Dissimilarity-based classification for vectorial representations

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

[1] Faculty of EEMCS, Delft University of Technology, The Netherlands
[2] School of Computer Science, University of Manchester,
e.m.pekalska@tudelft.nl, r.duin@ieee.org

Abstract

General dissimilarity-based learning approaches have been proposed for dissimilarity data sets [11, 10]. They arise in problems in which direct comparisons of objects are made, e.g. by computing pairwise distances between images, spectra, graphs or strings.

In this paper, we study under which circumstances such dissimilarity-based techniques can be used for deriving classifiers in feature vector spaces. We will show that such classifiers perform comparably or better than the nearest neighbor rule based either on the entire or condensed training set. Moreover, they can be beneficial for highly-overlapping classes and for non-normally distributed data sets, with categorical, mixed or otherwise difficult features.

1. Introduction

The nearest neighbor (NN) rule is a simple and widely applied technique thanks to its good asymptotic behavior in metric spaces [2]. It assigns an object to the class of its nearest neighbor as judged by the corresponding smallest distance. In practice, however, its performance may suffer from finite training sizes and/or noisy training examples. As a remedium, prototype optimization techniques are studied in vector spaces. They aim to make the 1-NN rule robust against noisy examples and to diminish its computational and storage requirements. Various algorithms have been proposed to determine small prototype sets, called also condensed sets, on which the 1-NN relies [1, 2, 3, 6, 12, 14].

Alternatively, a set of prototypes can be used to build a representation space in which general classifiers are trained. The main reason is to use such condensed sets efficiently (not only for finding the nearest neighbor) and to construct globally-aware classifiers. In this way, advantageous performance can be achieved as local distance-based information is combined with a more global classification technique. The simplest approach is to define a vector space in which each dimension represents a distance to a given prototype. Classifiers become then (non-)linear functions over a set of distances. This dissimilarity-based approach can be powerful, especially when prototypes are suitably optimized.

Support vector machines (SVMs) are mathematically elegant methods that can be cast out in such a framework. Numerous studies show that the Gaussian-SVM is often one of the best techniques for continuous variables [7]. The difficulty, however, arises for data with categorical or mixed variables as the Gaussian kernel becomes inappropriate and its hyperparameter σ is difficult to optimize.

In this paper, we will show that simple dissimilarity-based techniques, alternative to the 1-NN rule and the SVM, can work well for feature-based representations, especially if they consist of categorical or mixed types.

2. Prototype generation and selection

Dissimilarity spaces. Assume a training set T of N objects and a representation set $R = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ of n prototypes. Given a dissimilarity measure d (not necessarily metric), a new representation is based on the proximities to R. Every object $\boldsymbol{x} \in T$ is described by a vector of distances computed between $\boldsymbol{x}$ and the prototypes from R, i.e. as $D(\boldsymbol{x}, R) = [d(\boldsymbol{x}, \boldsymbol{p}_1), d(\boldsymbol{x}, \boldsymbol{p}_2), \ldots, d(\boldsymbol{x}, \boldsymbol{p}_n)]^T$. Here, $D(\cdot, R)$ is interpreted as a data-dependent mapping $D(\cdot, R) : X \to \mathbb{R}^n$ from a vector space X to a *dissimilarity space* defined by R. This is a vector space, equipped with the standard inner product and norm, in which each dimension $D(\cdot, \boldsymbol{p}_i)$ describes the dissimilarity to a prototype. The training data set becomes an $N \times n$ distance matrix $D(T, R)$. Since it is a vectorial representation, any traditional classifier built in vector spaces can be used. See [11, 10, 9] for more details.

