A Discussion on the Classifier Projection Space for Classifier Combining

Elżbieta Pękalska, Robert P.W. Duin, and Marina Skurichina

Pattern Recognition Group, Department of Applied Physics,
Faculty of Applied Sciences, Delft University of Technology,
Lorentzweg 1, 2628 CJ Delft, The Netherlands
{ela,duin,marina}@ph.tn.tudelft.nl

Abstract In classifier combining, one tries to fuse the information that is given by a set of base classifiers. In such a process, one of the difficulties is how to deal with the variability between classifiers. Although various measures and many combining rules have been suggested in the past, the problem of constructing optimal combiners is still heavily studied.
In this paper, we discuss and illustrate the possibilities of classifier embedding in order to analyse the variability of base classifiers, as well as their combining rules. Thereby, a space is constructed in which classifiers can be represented as points. Such a space of a low dimensionality is a Classifier Projection Space (CPS). In the first instance, it is used to design a visual tool that gives more insight into the differences of various combining techniques. This is illustrated by some examples. In the end, we discuss how the CPS may also be used as a basis for constructing new combining rules.

1 Introduction

When a pattern classification problem is too complex to be solved by training a single (advanced) classifier, the problem may be divided into subproblems. They can be solved one per time by training simpler base classifiers on subsets or variations of the problem. In the next stage, these base classifiers are combined. Many strategies are possible for creating subproblems as well as for constructing combiners [11]. Base classifiers are different by nature since they deal with different subproblems or operate on different variations of the original problem. It is not useful to store and use sets of classifiers that perform almost identically. If they differ somewhat, as a result of estimation errors, averaging their outputs may be worthwhile. If they differ considerably, e.g. by approaching the problem in independent ways, the product of their estimated posterior probabilities may be a good rule [8]. Other combining rules, like minimum, median or majority voting behave in a similar way. Having significantly different base classifiers in a collection is important since this gives raise to essentially different solutions. The concept of diversity is, thereby, crucial [9]. There are various ways to describe the diversity, usually producing a single number attributed to the whole collection of base classifiers. Later in this paper, we will use it differently.

F. Roli and J. Kittler (Eds.): MCS 2002, LNCS 2364, pp. 137–148, 2002.

A basic problem in classifier combining is the relation between base classifiers. If some are, by accident, identical, and others are very different, what is then the rationale of choosing particular combining rules? The outcomes of these rules depend on the distribution of base classifiers, but there is often no ground for the existence of such a distribution. Other combining rules, like maximum or minimum, are sensitive to outliers. Moreover, most fixed rules heavily rely on well established outputs, in particular, their suitable scaling.

One way to solve the above drawbacks is to use a trained output combiner. If the combiner is trained on a larger set, most of the above problems are overcome. Nevertheless, many architectures remain possible with different output combiners. A disadvantage of a trained combiner is, however, that it treats the outputs of base classifiers as features. Their original nature of distances or posterior probabilities is not preserved. Consequently, trained output combiners need a sufficiently large set of training examples to compensate this loss of information.

What we are looking for is a method of combining base classifiers that is not sensitive to their defects resulting from the way their collection is constituted. We want to use the fact that we deal with classifiers and not with arbitrary functions of the original features. To achieve that, we propose to study the collection of classifier pairwise differences, an $n \times n$ dissimilarity matrix D, before combining them into an output combiner. The dissimilarity value may be based on one of the diversity measures [9], like the disagreement [7]. Such a matrix D can be then embedded into a space $\mathcal{R}^k$, $k < n$, in a (non-)linear way. This means that classifiers are represented as a set of n points in $\mathcal{R}^k$ such that their Euclidean distances are identical to the original dissimilarities, given by D. It is also possible to perform an approximate embedding, where a space of a lower, fixed dimensionality is determined for an optimal approximation of D. We call this a Classifier Projection Space (CPS).

If the CPS is 2-dimensional, it can be visualised. Then, the collection of base classifiers, various combiners and, if desired, also other classifiers can be presented in a single 2D plot. The exact way of visualisation is explained in section 2. In sections 3 and 4, some examples are given for various sets of base classifiers constructed on real data. We will discuss how this illustrates some of the characteristics of the various techniques to generate both base classifiers and some combiners. We see it as a challenge to make use of the CPS for building a new type of combining classifier. This will be a trained output combiner, as it uses the training set. The construction of the CPS will be based on classifiers themselves and not on arbitrary feature functions. The possibilities will be discussed in the final section.

