The characterization of classification problems by classifier disagreements

Robert P.W. Duin, Elżbieta Pękalska, David M.J. Tax
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology, The Netherlands
{r.p.w.duin,e.pekalska,d.m.j.tax}@ewi.tudelft.nl

Abstract

In this paper we try to characterize a set of classification problems. For this, we use the disagreement between a set of standard classifiers. The disagreement patterns do not only point towards different types of classification problems, but also indicate the novelty and the usefulness of a classifier with respect to a set of classification problems and classifiers. Some experiments show when known classification problems become unknown after changing their feature size or their training set size.

1. Introduction

In designing effective procedures for building pattern recognition systems it is essential to have some knowledge on the set of classification problems one may encounter. As argued by Wolpert [11] , it is not expected to reach any generalization in training classifiers if all possible classification problems are equally probable. The study of Vapnik on the complexity of classifiers [10] showed that very large training sets are needed to guarantee a reasonable performance if one considers all possible relabelings of a given dataset as equally probable.

Such studies show that in building classifiers, we should already have some prior knowledge of the problems that have to be solved by these tools. A very common, often implicitly made assumption on the set of problems is that they obey the so-called 'compactness hypothesis' [1]: each classification problem is the result of a representation of real world objects, such that similar objects have similar representations [3]. The above mentioned studies by Wolpert and Vapnik indicate the necessity of such an assumption.

A general approach to the characterization of classification problems is difficult. Still it is important, as it may be a first step to describe the set of problems encountered in practice. Studies by Ho et al., e.g. [5] focus on the problem (data) complexity as a natural possible attribute. It appears that this is still ill-defined and there are numerous ways to measure the problem complexity.

In this paper our point of view is actually that the tools used to solve a classification problem provide its natural characterization. The performances of these tools may give a first indication how to solve the problem, as they tell whether the chosen classifiers are appropriate. However, it may be possible that specific, advanced tools are needed and that we are not in the position of trying them, since this would require an exhaustive search over all possible solutions to tackle the problem. So, we need some problem characteristics to assist us in searching for the appropriate tools. In order to define such characteristics we investigate here the differences in classification results (and not just in absolute performance) between individual classifiers, measured by their disagreements. A classification problem is represented by the disagreement pattern between a set of standard classifiers.

Once problems are characterized, they can be compared. We will illustrate how this may be done. If such a technique is fully developed we might be able to judge automatically whether a new classification problem is similar to a standard problem (e.g. related to a specific approach). This will give us a first indication on how it may be tackled. It may also be concluded that the problem does not fit to the standard set and that a novel procedure for its investigation is desirable.

We will present a first analysis in the direction sketched above. In section 2, a possible set of base classifiers will be presented. How their disagreement may be used to represent a problem is discussed in section 3. A set of classification problems is defined in section 4. This set has to be enlarged and purified. How this may be analyzed is presented in section 5. Some additional experiments are discussed in section 6.

2. The set of classifiers

In selection of a basic set of classifiers for arbitrary problems it should be realized that for every classifier a problem may be defined for which it is the best. This is the consequence of the fact that a classifier is based on some model or data assumptions. If they are fulfilled and all other assumptions made by other classifiers do not apply then this particular classifier will yield the best performance. So all classifiers are admissible. As a consequence we cannot neglect any classifier as it will always be better than all other ones for some problem.

What may happen, however, is that some classifiers are not of significant importance for the set of problems of interest. Here we encounter an essential difficulty in the analysis. We aim to characterize problems by classifiers, but we can only decide about the classifiers to be used once we have analyzed the set of problems of interest and we need the classifiers to do this.

In order to bootstrap this vicious circle, we may start with an initial, small set of classifiers for which we are convinced that they show considerable differences over interesting problems. Other classifiers can later be added when their contribution appears to be of help, i.e. if they show differences between problems that have not been distinguished before.

As an initial classifier set we have chosen the following 13 classifiers, see also [6]:

- NMC: the nearest mean classifier,
- Fisher: Fisher's linear discriminant,
- UNormalBC: the Bayes classifier assuming uncorrelated normal densities,
- NormalBC: the Bayes classifiers assuming arbitrary normal densities,
- NaiveBC: the naive Bayes classifier based on 10-bin histograms per feature,
- ParzenC: the Parzen classifier, Using a leave-one-out optimization of the smoothing parameter.
- 1-NN: the one nearest neighbor rule,
- k-NN: the k-nearest neighbor rule. The value of k is optimised for the leave-one-out classification error,
- LogC: the logistic classifier,
- SVC-1: the support vector classifier using a linear kernel, and with regularization parameter c = 1,
- SVC-2: the support vector classifier using a quadratic kernel, and with regularization parameter c = 1
- LM-NeurC: a neural net with one hidden layer with 5 neurons, trained by the Levenberg-Marquardt rule,
- CART: A CART like decision tree [2], maximizing the purity and using early pruning [9],

We used PRTools4 [4] for experiments.