Linear and quadratic functions in a dissimilarity space are weighted linear and quadratic combinations of the distances $d(\boldsymbol{x}, \boldsymbol{p}_i)$ between the object $\boldsymbol{x}$ and the prototypes $\boldsymbol{p}_i$. The advantage is that the weights are optimized over the entire training set T, which leads to globally-aware classifiers. Such normal density based classifiers tend to perform well in dissimilarity spaces [11, 10]. This holds for summation-based distances computed over a number of component differences with similar variances. Such distances tend to be approximately normally distributed thanks to the central limit theorem (if few variances are dominant, they will approximate the χ^2 distribution) [9]. Examples are Euclidean or city block distances computed over normalized features.

Prototype optimization. Condensed sets are determined to reduce the computational effort, while preserving a good performance of the 1-NN rule. They are optimized by adaptive [1, 8] and selective schemes [3, 6, 12, 14]. In case of small condensed sets or non-representative training

0-7695-2521-0/06/$20.00 (c) 2006 IEEE

Table 1. Data used in experiments. * denotes that non-informative (zero) features are removed.

Data	# objects	# features	# classes	Class sizes	Variable type	Scaling	Distance
Australian	690	14	2	383/307	Mixed	Domain	City block
Biomed	194	5	2	127/67	Mixed	Domain	City block
Breast (Wisconsin)	683	9	2	444/239	Categorical	Domain	City block
Diabetes	768	8	2	500/268	Mixed	Domain	City block
Heart	297	13	2	160/137	Mixed	Domain	City block
Ecoli	272	5^*	3	143/77/52	Continuous	Standardized	Euclidean
Glass	214	9	4	70/76/17/51	Continuous	Standardized	Euclidean
Ionosphere	351	32^*	2	225/126	Continuous	Standardized	Euclidean
Liver	345	6	2	145/200	Cont. Integer-valued	Standardized	Euclidean
Musk	476	166	2	199/277	Cont. Integer-valued	No scaling	Euclidean
Sonar	208	60	2	97/111	Continuous	No scaling	Euclidean
Wine	178	13	3	59/71/48	Continuous	Standardized	Euclidean

data, a better generalization can be achieved by a classifier built in a dissimilarity space than by the 1-NN rule, even if the condensed set is optimized for the latter.

In this paper,condensed sets are the representation sets R used to construct a dissimilarity space. Representation sets can be found by numerous approaches [10]. Here, we focus on five different ways. *EMgen* denotes a set of prototypes generated in the original feature vector space. These are cluster means (merged from the cluster members) of the clusters determined by an EM algorithm [4]. Clusters are modeled by Gaussian distributions with identical diagonal covariance matrices. Some noise is added to the data to prevent degenerated solutions for categorical variables.

Other methods are selective, i.e. they select prototypes from the training set T by working on dissimilarity representations $D(T,T)$. *FSel* describes a set of prototypes chosen in a dissimilarity space by a forward feature selection with the criterion based on the Mahalanobis distance (remember that 'features' correspond to objects in a dissimilarity space). *EdCon* stands for the traditional edited and condensed set optimized for the performance of the 1-NN rule [3]. *LP* refers to a prototype set optimized for the performance of a sparse linear programming (LP) machine, μ-LPM [5, 10], where μ is set to the value of the 10-fold crossvalidation 3-NN error. *LPauc* is a prototype set optimized for the performance of a sparse linear programming machine, defined by maximizing the area under the ROC curve as proposed in [13]. For the *EMgen* and *FSel* methods, the cardinality of R is set a priori such that either $k_i = \lceil \sqrt{|\omega_i|} \rceil$ prototypes are optimized per class ω_i or $\sum_i k_i$ prototypes are defined in total. In other cases, the cardinality of R is automatically determined by the used methods.

3. Experiments and results

Our goal is to illustrate the potential of dissimilarity-based linear and quadratic classifiers defined for vectorial representations. A more detailed study can be found in [8], in which two adaptive and two selective schemes (of which three are different than the ones presented here) were empirically analyzed for prototype optimization. The main conclusion was that a dissimilarity-based Fisher linear discriminant combined with prototype generation schemes (run in normalized feature spaces) offers the best trade-off between the computational effort and classification performance.

