Mapping Classifiers and Datasets

Olcay Taner Yıldız

Department of Computer Engineering, Işık University, 34398, İstanbul Turkey

Abstract

Given the posterior probability estimates of 14 classifiers on 38 datasets, we plot two dimensional maps of classifiers and datasets using Principal Component Analysis (PCA) and Isomap. The similarity between classifiers indicate correlation (or diversity) between them and can be used in deciding whether to include both in an ensemble. Similarly, datasets which are too similar need not both be used in a general comparison experiment. The results show that (i) most of the datasets (approximately two third) we used are similar to each other, (ii) multilayer perceptrons and k-nearest neighbor variants are more similar to each other than support vector machine and decision tree variants, (iii) the number of classes and the sample size has an effect on similarity.

Key words: classifiers, datasets, no free lunch theorem, pca, isomap

1. Introduction

In machine learning, when we draw conclusions, it is conditioned on the dataset we are given. When we compare two different classification algorithms on a particular dataset, any result we have will be true only for that

Email address: olcaytaner@isikun.edu.tr (Olcay Taner Yıldız)

particular dataset. There is no such thing as the "best" learning algorithm. For an algorithm, there may be a dataset where it is very accurate and another dataset where its performance is very poor. According to the no free lunch theorem, when we say a classification algorithm is good, we only say how well its inductive bias matches the properties of the dataset (1).

In this paper, our aim is to 'map' well known classification algorithms and datasets to a two-dimensional space so that we can easily visualize how similar and how different classifiers / datasets are. To accomplish this, we first produce two meta-datasets, for classifiers and datasets respectively. The attributes of those two datasets are generated from the posterior probability estimates of 14 classifiers on the test sets of 38 datasets. We use PCA and Isomap as linear and nonlinear dimension reduction techniques respectively to reduce number of dimensions to two and plot classifiers / datasets as points in this 2D plane.

In Section 2, we give brief descriptions of two dimension reduction techniques we used in the paper. We give our experiments and results in Section 3 and conclude in Section 4.

2. Dimension Reduction Techniques

2.1. Principal Component Analysis

Principal Component Analysis (PCA) (2) projects data points $x_i \in \Re^d$ onto lower dimensional coordinates $y_j \in \Re^p$ for best information preservation. The linear projection is given by

$$\mathbf{Y} = \mathbf{X}\mathbf{W} \tag{1}$$

where $\mathbf{W}$ is an $d \times p$ projection matrix found to maximize the variance of $\mathbf{Y}$. To satisfy this purpose, $\mathbf{W}$ contains eigenvectors (principal components) in decreasing order of respective eigenvalues of the covariance matrix of $\mathbf{X}$ as columns. The top two eigenvectors are used to reduce dimension to two.

2.2. Isomap

Isomap inherits the advantages of PCA and multidimensional scaling (MDS) and extends these to learn non-linear structures that are hidden in high dimensional data (3).

Normally to calculate the similarity of two instances, Euclidean distance is used. However, the use of the Euclidean distance to represent pairwise distances makes the model unable to preserve the intrinsic geometry of the manifold. Two nearby points, in terms of Euclidean distance, may indeed be distant, because their actual distance is the path between these points along the manifold. The length of the path along the manifold is referred to as the geodesic distance. Isomap uses this distance metric and then performs classical MDS. Geodesic distance represents similar or different data points more accurately than the Euclidean distance, but the task is to estimate it accurately. Here the local linearity principle is used and it is assumed that neighboring points lie on a linear patch of the manifold, so for nearby points the Euclidean distances correctly estimate the geodesic distances. For distant points, the geodesic distances are estimated by adding up neighboring distances over the manifold.

Isomap finds the true dimension of nonlinear structures as long as sufficient data is supplied. The only parameter of the method is k which determines the neighboring information, and which should be fine tuned to get

accurate results.

3. Experiments

3.1. Experimental Setup

3.1.1. Base Datasets

We use a total of 38 base datasets where 35 of them are from UCI (4) and 3 are from Delve (5) repositories (see Table 1).

3.1.2. Base classifiers

We use fourteen base classifiers which we have chosen to span as much as possible the wide spectrum of possible machine learning algorithms:

1–3) *knn*: k-nearest neighbor with $k = 1, 3, 5$.

4–8) *mlp*: Multilayer perceptron where with D inputs and K classes, the number of hidden units is taken as D (*mlp1*), K (*mlp2*), $(D + K)/2$ (*mlp3*), $D + K$ (*mlp4*), $2(D + K)$ (*mlp5*).