2 Construction of the Classifier Projection Space

Let us assume n classifiers trained on a dataset. For each pair of classifiers, their diversity value is determined, by using an evaluation set. This gives an $n \times n$ symmetric diversity matrix D. To take into account the original characteristics of the base classifier outputs, a suitable diversity measure should be chosen

to establish the basic difference between classifiers. A spatial representation of classifiers can be found by a projection to a CPS such that the points correspond to classifiers and the diversities, reflected by Euclidean distances between the points, are preserved as well as possible. Studying the relations between classifiers in the CPS allows us for gaining a better understanding than by using the mean diversity only. The latter might be irrelevant e.g. for an ensemble consisting of both similar and diverse classifiers, where their contributions might average out.

The joint output of two classifiers, $\mathcal{C}_i$ and $\mathcal{C}_j$ can be related by counting the number of occurrences of correct (1) or wrong (0) classification, e.g. a is the number of correct classifications for both $\mathcal{C}_i$ and $\mathcal{C}_j$ in Fig. 1. This requires the knowledge of correct labels (e.g. not available for a test set), which can be avoided when the outputs of classifiers are compared and (1) describes the agreement between them. Many known (dis)similarity measures can be used; see e.g. [4, 9]. Here, we will consider a simple diversity measure, the disagreement [7], which for $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as (see Fig. 1)

	$\mathcal{C}_i(1)$	$\mathcal{C}_i(0)$
$\mathcal{C}_j[1]$	a	b
$\mathcal{C}_j[0]$	c	d

Fig. 1. $\mathcal{C}_i$ vs. $\mathcal{C}_j$.

$$D_{i,j} = \frac{b+c}{a+b+c+d}, \qquad i,j = 1,\ldots,n \tag{1}$$

Given the complete diversity matrix D, reflecting the relations between classifiers, the CPS can be found by a (non-)linear projection, a variant of Multidimensional Scaling (MDS) [4]. Such a mapping is insensitive to redundant classifiers and perhaps also to outlier classifiers that do not have much support from the data. It is, however, sensitive to noise in the estimates of the dissimilarities. Below, we explain how from a dissimilarity matrix D one obtains a spatial representation.

2.1 Classical Scaling and Generalization to New Objects

Given an $n \times n$ Euclidean distance matrix D, between the elements of a set T, a configuration X of n points in $\mathcal{R}^m$ $(m \leq n)$ can be found, up to rotation and translation, such that the distances are preserved exactly. The process of such a linear mapping is called embedding and it is known as *classical scaling* [4]. Without loss of generality, the mapping is constructed such that the origin coincides with the mean. X is determined, based on the relation between distances and inner products. The matrix of inner products B can be expressed only by using the square distances $D^{(2)}$ [4, 14] as $B = -\frac{1}{2} J D^{(2)} J$, where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \in \mathcal{R}^{n\times n}$ (I is the identity matrix) projects the data such that X has a zero mean. By the eigendecomposition of $B = XX^T$, one obtains $B = Q\Lambda Q^T$, where Λ is a diagonal matrix of decreasing positive eigenvalues, followed by zeros. Q is the matrix of the corresponding eigenvectors. Then, X can be represented in an uncorrelated way in the space $\mathcal{R}^m$ as $X = Q_m \Lambda_m^{\frac{1}{2}}$.

To add s novel objects, represented by an $s \times n$ matrix of square distances $D_s^{(2)}$, relating s objects to the set T, a configuration X_s, projected onto $\mathcal{R}^m$, is

Table 1. Disagreement values, D*100, between classifiers built on the morphological features of the MFEAT set; since D is symmetric, only the upper part is presented.

