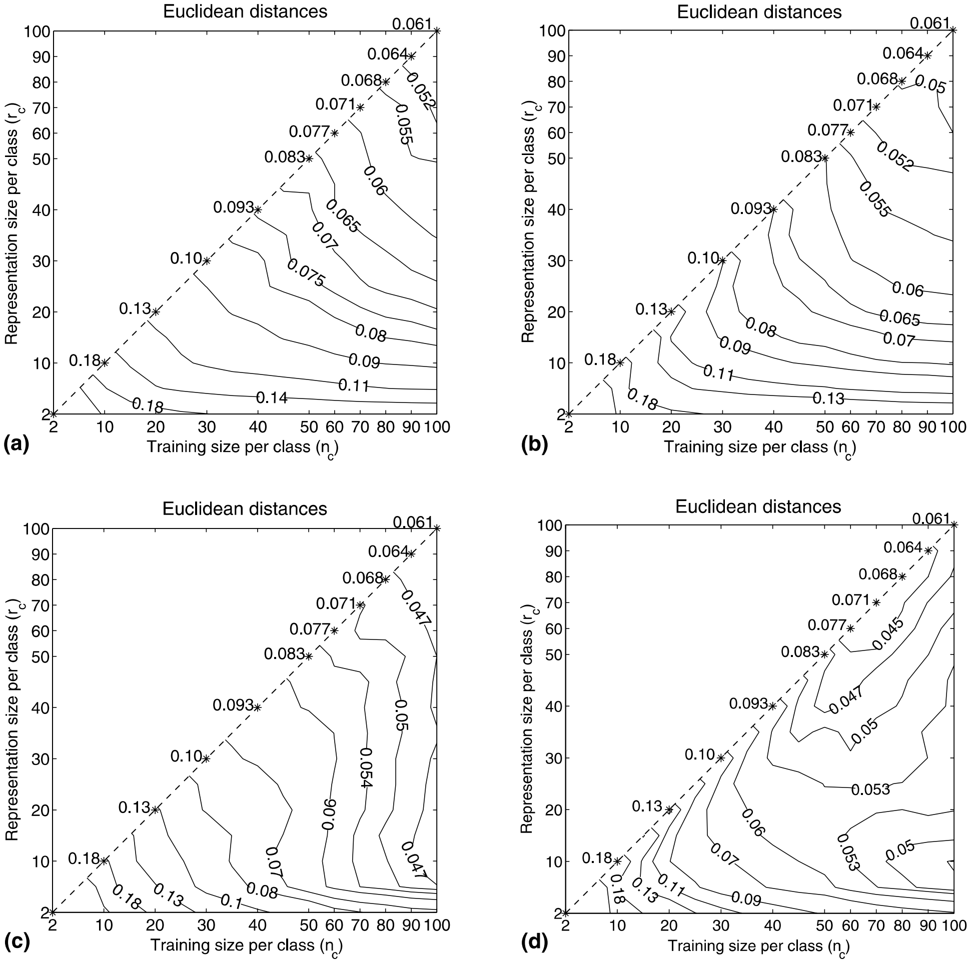


Fig. 3. Averaged generalization error of the RLNC/RQNC (top/bottom row) and the k-NN rule (indicated by stars) for the Euclidean representation of the pixel-based digit dataset. The representation sets are chosen either according to the random or MD selection. The k-NN's errors are shown for the random selection: (a) The RLNC; random selection. (b) The RLNC; MD criterion. (c) The RQNC; random selection. (d) The RQNC; MD criterion.