3. The disagreement between classifiers

Different classification rules usually give rise to different classifiers. One way to measure the difference between two classifiers C_1 and C_2 trained on a classification problem P_j ($j = 1 ,..., N$; N is the size of the set of problems) is the disagreement $d_j(C_1,C_2)$: the probability that an arbitrary object $\boldsymbol{x} \in P_j$ gets different labels assigned by the classifiers C_1 and C_2 trained on the problem P_j:

$$d_j(C_1,C_2) = \mathrm{Prob}(C_1(\boldsymbol{x}) \sim= C_2(\boldsymbol{x}) \mid \boldsymbol{x} \in P_j) \qquad (1)$$

$C_i(\boldsymbol{x})$ returns the label for object $\boldsymbol{x}$ according to classifier C_i. M classifiers constitute an M x M disagreement matrix D_j^c for problem P_j, with elements $D_j^c(m,n)$ = $d_j(C_m,C_n)$. In [7] it is discussed how such a dissimilarity matrix can be visualized by an embedding it into a 2D Euclidean space. The so-called Classifier Projection Space (CPS) shows the individual classifiers as points such that the realized 2D Euclidean distances approximate the actual disagreements, see fig 1.

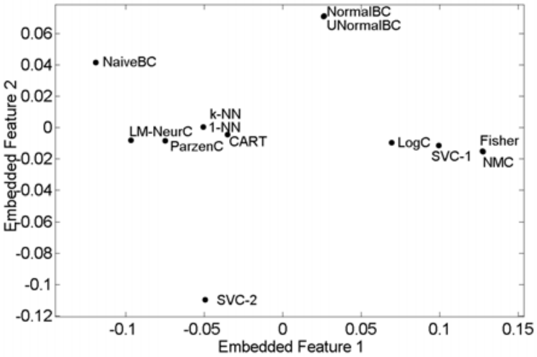

Fig. 1. In the CPS the classifiers are represented such that the visual distances optimally preserve the disagreements. This is the result for the Highleyman problem with 10+10 objects (see section 4)

The disagreement measure is metric, yet not Euclidean. As a consequence, there does not exist a Euclidean space that perfectly embeds a given disagreement matrix. The embedding shown in fig. 1, however, still explains about 75% of the squared classifier disagreements by the squared Euclidean distances. In this figure some classifier and problem characteristics can be recognized: the similarities between the linear classifiers NMC, Fisher, SVC-1 and LogC, as well as, between the nonparametric procedures 1-NN, k-NN and ParzenC. As the true covariance matrices in the Highleyman classification problem are uncorrelated, the normal densities Bayes classifiers assuming equal or different covariance matrices, UNormalBC and NormalBC, are very close. The other quadratic classifier, SVC-2, is remote from them. Also the neural net, LM-NeurC and the Naive Bayes Classifier, NaiveBC, show their own, different characteristics.

4. The set of problems

We now define a set of 18 2-class problems that will be related by their disagreement matrices, see table 1. Some of them ('*') are artificial, available in PRTools

[4], others are taken from the UCI repository [12]. The Banana dataset shows two 2D banana shaped classes. The NCorrX datasets are based on highly correlated X-dimensional normal distributions with equal covariance matrices. Digit38 consists of the digits 3 and 8 of the multi-feature set (mfeat) dataset in the UCI repository. We used the Karhunen-Loève moments (Digit38-kar) and the Zernike moments (Digit38-zer). In table 1 the dimensionalities and sample sizes are listed. The columns R and B will be explained later.