	NMSC	LDC	UDC	QDC	1-NN	k-NN	Parzen	SVC1	SVC2	DT	ANN20	ANN50
NMC	47.1	47.3	43.4	50.3	53.5	30.9	24.1	63.1	71.4	50.4	77.5	72.8
NMSC	–	13.7	43.3	30.2	54.0	46.9	46.8	54.7	59.9	21.9	71.5	69.1
LDC	–	–	48.5	24.1	53.9	49.0	48.4	53.0	58.0	24.3	72.1	69.1
UDC	–	–	–	53.8	64.8	54.5	50.5	55.5	76.8	54.7	72.1	75.2
QDC	–	–	–	–	53.8	39.5	39.9	50.3	65.5	31.5	67.5	57.1
1-NN	–	–	–	–	–	48.5	49.5	65.5	78.7	53.9	77.7	77.0
k-NN	–	–	–	–	–	–	7.5	56.5	75.5	48.0	68.1	72.2
Parzen	–	–	–	–	–	–	–	56.7	73.2	48.1	68.8	71.2
SVC-1	–	–	–	–	–	–	–	–	79.1	54.2	36.7	89.9
SVC-2	–	–	–	–	–	–	–	–	–	65.0	84.2	86.7
DT	–	–	–	–	–	–	–	–	–	–	70.1	71.4
ANN20	–	–	–	–	–	–	–	–	–	–	–	100.0

then sought. Based on the matrix of inner products $B_s = -\frac{1}{2}(D_s^{(2)}J - UD^{(2)}J)$, where $U = \frac{1}{s}\mathbf{1}\mathbf{1}^T \in \mathcal{R}^{s\times n}$, X_s becomes $X_s = B_s X \Lambda_m^{-1}$ [6, 14].

In practice, often $m \approx n$, but the intrinsic dimensionality of the data is much smaller. Since X is an uncorrelated representation, the reduced configuration, preserving the distances approximately, is determined by k largest eigenvalues [4, 14]. Therefore, $X^{red} \in \mathcal{R}^k$, $k < m$, is found as $X^{red} = Q_k \Lambda_k^{\frac{1}{2}}$. If D is the matrix of diversities values between classifiers, X^{red} is the configuration in the sought CPS.

For a non-Euclidean distance, B has negative eigenvalues [4, 6] and X cannot be determined. One possibility is to consider a pseudo-Euclidean space, see [6, 14], another one is to skip the directions corresponding to the negative eigenvalues.

2.2 Multidimensional Scaling - A Nonlinear Projection

For an $n \times n$ dissimilarity matrix D, Sammon mapping [4] is a nonlinear MDS projection onto a a space $\mathcal{R}^k$ such that the distances are preserved. For this purpose, an error function, called *stress*, is defined, which measures the difference between the original dissimilarities and Euclidean distances of the configuration X of n objects. Let D be the given dissimilarity matrix and $\tilde{D}$ be the distance matrix for the projected configuration X. A variant of the stress [4] is here considered as

$$S = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}^2} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} (d_{ij} - \tilde{d}_{ij})^2.$$

To find an MDS representation, one starts from an initial representation and proceeds in an iterative manner until a configuration corresponding to a (local) minimum of S is found [4]. Here, for a stable solution, classical scaling is used to initialize the optimization procedure. If D is the matrix of diversities values between classifiers, X is the configuration in the CPS. Since there is no straightforward way of adding new objects to an existing MDS map, a modified version of the mapping has been proposed, which generalizes to new objects; see [3, 13].

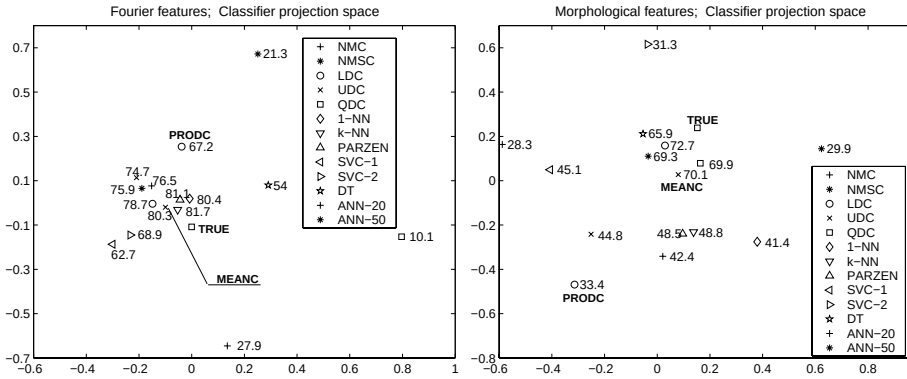

Fig. 2. A 2-dimensional CPS for the MFEAT dataset. Points correspond to classifiers; numbers refer to their accuracy. The 'perfect' classifier is marked as TRUE.