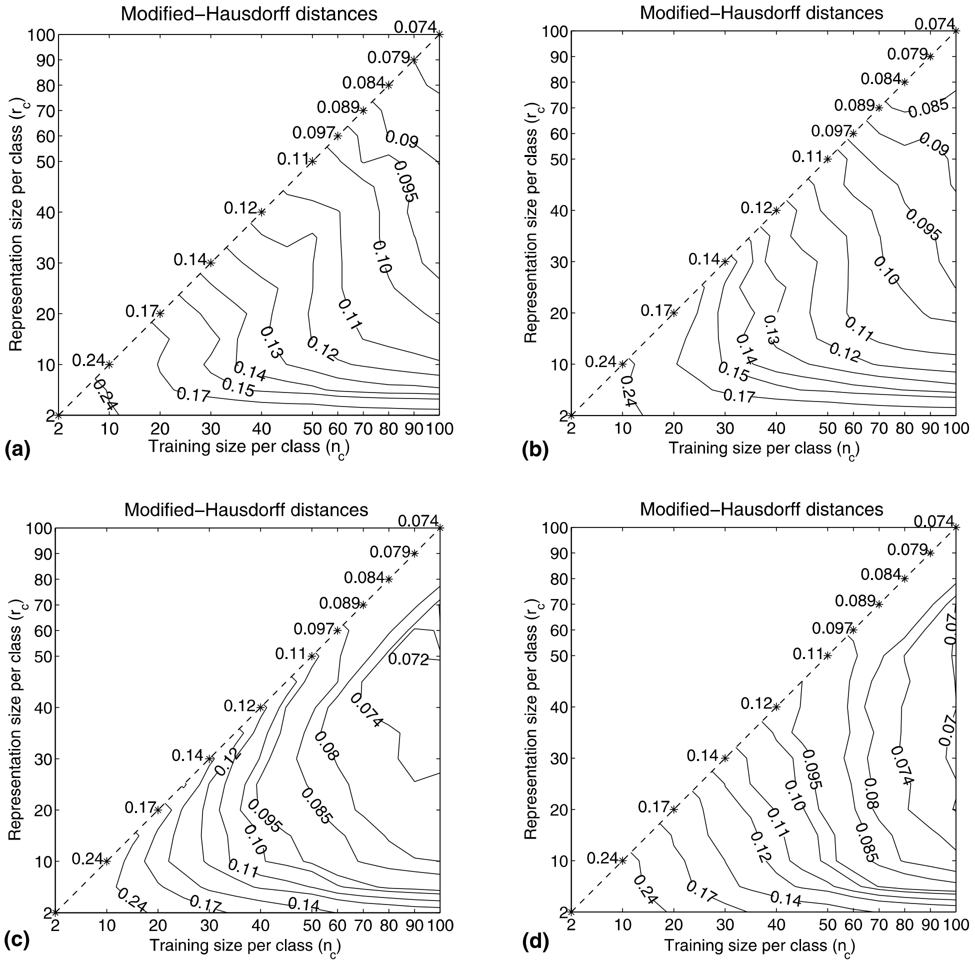


Fig. 4. Averaged generalization error of the RLNC/RQNC (top/bottom row) and the k-NN rule (marked by stars) for the modified-Hausdorff representation of the contour digit dataset. The representation sets are chosen either according to the random or MD selection. The k-NN's errors are shown for the random selection: (a) The RLNC; random selection. (b) The RLNC; MD criterion. (c) The RQNC; random selection. (d) The RQNC; MD criterion.

set, for simplicity, we will use the notation of n_c/r_c as the number of training/prototype examples per class.

Tables 2–4 summarize the results obtained for the datasets under study and three choices of the representation set. Each table is dedicated to one dataset, where for each selection method the averaged generalization errors of both the k-NN rule and the RLNC/RQNC are provided. For the RLNC/RQNC and the fixed size of the representation set, the worst and best results are reported, depending on the size of the training set. The CNN selection provides only one R of a fixed size, therefore only one row can be filled for this method.

4.1. The k-NN rule versus the RLNC

When T and R are identical, the RLNC (with error curves starting on the $r_c = n_c$ line in Figs. 3–5), mostly yields a better performance than the equivalent k-NN rule based on the same R (compare also the k-NN results with the worst cases of the RLNC in Tables 2–4).