9) *lp*: Linear perceptron with softmax outputs trained by gradient-descent to minimize cross-entropy.

10) *c45*: The most widely-used C4.5 decision tree algorithm (6).

11) *ldt*: This is a multivariate tree where unlike C4.5 which uses univariate and axis-orthogonal splits uses splits that are arbitrary hyper-planes using all inputs (7).

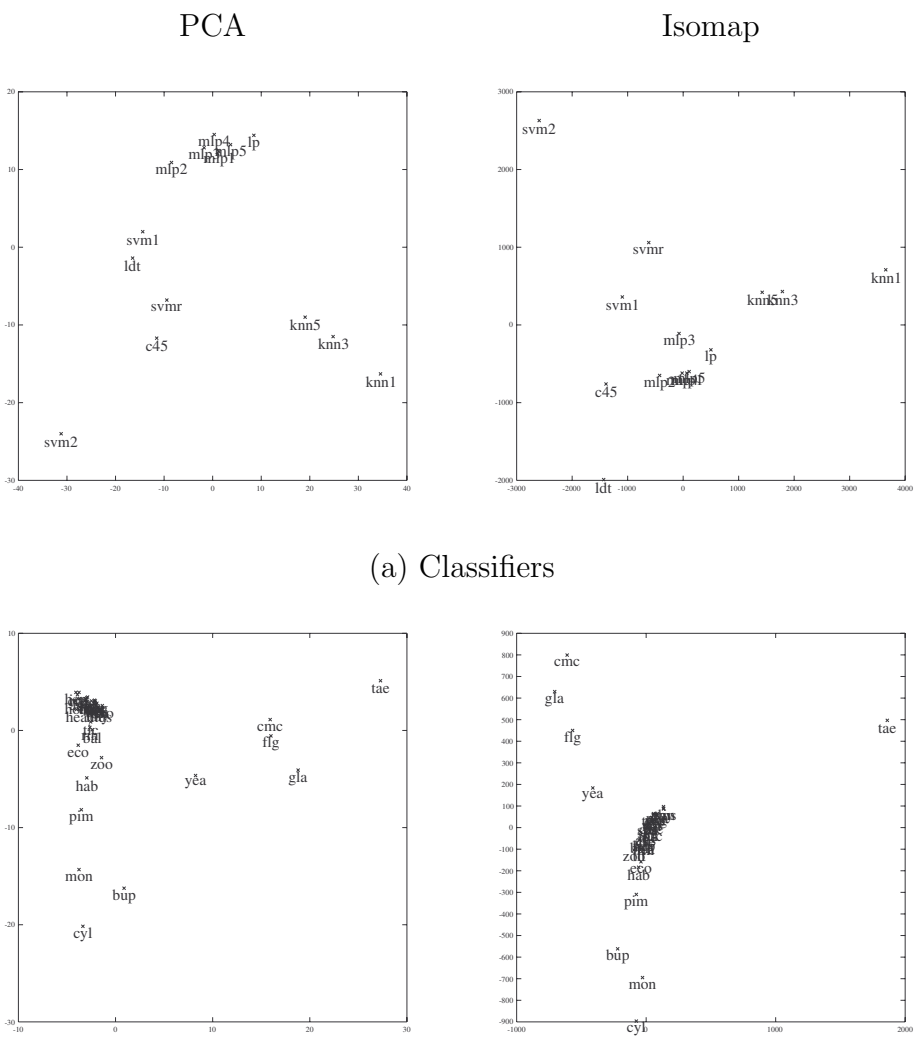
12–14) *svm*: Support vector machines (SVM) with a a linear kernel (*sv1*), polynomial kernel of degree 2 (*sv2*), and a radial (Gaussian) kernel (*svr*). We use the LIBSVM 2.82 library that implements pairwise SVMs (8).