2.3 An Example

To present a 2-dimensional CPS, the 10-class MFEAT digit dataset [12] is considered. For our presentation, Fourier (74D) and morphological (6D) feature sets are chosen with a training set consisting of 50 randomly chosen objects per class. The classifiers considered are: the nearest (scaled) mean classifier NM(S)C, linear/uncorrelated quadratic/quadratic discriminant classifier LDC/UDC/QDC, 1/k-nearest neigbour rule 1-NN/k-NN, Parzen classifier, linear/quadratic support vector classifier SVC-1/SVC-2, decision tree DT and feed-forward neural network with 20/50 hidden units ANN20/ANN50. For each feature set, the disagreement matrix between all classifiers and two combiners, the mean (MEANC) and the product (PRODC) rules, is computed by formula (1); see also Table 1. This is done for the test set of 150 objects per class. The diversity matrix served then for a construction of a 2-dimensional CPS by the MDS procedure, described in section 2.2. Such examples of the CPS can be seen in Fig. 2. Note that the points correspond to classifiers. The distances between them approximate the original pairwise disagreement values, by which we can visually judge the similarities between classifiers. The hypothetical perfect classifier, i.e. given by the original labels, marked as TRUE, is also projected. The numbers in the plots show the accuracy reached on the test set. Let us emphasize that the axes cannot be interpreted themselves; it is simply distances that count.

In both cases, we can observe that the mean combiner is better than the product combiner. The latter, apparently deteriorates w.r.t. some, although diverse, but very badly performing classifiers. The mean rule seems to reflect the averaged variability of the most compact cloud. Note also that diversity might not be always correlated with accuracy. See, for instance, the right plot in Fig. 2, where the NMSC is more similar (less diverse) to the hypothetical classifier than ANN20, although the accuracy of the latter is higher.

3 Bagging, Boosting, and the Random Subspace Method

Many combining techniques can be used to improve the performance of weak classifiers. Examples are bagging [2], boosting [5] or the random subspace method (RSM) [7, 15]. They modify the training set by sampling the training objects (bagging), or by weighting them (boosting), or by sampling data features (the RSM).

Next, they build classifiers on these modified training sets and combine them into a final decision. Bagging is useful for linear classifiers constructed when the training size is about the data dimensionality. Boosting is effective for classifiers of low-complexity built on large training sets [15]. The RSM is beneficial for small training sets of a relatively large dimensionality, or for data with redundant features (where the discrimination power is spread over many features) [15].

To study the relations within those ensembles, the 34-dimensional, 2-class ionosphere data [1] is considered. The NMC is used for constructing the ensembles of 50 classifiers. The training is done on $T_1 = 100$ and $T_2 = 17$ objects per class (randomly chosen) to observe a different behaviour of base classifiers. The following combining rules are used: (weighted) majority voting, mean, product, minimum, maximum, decision templates and naive bayes (NB). The test set consists of 151 objects for which the disagreement matrix between the base classifiers of the mentioned ensembles and the combiners is com-