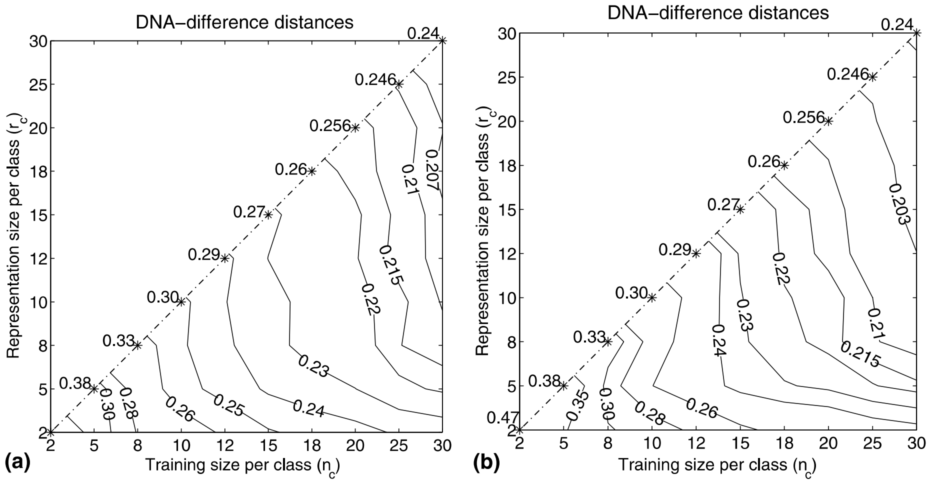


Fig. 5. Averaged generalization error of the RLNC and the k-NN rule (indicated by stars) for the DNA-difference representation of the chromosome band dataset. The representation sets are chosen either according to the random or MD selection. The k-NN's errors are shown for the random selection: (a) The RLNC; random selection. (b) The RLNC; MD criterion.

When the size of R is fixed (i.e., in the horizontal directions of Figs. 3–5), the classifiers have the same computational complexity for an evaluation of new objects. However, larger training sets reduce the error rate of the RLNC by a factor of 2 in comparison to the k-NN's error (based on the same R). For instance, in Fig. 3(a), we observe that the classification error of 0.18 is achieved by the k-NN rule based on $r_c = 10$ prototypes for which the RLNC offers a higher accuracy if trained also with $n_c = 10$ objects, reaching 0.09 when this size is increased to 100. In Fig. 5(a) the k-NN error of 0.33 obtained for $r_c = 8$ prototypes is already improved by the RLNC up to 0.22 when based on $n_c = 30$ training objects.

In other words, for a chosen representation set R (thus fixed computational complexity) the RLNC's error, with the increase of training size, decreases significantly to the values that can only be obtained by the k-NN method if it is based on a much larger R. For instance, in Fig. 4(a), the RLNC built on $r_c = 10$ prototypes (and the training set of $n_c = 100$ objects) reaches an accuracy (an error of 0.12) for which the k-NN rule needs 40 objects in its representation set. The computational load with respect to the number of computed dissimilarities of the RLNC for the same classification accuracy is thereby reduced to 25%.

Following the RLNC's curves of constant error, it can be observed, that for larger training sets, much smaller representations sets are needed for the same performance. The RLNC may sometimes demand only half the computational effort for evaluation of new objects when compared to the k-NN method. Also, for the fixed, possibly larger training set (i.e., in the vertical directions of the considered figures), the RLNC constructed on a somewhat smaller representation set, might gain a similar or higher accuracy than the k-NN rule, but now based on $D(T,T)$. This is observed, e.g., in Fig. 3(a) for $n_c = 40$. The k-NN yields an error of 0.093 and the RLNC reaches a smaller error when trained on $D(T,R)$ with R consisting of $r_c \geqslant 20$.

Comparing the two selection criteria: random and MD in Tables 2–4, it can be noticed that the performance of the k-NN rule based on the MD criterion is much worse than based on a random selection. A possible interpretation is as follows. When R is small, it only consists of the objects that differ much from each other. It contains the most remote examples, potentially also outliers, which negatively affect the behavior of the k-NN method (thus demonstrating its sensitivity to noise). When

R becomes larger, the objects become more similar to each other and the performance is better, however, still worse than reached in random selection. By comparing the best and worst cases of both selection methods in Tables 2–4, it can be observed that the RLNC's errors are often much larger for the worst cases (identical T and R) of the MD selection. The best cases seem to be more alike. However, Figs. 3(b), 4(b) and 5(b) show that the RLNC's curves are very steep in the beginning (starting from the diagonals) for the MD criterion. Therefore, it is possible that when R is kept fixed, a smaller training set is needed (when T is kept fixed, a smaller representation set is required) for the RLNC to keep the same performance as in case of the random selection. For instance, for R consisting of $r_c = 15$ prototypes, the RLNC's error of 0.09 is reached with about $n_c = 50$ training objects in Fig. 3(a) and $n_c = 30$ training examples in Fig. 3(b).