Table 1: Datasets

Dataset	**Class**	**Instance**	**Dataset**	**Class**	**Instance**
australian	2	690	monks	2	432
balance	3	625	mushroom	2	8124
breast	2	699	nursery	5	12960
bupa	2	345	optdigits	10	3823
car	4	1728	pageblock	5	5473
cmc	3	1473	pendigits	10	7494
credit	2	690	pima	2	768
cylinder	2	540	ringnorm	2	7400
dermatology	6	366	segment	7	2310
ecoli	8	336	spambase	2	4601
flags	8	194	tae	3	151
flare	3	323	thyroid	4	2800
glass	6	214	tictactoe	2	958
haberman	2	306	titanic	2	2201
heart	2	270	twonorm	2	7400
hepatitis	2	155	vote	2	435
horse	2	368	wine	3	178
iris	3	150	yeast	10	1484
ionosphere	2	351	zoo	7	101

3.1.3. Division of training, validation, and test sets

The methodology is as follows: A dataset is first divided into two parts, with 1/3 as the test set, *test*, and 2/3 as the training set, *train-all*. The training set, *train-all*, is then resampled using 5×2 cross-validation (cv) (9) where 2-fold cv is done five times (with stratification) and the roles swapped at each fold to generate ten training and validation folds, tra_i, val_i, $i = 1, \ldots, 10$. tra_i

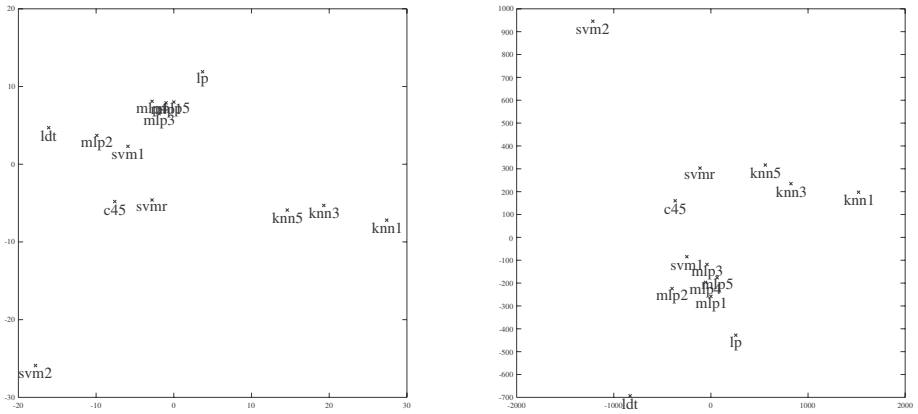


(a) Classifiers

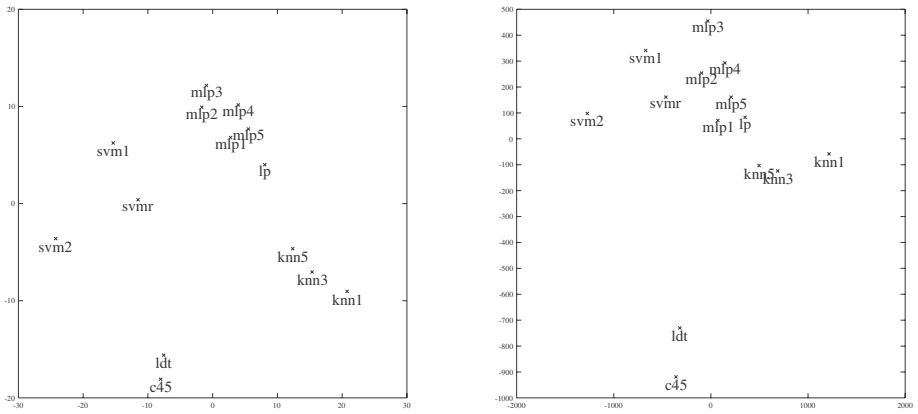
(b) Datasets

Figure 1: Plot of classifiers and datasets after PCA and Isomap.

are used to train the base classifiers. These ten trained algorithms are tested on the same *test* and we have ten $test_i$ accuracy results on which we run the dimension reduction methods.