Ten data sets are here considered from the UCI Repository, http://www.ics.uci.edu/ mlearn/MLRepository.html. They describe problems with categorical, continuous and mixed features. For the sake of simplicity we use either the city block distance or the Euclidean distance. Note that the distances are by no means restricted to such metrics, any can be used in general. City block distances are employed for data with categorical or mixed features. Euclidean distances are used otherwise. The details are shown in Table 1.

In our experiments, each data set is split into a training set T and a test set S in the ratio of 75%:25%. If applicable, the training and test sets are first scaled and then the Euclidean or city block distance representations are defined. Five different representation sets are used, as described in Section 2. Classifiers are then trained in dissimilarity spaces $D(T,R)$ and tested on $D(S,R)$. Additionally, the 1-NN and k-NN rules are applied; k is optimized by a leave-one-out (LOO) error. Three classifiers are used in dissimilarity spaces. These are the NLC, a normal density based linear classifier, the NQC, a normal density based quadratic classifier and the LOGC, logistic linear classifier [4]. The NQC is regularized by a small parameter $\lambda = 10^{-4}$ [10]. Additionally, also *1NNd* - the 1-NN is directly applied to $D(S,R)$ and *1NN* - the 1-NN is used in a dissimilarity space (the Euclidean distances are computed in this space). Prior probabilities are estimated by class frequencies. The entire procedure is repeated 30 times and the classification results are averaged out. As an indication, three classifiers are used in the original vector spaces *OrigFS*: the NQC, NaiveBC (naive Bayes classifier) and the Gaussian-SVM with the hyperparameter σ optimized in a 10-fold crossvalidation.

The results are presented in Fig. 1. Non-listed classifiers perform outside the top plotted range. Our basic observation is that linear and quadratic classifiers in dissimilarity spaces outperform the direct NN rule, *1NNd*, when based on the same representation sets. This also holds for the condensed *EdCon*-sets, which are optimized for the NN performance. Secondly, either the NLC or the NQC trained in dissimilarity spaces outperform the direct 1-NN rule and perform similarly or better than the direct k-NN rule, the former based on the complete training set. Thirdly, in case of

discriminative continuous features, the *EMgen*-prototypes lead to somewhat better results than the *Fsel*-prototypes. Our rule of thumb of fixing the cardinality for these sets works well when the number of features is not too large; it is however insufficient for high-dimensional data, such as Musk and Sonar. Next, the condensed sets chosen by the *EdCon*, tend to work worse for the city block distances on mixed data than in other situations. On the contrary, they are good for the high-dimensional Musk and Sonar data. Next, *LP* can lead to bad results for non-continuous and difficult data, even to no-sparse solutions as for the Australian, Diabetes and Glass data. In contrast to the *LPauc*, it works well for the high-dimensional Musk and Sonar data. Finally, the *LPauc* is well suited for highly-overlapping classes as present in the Diabetes, Heart, Glass or Liver data. The number of prototypes determined by the *LPauc* is much smaller than by the *LP*. Still, the prototype sets found automatically by the *EdCon*, *LP* and *LPauc* methods have sizes up to a few times our fixed size of the *EMgen* and the *FSel*.

In general, the dissimilarity-based NLC and NQC perform better than the LOGC, which supports our claim that dissimilarity data are approximately normally distributed. The LOGC gives also large standard deviations. When the NaiveBC works the best for the original categorical/mixed data (or clearly non-normal data), then the NLC performs the best in dissimilarity spaces. The Gaussian-SVM built in the original vector space often works the best for continuous data (Ionosphere, Musk or Sonar), provided that the class overlap is not too high. It may, however, need a very large amount of support vectors. In more difficult cases of multi-class problems, variables of mixed types or a high class overlap, the dissimilarity-based NLC and NQC are to be preferred.