Since the best k-NN results for both digit datasets are found mainly for $k = 1$ or 3 (see Fig. 6), the 1-NN rule based on the CNN criterion obtains better results than the k-NN rule in case of the other two selections (the CNN representation set is optimized for this classifier). However, the RLNC still outperforms the 1-NN rule. The RLNC for the CNN selection might generalize better than in the other two criteria for identical R and T (compare the worst cases of the RLNC in Tables 2 and 3).

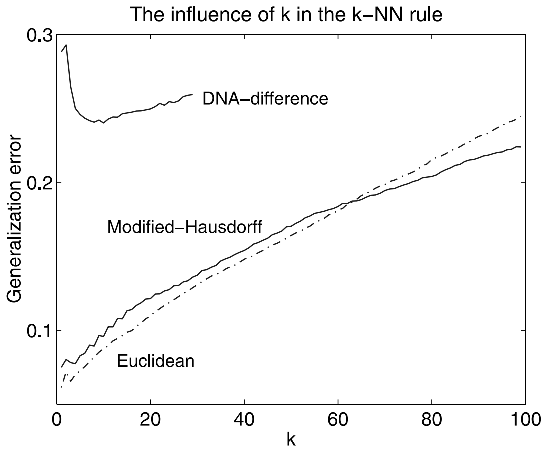


Fig. 6. The averaged performance of the k-NN rule as a function of k for the training set T consisting of $n_c = 100$ objects for the Euclidean and modified-Hausdorff representations and of $n_c = 30$ objects for the DNA-difference.

4.2. The RLNC versus the RQNC

In general, the RQNC performs better than the RLNC for both digit datasets, whose results can be studied in Tables 2 and 3, and Figs. 3(c), (d) and 4(c), (d). Since the RQNC is based on the class covariance matrices in the dissimilarity space, a larger number of samples is needed to obtain reasonable estimates, which is not the case for the chromosome band dataset. Therefore, no RQNC's results for this dataset are reported here.

The RQNC may reach a worse accuracy than the RLNC for identical T and R. However, following the curves of the RQNC's constant error, both smaller representation and training sets are needed for the same error when compared to the RLNC. The RQNC's curves are simply much more steeper than those of the RLNC. Thereby, the RQNC outperforms the RLNC for training sets larger than the given R. The most significant improvement can be observed for small R. For instance, the training set of $n_c = 100$ examples allows the RLNC to reach the the error of 0.049 when based on at least $r_c = 70$ prototypes see Table 2), where the RQNC for a similar performance requires only $r_c = 5$–30 prototypes (see Fig. 3(c)). When the largest training sizes are considered (the best results in Tables 2 and 3) for the fixed set R, the error of the RQNC decreases, yielding better results than the k-NN rule. Still, when the smallest errors of the RLNC and RQNC are compared, the RQNC generalizes better.

Also, for the fixed training set T, i.e., in the vertical direction in Figs. 3(c), (d) and 4(c), (d), a smaller representation set R, often allows the RQNC (trained on $D(T,R)$), to reach a better performance than the k-NN rule based on the $D(T,T)$. This is especially observable for the MD criterion. For instance, in Fig. 4(d), the RQNC, trained on $n_c = 40$ objects and $r_c \geqslant 10$ prototypes, achieves a smaller error than 0.12, given by the k-NN rule based on $r_c = 40$ prototypes.

Comparing the two selection methods: random and MD, a similar pattern to the behavior of the

RLNC can be observed; the MD criterion can work better than the random selection (compare subplots (c) and (d) in Figs. 3 and 4).