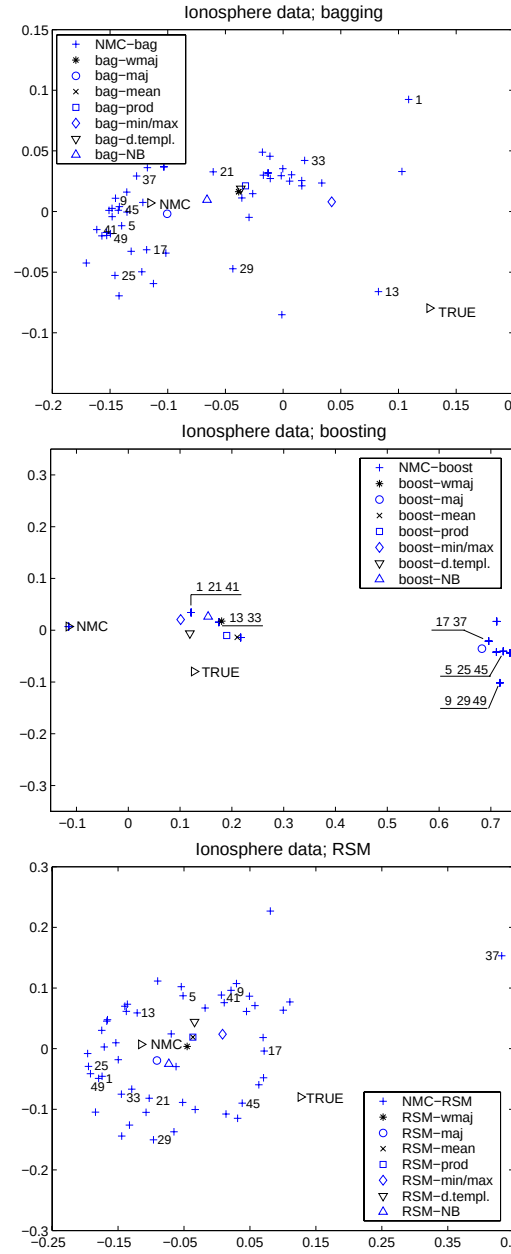

Fig. 3. A 2-dimensional CPS; Ionosphere dataset trained with T_1.

puted. Such a matrix serves for obtaining the CPS by using the MDS mapping, as described in section 2.2. The hypothetical, perfect classifier, representing true labels (marked as TRUE) has been added, as well; see Fig. 3.

To understand better the relation between the diversity and accuracy of the classifiers, while maintaining the clarity of presentation, another plots have been made; see Fig. 4 – 5. They show a 1-dimensional CPS (representing the relative difference in diversity) vs accuracy. So, the differences between classifiers in the horizontal and vertical directions correspond to the change in diversity and accuracy, respectively.

Analysing Fig. 3, 4 and 5, the following conclusions can be made. First of all, in the CPS, the classifiers obtained by bagging and the RSM are grouped around the single (original) NMC, creating mostly a compact cloud. The variability relations between the bagged and RSM classifiers might be very small. On the contrary, the boosted classifiers do not form a single cloud. In terms of both diversity and accuracy, they are reduced to 9 – 14 different ones (depending on the training set). A group of 5 – 8 poor classifiers is then completely separated from the others, as well as from the bagged and RSM classifiers.

Secondly, for a small training size T_2, Fig. 5, the RSM and bagging create classifiers that behave similarly in variability, since the classifier clouds in the 1D CPS are in the same range and of a similar size. For a larger training size T_1, Fig. 4, the diversity for the RSM classifiers is larger.

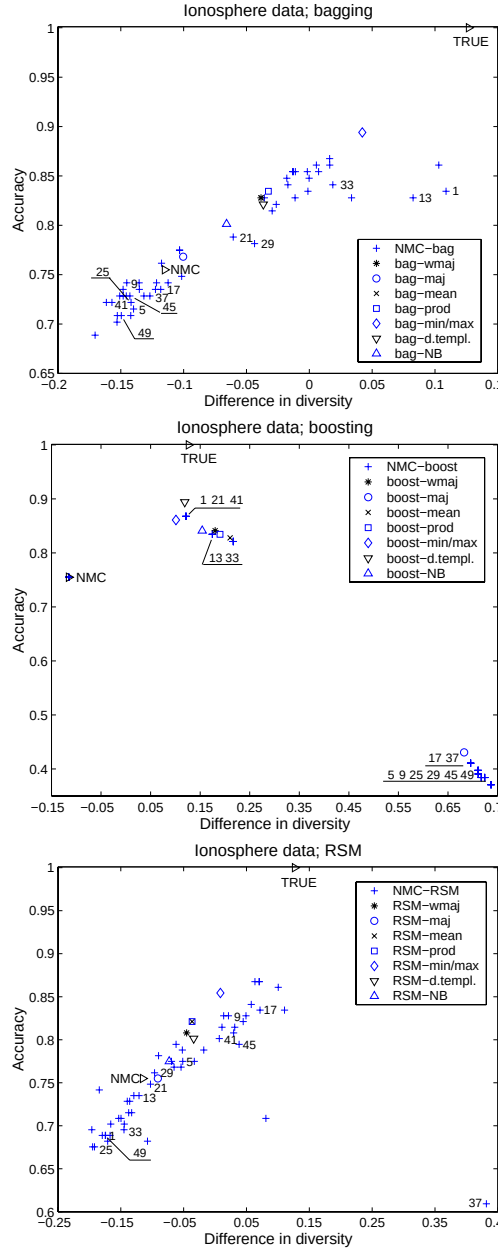

Fig. 4. Accuracy vs. 1D CPS; Ionosphere dataset trained with T_1.