(a) $K = 2$ class datasets

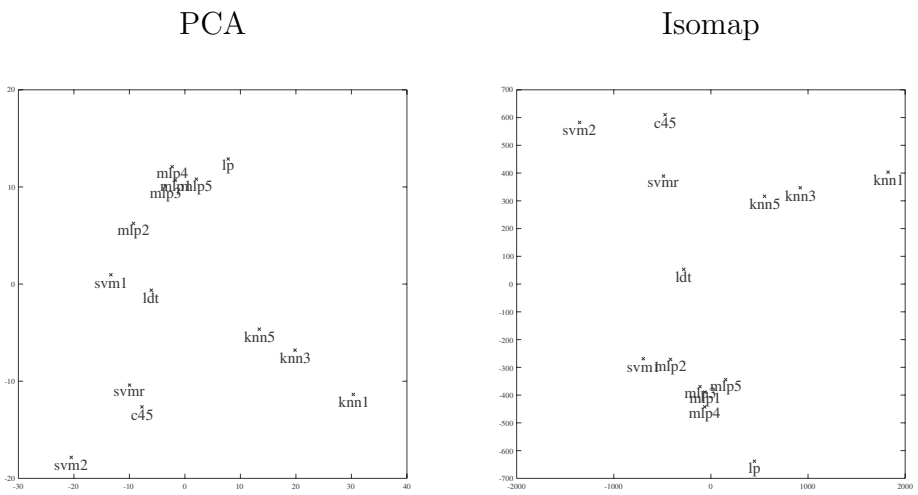


(b) $K > 2$ class datasets

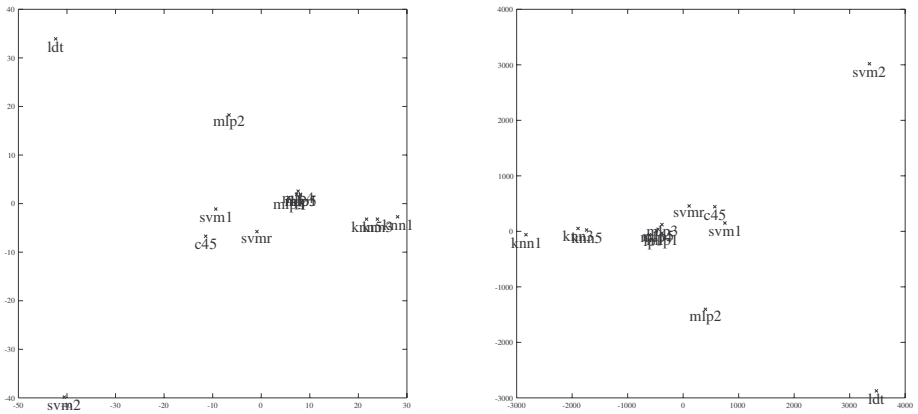
Figure 2: Plot of classifiers for two class and $K > 2$ class datasets after PCA and Isomap.

3.2. Meta-datasets

From the results of base-classifiers on all datasets we generate two meta-datasets for classifiers and datasets respectively.



(a) Small size ($N < 1000$) datasets



(b) Large size ($N > 1000$) datasets

Figure 3: Plot of classifiers for small size and large size datasets after PCA and Isomap.

The first meta-dataset contains 14 instances for the classifiers. From each of the 38 datasets, we randomly take 30 instances and the prediction of the classifier for the correct class is recorded, when concatenated this forms a

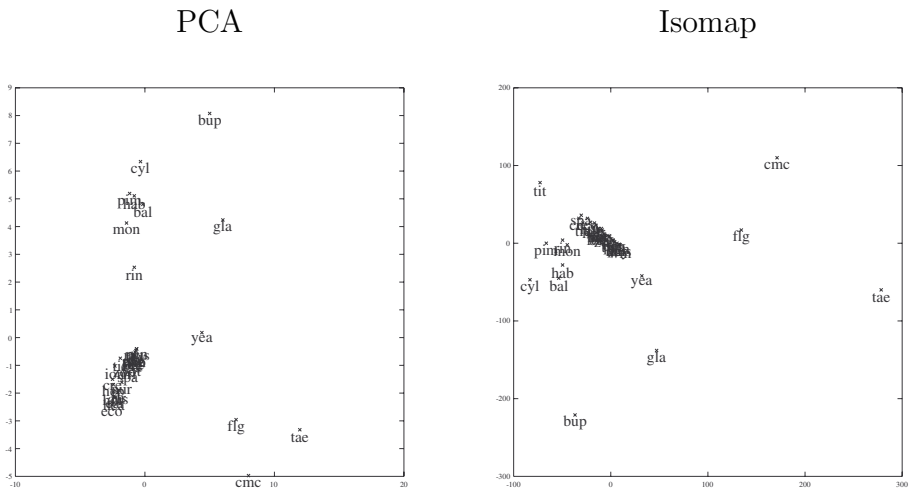


Figure 4: Plot of datasets for knn base classifiers after PCA and Isomap.