5. Discussion and conclusions

Our experiments confirm that the RLNC constructed on the dissimilarity representations $D(T,R)$ nearly always outperforms the k-NN rule based on the same R. This holds for the RQNC as well, provided that each class is represented by a sufficient number of objects. Since the computational complexity (here mainly indicated by the number of prototypes, as explained in Section 2.3) for evaluation of new objects is an important issue, our study is done with such an emphasis. We have found out that for the fixed representation set, larger training sets improve the performance of the RLNC/RQNC. Such results are compared to the k-NN based on the same R ($R \subseteq T$) and are found often better. Also, for the fixed training set T, smaller representation sets allow the RLNC/RQNC, trained on $D(T,R)$, to gain a high accuracy (especially in case of the MD criterion). When R is only somewhat smaller than T, such results can be better than those of the k-NN based on $D(T,T)$; see Fig. 7 as an exemplary illustration. The plots are shown for the fixed training set T, consisting of $n_c = 50$ objects per class for both digit datasets and 30 examples for the chromosome band dataset.

The potentially good performance of the RLNC can be understood as follows. It is in fact a weighted linear combination of dissimilarities between an object x and the representation set $R = \{p_1, \ldots, p_r\}$. It seems practical to allow a number of representative examples of each category to be involved in a discrimination process. This is already offered by the k-NN rule, however it provides an absolute answer (based on the majority vote). The k-NN method is still sensitive to noise, so the k-NN found might not include the best representatives of a class to which an object should be assigned. The training process of the RLNC, using a larger training set T, emphasizes prototypes which play a crucial role during discrimination, but it still allows other prototypes to influence the decision. The importance of prototypes is reflected in the weights. In such a way, a more globally sensitive classifier is built, which cannot be achieved by the k-NN rule.

The RQNC includes also a sum of the weighted products between pairs of distances from an object to the set R. By doing this, some interactions between the prototypes are emphasized. The RQNC is based on the class covariance matrices in the dissimilarity space, estimated separately for each class. Those matrices may really differ from class to class. Therefore, this decision rule might achieve

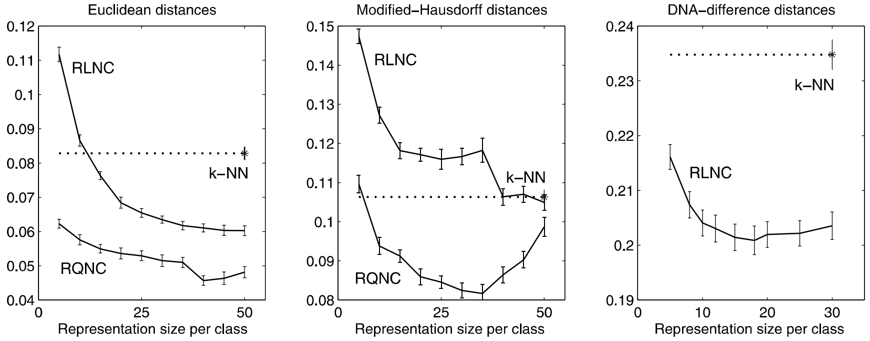


Fig. 7. Averaged generalization error (with its standard deviation) of the RLNC/RQNC based on the training set T consisting of $n_c = 50$ objects for the distance representations of the pixel-based digit (left) and the contour digit (middle) datasets, and $n_c = 30$ objects for the chromosome band (right) dataset, compared with the k-NN result based on $r_c = 50$ prototypes. The representation sets for the RLNC/RQNC are chosen according to the MD criterion.

a higher accuracy than the RLNC, where all the class covariance matrices are averaged. However, a larger number of samples (with respect to the size of R) is required to obtain reasonable estimates for all covariance matrices, thereby a good generalization ability of the RQNC. The better performance of the RQNC can be clearly observed in Figs. 3(c), (d) and 4(c), (d).

Our paper is an attempt to investigate new possibilities for learning from dissimilarity representations. In the traditional approach, the nearest neighbor rule is mostly applied to classify an unknown object. It has, however, a number of disadvantages, which are diminished by using classifiers as proposed here. The essential novelty is that such a decision rule is based on weighted combination of dissimilarities computed to all prototypes. Thereby, a more global classifier is constructed and its sensitivity to noisy prototypes is significantly reduced. Here, we use the normal density-based classifiers, however, in general, other linear or quadratic classifiers might be considered as well. Our method allows also for making use of additional objects, which although enlarge the training set, do not increase the computational complexity for evaluation of unknown examples. As a result, a classifier that generalizes significantly better than the k-NN rule can be constructed.