Thirdly, the classifiers in all ensembles, even in boosting, seem to be constructed in a random order w.r.t. the diversity and accuracy.

Concerning the combiners studied here, the minimum rule (equivalent to the maximum rule for a 2-class problem) achieves, in most cases, the highest accuracy. It is even better than the weighted majority, used for the boosting construction. For a small sample size problem, Fig. 5, most of the combining rules for bagging and the RSM are alike, both in diversity and accuracy. A much larger variability is observed for boosting; a collection of diverse both classifiers and combiners is here obtained.

Finally, a striking observation is that nearly all classifiers, as well as their combiners, are placed in the CPS at one side (i.e. not around) of the perfect classifier (this was less apparent for the MFEAT data; compare to Fig. 2).

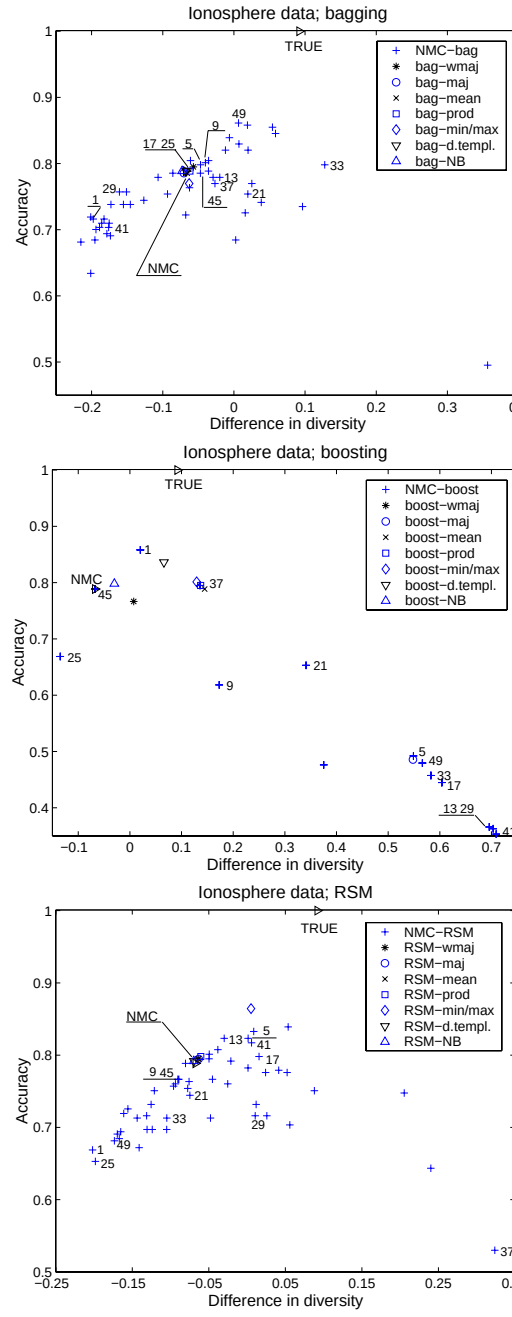

Fig. 5. Accuracy vs. 1D CPS; Ionosphere dataset; trained with T_2

4 Image Retrieval

In the problem of image database retrieval, images can be represented by single feature vectors or by clouds of points. Usually, given a query image Q, represented by a vector, images in the database are ranked according to their similarity to Q, measured e.g. by the normalized inner product. A cloud of points offers a more flexible representation, but it may suffer from overlap between cloud representations, even for very distinct images. Recently, we have pro-