Table 1 The set of datasets

Dataset name	#features	#objects	R	B
Highleyman-20*	2	10 + 10		
Highleyman-100*	2	50 + 50		
Banana-20*	2	10 + 10		
Banana-100*	2	50 + 50	X	
NCorr2-20*	2	10 + 10		
NCorr2-100*	2	50 + 50		
NCorr5-100*	5	50 + 50		
NCorr20-100*	20	50 + 50		
Spirals*	2	97 + 97	X	X
Sonar	60	97+ 111		
Biomed	5	127 + 67		
Diabetes	8	500 + 268	X	
Auto-mpg	6	229 + 169		
Ionosphere	34	225 + 126		
Liver	6	145 + 200		X
Breast	9	444 + 239		
Digit38-kar	64	200 + 200		X
Digit38-zer	47	200 + 200	X	X

For each of the problems P_j (j=1,...,18) the disagreement matrix D^c_j was estimated. For the artificial datasets (j=1,...,8) a test set of 1000 objects per class was used. For the real world problems 10-fold cross-validation was applied. Note that by both systems, a random component in the disagreement estimation is introduced: the estimation is not a unique function of the dataset, but depends on the seed of a random generator.

5. Problem characterization by classifiers

To compare the full disagreement matrices, a dissimilarity measure between the problems is needed. As they all have the same size (M x M, with M the number of classifiers) and their values are probability estimates on the [0,1] interval, no normalization is required. To compare the problems P_r and P_s we rather arbitrarily used

$$D^p(r,s) = \sum_{m,n} \left| D^c_r(m,n) - D^c_s(m,n) \right| \qquad (2)$$

i.e. the sum of all absolute differences between the classifier disagreements, to define the problem dissimilarity matrix D^p. Similar to the CPS a Problem Projection Space (PPS) can be defined. Fig. 2 shows a scatter plot of the problems of the first two dimensions of the PPS.

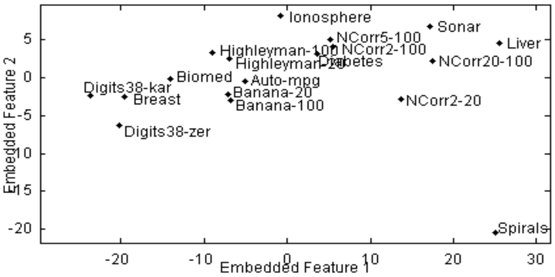

Fig. 2. In the Problem Projection Space (PPS) the datasets are represented such that their visual distances optimally agree with their dissimilarities.

We will now face the question how it can be detected whether a new problem c fits in a set of problems represented by D^p. A possible way to answer this is to use novelty detectors or one-class classifiers. To that end, a compact description of D^p has to be derived, such that for a new problem P_k, represented by its disagreement matrix D^c_k it can be established whether it fits to this description or or it does not.

Since D^p, like D^c_k, is based on non-Euclidean dissimilarity measures, embedding of the set of problems in an Euclidean space will cause difficulties, similar to the embedding discussed in section 3. As a consequence, density estimators or support vector machines using Mercer kernels cannot be used, unless the dissimilarity data is transformed to be Euclidean. Here we want to study the use of the original D^p. In [8] a one-class classifier was proposed for general dissimilarity data based on a linear programming technique which we will apply here to define a compact description of D^p.

Let $\mathbf{d}^p_k = D^p(k,:)$ be the vector of dissimilarities of a problem P_k with disagreement matrix D^c_k to all known problems in our standard set. P_k can be inside or outside this set. The classifier $W(\mathbf{d}^p_k)$ is now defined as

$$W(\mathbf{d}^p_k) = \mathbf{w} \bullet \mathbf{d}^p_k = \sum_j^N w_j d^p(k,j) - w_0 , \qquad (3)$$

if $W(\mathbf{d}^p_k) \le 0$, $\mathbf{d}^p_k$ is accepted, otherwise rejected w.r.t. the class of objects (problems) defined by D^p. The classifier $W(\mathbf{d}^p_k)$ is optimized by a linear programming procedure in which w_0 is maximized subject to

$$\forall i \left(\sum_j^N w_j d^p(i,j) - w_0 < 0 \right), \sum_j^N w_j = 1, \forall j (w_j \ge 0) \quad (4)$$

This procedure results in a number of zero weights $w_j = 0$. The corresponding objects (in our case problems) are therefore not needed for the generalization.

We may say that the 'problem space' is defined by just the problems for which $w_j > 0$. We call them the representation objects. Other, or the same objects (problems) are on the boundary of this description. They correspond to objects that would be rejected in a leave-one-out approach: they do not belong to the class defined by the other n-1 objects.

In our example, the representation set of problems appears to consist of four datasets, indicated in column R of table 1. This implies that the other datasets are not needed for building the classifier. New classification problems should be compared by (3) to just these ones.