In particular, our experiments demonstrate that the regularized normal density-based linear classifier (RLNC) built on dissimilarities mostly outperforms the k-NN rule based on the same representation set. The quadratic classifier (RQNC) performs even better, when the class covariance matrices can be estimated in a reliable way. This is observed in the bottom row of Figs. 3 and 4. Still, an open question remains how the regularization parameters should be chosen in an optimal way. Here, as a rule of thumb, relatively large fixed values are used for training sets only somewhat larger than the representation sets. In case of small representation sets, no regularization is needed.

In conclusion, our results encourage to explore meaningful dissimilarity information in new, advantageous ways, of which our proposal is an example. The use of other classifiers (e.g., the support vector classifier (Vapnik, 1998)) and the study of representation set selection is of interest for further research.

Acknowledgements

This work is partly supported by the Dutch Organization for Scientific Research (NWO).

References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach. Learning 6, 37–66.

Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 13 (1), 21–27.

Devijver, P.A., Kittler, J., 1982. Pattern Recognition: A Statistical Approach. Prentice-Hall, London.

Dubuisson, M.P., Jain, A.K., 1994. Modified Hausdorff distance for object matching. In: Proc. 12th Internat. Conf. on Pattern Recognition, Vol. 1, pp. 566–568.

Duda, R.O, Hart, P.E., Stork, D.G, 2001. Pattern Classification, second ed. Wiley, New York.

Duin, R.P.W., 2000. Classifiers in almost empty spaces. In: Proc. 15th Internat. Conf. on Pattern Recognition, Barcelona, Spain. Pattern Recognition and Neural Networks, Vol. 2. IEEE Computer Society Press, Los Alamitos, USA, pp. 1–7.

Duin, R.P.W., Pękalska, E., de Ridder, D., 1999. Relational discriminant analysis. Pattern Recognition Lett. 20 (11–13), 1175–1181.

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

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, New York.

Goldstone, R.L., 1999. Similarity. In: Wilson, R.A., Keil, F.C. (Eds.), MIT Encyclopedia of the Cognitive Sciences. MIT Press, Cambridge, MA, pp. 763–765.

Hart, P.E., 1968. The condensed nearest neighbor rule. IEEE Trans. Inf. Theory 14, 515–516.

Houtepen, J., 1994. Chromosome banding profiles: how many samples are necessary for a neural network classifier? Master thesis, Delft University of Technology, pp. 1–120.

Houtepen, J., Duin, R.P.W., 1994. Classification of chromosomes using coarsely sampled banding profiles. In: Abstr. 2nd Plenary Workshop of the ECA-AMCA, Galway, Ireland.

Houtepen, J., Vossepoel, A.M., 1994. Influence of spatial resolution on classification of banded chromosomes by means of neural nets and k-NN methods. In: Abstr. 2nd Aalborg Symp. on Chromosome Analysis: 'Segmentation and Classification', Aalborg, Denmark.

Pękalska, E., Duin, R.P.W., 2000. Classifiers for dissimilarity-based pattern recognition. In: Proc. 15th Internat. Conf. on Pattern Recognition, Barcelona, Spain. Pattern Recognition and Neural Networks, Vol. 2. IEEE Computer Society Press, Los Alamitos, USA, pp. 12–16.

Pękalska, E., Duin, R.P.W., 2001. Automatic pattern recognition by similarity representations. Electron. Lett. 37 (3), 159–160.

Ripley, B., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Wharton, C.M., Holyoak, K.J., Downing, P.E., Lange, T.E., Wickens, T.D., 1992. The story with reminding: memory retrieval is influenced by analogical similarity. In: Proc. 14th Annual Conf. of the Cognitive Science Society, Bloomington, IN, pp. 588–593.

Wilson, C.L., Garris, M.D., 1992. Handprinted character database 3. National Institute of Standards and Technology, Advanced Systems division.