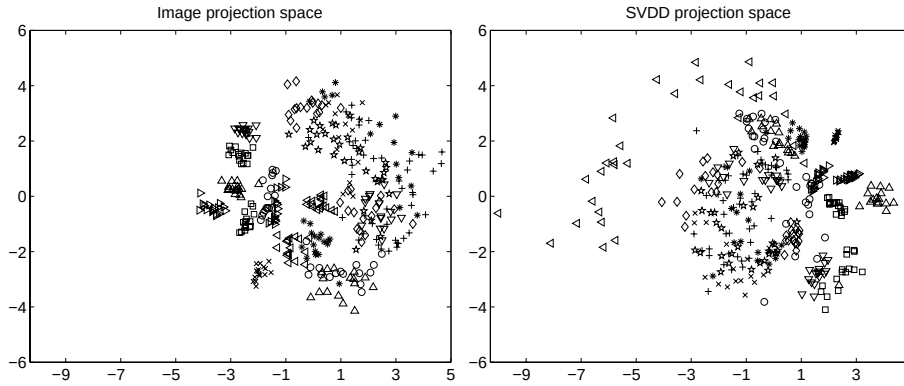

Fig. 6. 2D representations: images (left) and the CPS configuration for the SVDD-classifiers (right); different marks correspond to different classes.

posed a novel approach for describing clouds of points based on support vector data description (SVDD) [16], which try to describe the domain (boundary) of such a cloud. For each image in the database, such a SVDD is trained. The retrieval is based on the fraction of the points rejected by the SVDD's and the lowest ranks are returned. A single SVDD still suffers if the clouds of points between different images are highly overlapping. We have shown, however, that combining of the SVDD classifiers may improve the retrieval precision; see [10] for details.

In our experiment, performed on a dataset of texture images, 23 different images are given. Each original image is cut into 16 128×128 non-overlapping pieces. These correspond to a single class. Such pieces are mostly homogeneous and represent one type of a texture. The images are, one by one, considered as queries, and the 16 best ranked images are taken into account. The retrieval precision is computed using all 368 images; see [10] for details.

Each image is represented by a combined profile of all SVDD-classifiers. In our approach to the retrieval problem, a dissimilarity between a profile of the SVDD's for the query Q and the other images is considered. This might be based on the Euclidean distance. In order to see all the relations between images, a distance matrix and the resulting spatial representation of the images can be found, see Fig. 6, left plot. On the other hand, we can build the CPS, now based on the differences between SVDD-classifiers, see Fig. 6, right plot. Remind that in this case classifiers correspond directly to images, since each SVDD is a more conceptual description of an image. Comparing those two graphs, we see that the image space maintains a better separation, which was confirmed by our good retrieval precision [10].

5 Perspective on Possibilities of the CPS

The CPS has been used as a visualisation tool for analysing the differences between base classifiers and as an argument for the selection of some combining rules. Let us now discuss whether a CPS can be used for building classifier combiners. The figures presented in the previous sections contain just classifiers. If points, corresponding to classifiers, are close in a plot, the classifiers are similar. This may be an argument to select just one of a cluster of related classifiers or to average their outputs in order to reduce the noise. Very different classifiers should be preserved since they may be candidates for the product combiner. In this way, the relative positions of the classifiers in the CPS may serve for a construction of the overall system architecture.

In order to train a new classifier, using the CPS space, it is highly desirable to project training objects in such a space. In case of a linear embedding (section 2.1), the mapping of new classifiers into an existing CPS is well defined. In order to project an object into this space, an equivalent dissimilarity measure between objects and classifiers should be defined. Here, we face the problem that an object belongs to a single class and a classifier is a multi-class entity. In section 2, the behaviour w.r.t. the distinct classes was averaged, as in the disagreement measure (1), just differences in label assignments are counted, neglecting further class differences. The measure is, therefore, modified into a matrix of numbers. The disagreement between the classifiers $\mathcal{C}_i$ and $\mathcal{C}_j$ w.r.t. the classes p and q can be written as

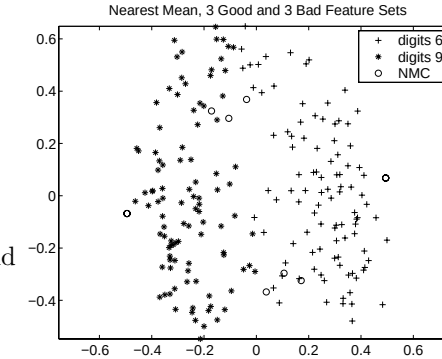

Fig. 7. The CPS with objects.