4. Discussion and conclusions

Prototype sets are usually studied to maximize the accuracy of the 1-NN rule. Here, an alternative use of prototypes is proposed for dissimilarity-based classifiers derived from the original vectorial representations; see also [8]. Given a small prototype set (even when optimized for the performance of the NN method), an entire training set can be used to increase the accuracy of a classifier in the dissimilarity space. Such dissimilarity-based classifiers tend to be more accurate than the 1-NN rule since they are globally-aware and the parameter values are optimized in a better way. The computational costs of applying both the NN rule and linear dissimilarity-based classifiers are similar.

This paper explores several aspects concerning the possible advantage of dissimilarity spaces over the original feature spaces, with the focus on the nature of the feature measurements (categorical/mixed vs continuous) and data with high class overlap. Two main results are presented. First, normal density based linear (NLC) and quadratic (NQC) classifiers built in dissimilarity spaces are often more beneficial than the 1-NN and k-NN rule directly applied. (Note that all these classifiers are non-linear functions in the original feature space. So, they should be applied when they are indeed needed.) Moreover, they perform similarly or better than the best NN rule over the entire training set. This is in agreement with our earlier findings concerning general dissimilarity data (not derived in vector spaces) [10, 9].

Secondly, the dissimilarity-based NLC and the NQC can be recommended for data which consist of categorical or mixed variables or with a potentially high class overlap. These are the cases in which the Gaussian-SVM and other classifiers built in the original vector space tend to loose. In general, the Gaussian-SVM is the largest margin (hence optimal) classifier for non-overlapping classes in Euclidean feature spaces. Its performance deteriorates, when a high overlap occurs. In such cases, linear and quadratic dissimilarity-based classifiers are advantageous. Depending on the problem characteristics, a particular prototype optimization technique can be suggested, as described in Section 3. Adaptive techniques are especially of interest since they generate new prototypes (e.g. by merging a number of suitable training examples) and allow one to control their size. A more thorough study is left for future research.

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

References

[1] C. Chang. Finding prototypes for nearest neighbor classifiers. *IEEE Tran. on Comp.*, 23:1179–1184, 1974.

[2] B. Dasarthy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[3] P. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice/Hall, 1982.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2nd edition, 2001.

[5] T. Graepel, R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K.-R. Müller, K. Obermayer, and R. Williamson. Classification on proximity data with LP-machines. In *Int. Conference on Artificial Neural Networks*, pages 304–309, 1999.

[6] P. Hart. The condensed nearest neighbor rule. *IEEE Trans. on Information Theory*, 14:515–516, 1968.

[7] Website. `http://www.kernel-machines.org/`.

[8] M. Lozano, J. Sotoca, J. Sánchez, F. Pla, E. Pękalska, and R. Duin. Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, accepted, 2006.

[9] E. Pękalska and R. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, Singapore, 2005.

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

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

[12] J. Sánchez, F. Pla, and F. Ferri. Improving the k-NCN classification rule through heuristic modifications. *Pattern Recognition Letters*, 19:1165–1170, 1998.

[13] D. Tax and C. Veenman. Tuning the hyperparameter of an AUC-optimized classifier. In *Belgium-Netherlands Conference on Artificial Intelligence*, pages 224–231, 2005.