The four boundary cases we found (that were rejected by the leave-one-out test), are indicated in table 1 in column B. They are in one way or another the most extreme. Note that Spirals and Digits38-zer belong to both sets. They are in fact rather atypical: e.g. the Spiral problem is a structured, noise free 2D dataset and the Zernike moments have very different scales.

6. Experiments

The one-class classifier found above was also used for classifying a series of new or modified problems. For the artificial datasets we tried various sizes of the training set or a range of dimensionalities. This was repeated 10 times to test the stability. An example is given in fig. 3 showing the frequency of accepting the NCorrX-20 dataset to the set of standard problems for various feature sizes. This shows a deteriorating acceptance as a 'standard problem' for feature sizes larger than 5.

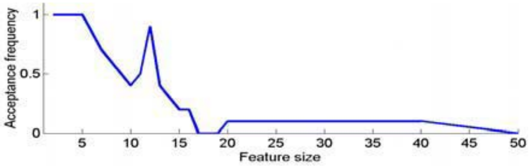

Fig. 3. The frequency in 10 experiments that the NCorrX-20 dataset is accepted as a 'standard problem' for various feature sizes.

In a second series of experiments we modified some of the real world problems by decreasing their dataset sizes. In table 2 we show for three problems whether they are still accepted as one of the 'standard' problems or are rejected. Sampling fractions are listed above the columns. The 'Liver' dataset, which is a boundary case, is directly rejected, while 'Sonar' is accepted to 50% of its size and 'Biomed' even to 35%.

Table 2 Classification of sampled datasets

problem	1	0.9	0.8	0.65	0.5	0.35
Sonar	accept	accept	accept	accept	accept	reject
Biomed	accept	accept	accept	accept	accept	accept
Liver	accept	reject	reject	reject	reject	reject

7. Discussion

We showed that it is possible to use a standard set of classification problems for the construction of a rule that decides about the similarity of new problems to the existing ones. This is based on the disagreements between a set of classifiers. They thereby influence to what extent problems can be distinguished. It has to be investigated whether other, possibly more advanced classifiers, can be used for that purpose.

If it is possible to group classification problems in a consistent way, this may be of a great help to select the appropriate tools for solving new problems. In this paper we just sketch this perspective. We admit that still much has to be investigated about the proposed procedure, e.g. its consistency and its stability, and also about the selection of the tools for new problems. Nevertheless, we judge the preliminary results as encouraging.

Acknowledgement

This work was supported by the Dutch Organization for Scientific Research (NWO)

8. References

[1] A.G. Arkedev and E.M. Braverman, *Computers and Pattern Recognition*, Thompson, Washington, D.C., 1966.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[3] R.P.W. Duin, Compactness and Complexity of Pattern Recognition Problems, in: C. Perneel (ed.), Proc. *Int. Symposium on Pattern Recognition "In Memoriam Pierre Devijver"*, Royal Military Ac., Brussels, 1999, 124-128.

[4] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, *PRTools4, a Matlab toolbox for Pattern Recognition* [http://prtools.org], Delft Univ. of Techn., 2004.

[5] T.K. Ho, M. Basu, Complexity Measures of Supervised Classification Problems, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24 , 3, 2002, 289-300.

[6] A.K. Jain, R.P.W. Duin, and J. Mao, Statistical Pattern Recognition: A Review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000, 4-37.

[7] E. Pekalska and R.P.W. Duin, A discussion on the classifier projection space for classifier combining, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems*, Lecture Notes in Comp. Sc., vol. 2364, Springer, 2002, 137-148.

[8] E. Pekalska, D.M.J. Tax, and R.P.W. Duin, One-Class LP Classifiers for Dissimilarity Representations, in: S. Becker, S. Thrun and K. Obermayer (eds.), *Adv. in Neural Inf. Processing Systems*, vol. 15, MIT Press, 2003, 761-768.

[9] J.R. Quinlan, *Simplifying Decision Trees*, Int. J. Man - Machine Studies, vol. 27, 1987, pp. 221-234.

[10] V.N. Vapnik, *Estimation of Dependencies Based on Empirical Data*, Springer, 1982.

[11] D.H. Wolpert, *The Mathematics of Generalization*, Addison-Wesley, London, 1995.

[12] C.L. Blake, and C.J. Merz, *UCI Repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: Univ. of California. 1998.