$$D_{p,q}(\mathcal{C}_i, \mathcal{C}_j) = \text{Prob}\,(\mathcal{C}_i = p, \mathcal{C}_j = q) + \text{Prob}\,(\mathcal{C}_i = q, \mathcal{C}_j = p), \tag{2}$$

where the probabilities are taken w.r.t. the set of objects x to be classified. For a c-class problem and n classifiers, the total size of the dissimilarity matrix is then $nc \times nc$. Now, for an object y, a similar quantity is defined as

$$D_{y,q}(y, \mathcal{C}_j) = \text{Prob}\,(\mathcal{C}_j(q, x) > \mathcal{C}_j(q, y)\,), \tag{3}$$

resulting in a $1 \times nc$ row vector, since $\mathcal{C}_j(q, x)$ is the support of classifier $\mathcal{C}_j$, $j = 1, \ldots, n$ for an object x w.r.t. the class q, $q = 1, \ldots, c$. The probability is zero if no other object x exists with more support for the given class q than the presented object y. This is in agreement with the concept of a dissimilarity since this implies that this object is very q-like according to $\mathcal{C}_j$.

An example is presented in Fig. 7. We computed six NMC for all six feature sets of the MFEAT dataset [12] between the classes '6' and '9'. From the 12×12 dissimilarity matrix between the classifiers, a 2-dimensional CPS is found by

classical scaling. 100 objects per class are projected into this space. Classifiers and objects are shown in Fig. 7. The 2×3 classifiers at the top and at the bottom correspond to bad feature sets that cannot distinguish between classes '6' and '9' as they are based on rotation invariant properties. The 3 classifiers, corresponding to good feature sets, are projected right on the top of each other on the left and on the right sides.

6 Discussion and Conclusion

We presented a new way of representing classifiers. The classifier projection space (CPS), based on (approximate) embedding of the diversities between the classifiers, offers a possibility to study their differences. This may increase the understanding of the recognition problem at hand and, thereby, offers an analyst a tool based on which he can decide on the architecture of the entire combining system.

We also showed how objects can be mapped into the CPS. It has to be further investigated under what circumstances the construction of a combined output classifier in such a space is beneficial. This will be a trained combiner and its performance has to be compared with the direct use of the base classifier outputs as their features. The advantage of the presented approach is that by the choice of the dissimilarity measure the character of these 'features' as classifier outputs may be preserved.

Acknowledgments

This work is supported by the Dutch Organization for Scientific Research (NWO). The authors thank dr Ludmila Kuncheva for supplying some of the routines.

References

[1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
[3] D. Cho and D.J. Miller. A Low-complexity Multidimensional Scaling Method Based on Clustering. *concept paper*, 2002.
[4] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, 1995.
[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proc. of the 13th International Conference*, pages 148–156, 1996.
[6] L. Goldfarb. A new approach to pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier Science Publishers B.V., 1985.
[7] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on PAMI*, 20(8):832–844, 1998.
[8] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on PAMI*, 20(3):226–239, 1998.

[9] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles. *submitted*, 2002.
[10] C. Lai, D.M.J. Tax, R.p.W. Duin, P. Paclík, and E. Pękalska. On combining one-class classifiers for image database retrieval. In *International Workshop on Multiple Classifier Systems*, Cagliari, Sardinia, 2002.
[11] L. Lam. Classifier combinations: implementation and theoretical issues. In *Multiple Classifier Systems, LNCS*, volume 1857, pages 78–86, 2000.
[12] MFEAT: ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/.
[13] E. Pękalska and R.P.W. Duin. Spatial representation of dissimilarity data via lower-complexity linear and nonlinear mappings. In *Joint International Workshop on SSPR and SPR*, Windsor, Canada, 2002.
[14] E. Pękalska, P. Paclík, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity Based Classification. *Journal of Mach. Learn. Research*, 2:175–211, 2001.
[15] M. Skurichina. *Stabilizing Weak Classifiers*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2001.
[16] D.J.M. Tax. *One-class classifiers*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2001.