[14] R. Wilson and T. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

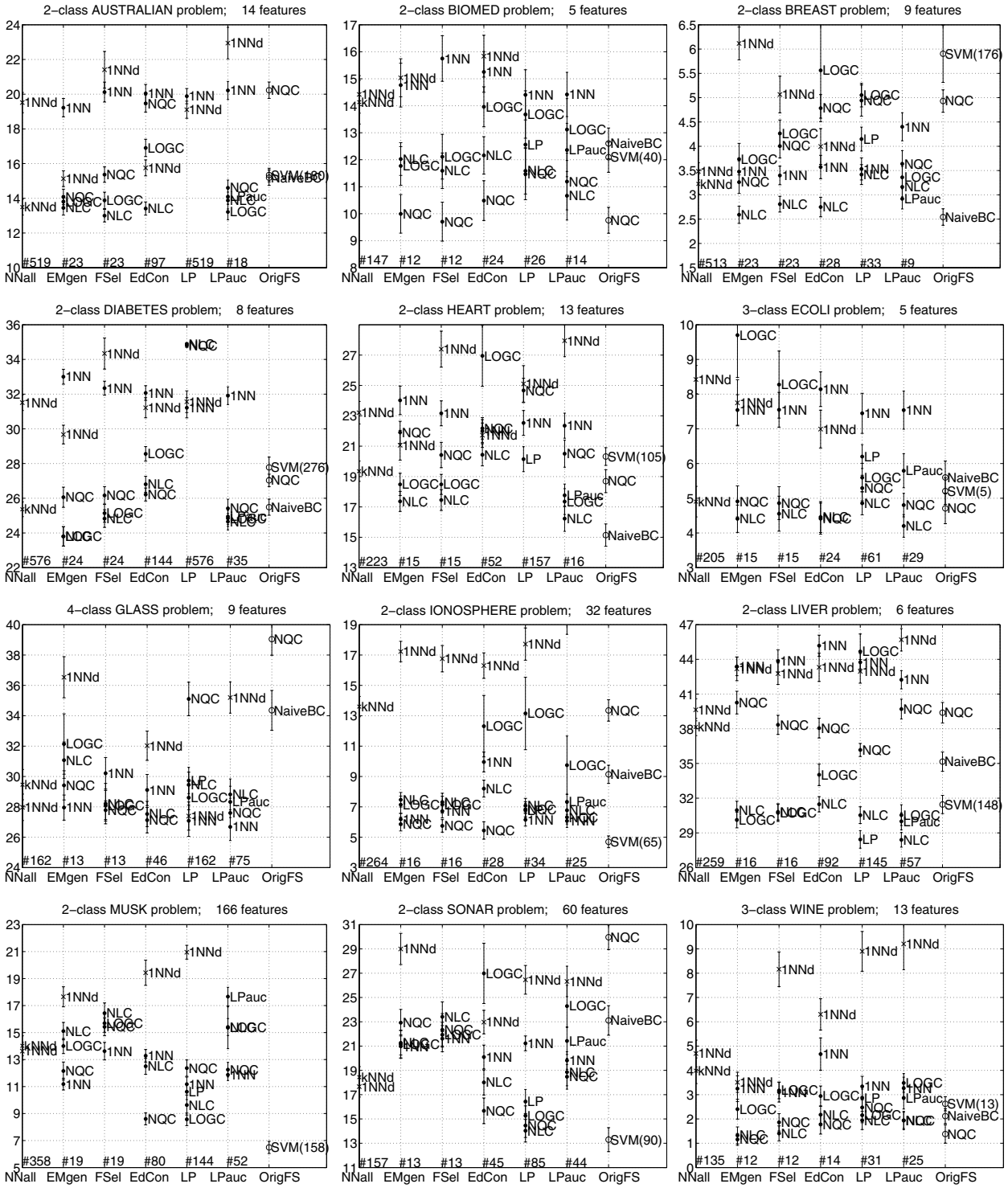

Figure 1. Averaged (over 30 runs) classification errors (in %) with their standard deviations. Black dots denote classifiers in dissimilarity spaces, circles denote classifiers in original features spaces and 'x' denotes the direct NN rule. The *NNall* refers to the performance of the 1-NN and k-NN rules directly applied to the entire training set. Representation sets are optimized in five ways: *EMgen*, *FSel*, *EdCon*, *LP* and *LPauc*; see text for details. The cardinalities of T (for *NNall*) and R (for other cases) are printed on the horizontal axis above the methods. *OrigFS* refers to the original feature space. The average number of support vectors is shown in the brackets behind the SVM. The LP and LPauc on the error axis denote the errors of the optimized LP machines.