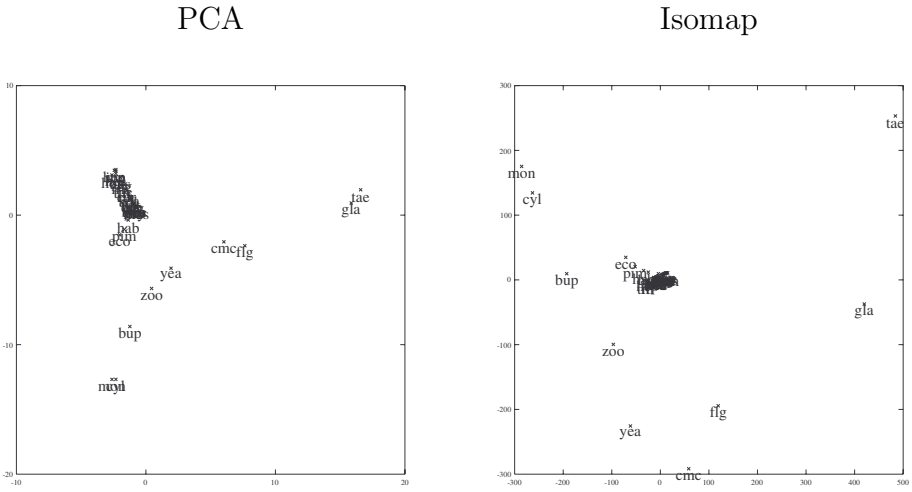


Figure 5: Plot of datasets for mlp base classifier after PCA and Isomap.

$30 \cdot 38 = 1140$ dimensional vector which is the data point for a classifier. So we have a dataset of size 14×1140.

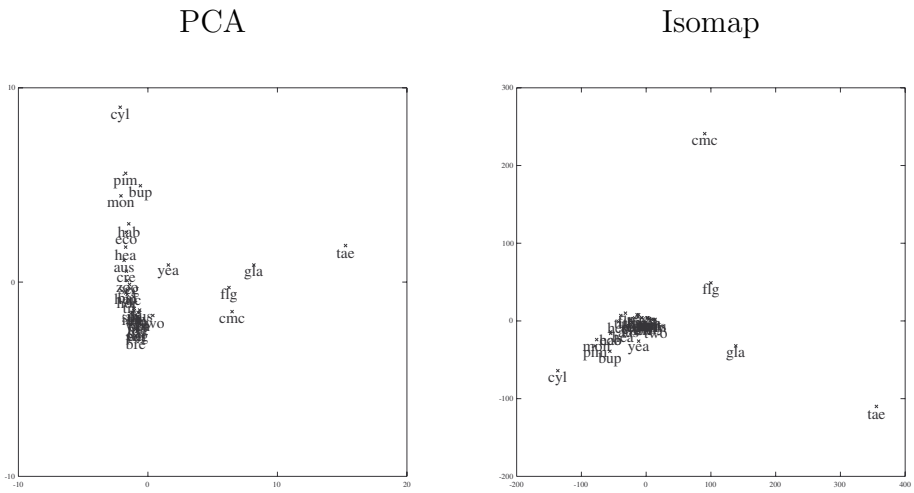


Figure 6: Plot of datasets for svm base classifier after PCA and Isomap.

The second meta-dataset contains 38 instances for datasets. For each of the 14 classifier, its accuracy on the ten test folds need be reported. For this, we divide the percentage into 40 equal intervals (0-2.5, 2.5-5, ..., 95-97.5, 97.5-100) and count how many of the ten $test_i$ accuracy results fall into each interval (that is we form a histogram with 40 bins). So we have a dataset of size $14 \times (14 \cdot 40 = 560)$.

3.3. Results

Figure 1 shows the plot of classifiers and datasets after PCA and Isomap. If we look at Figure 1(a), after both PCA and Isomap, we see that multi-layer perceptron (mlp) algorithms, k-nearest neighbor algorithms (k-nn) and decision tree algorithms form clusters of their own. This is expected; changing the hyper-parameter causes a slight change. k-nn variants get similar to other algorithms as k increases. Support vector machine (svm) with the

quadratic kernel seems an outlier. Linear perceptron (lp) is similar to mlp variants which may be due to easiness of the datasets where linear models work nearly as well as nonlinear methods.

If we look at Figure 1(b), we see that almost two third of all datasets are similar to each other. Therefore, one must be very careful in selecting datasets to include in a comparison experiment. Other than those, there are five different dataset groups (*pim*, *hab*, *zoo*, *eco*), (*mon*, *bup*, *cyl*), (*cmc*, *flg*, *gla*), (*tae*), (*yea*). Though the exact coordinates may differ, both PCA and Isomap seem to be finding the same clustering and in that respect, there is not much difference between the results of the two methods.

We then checked if the number of classes is a factor. For this, we divide the datasets into two, with $K = 2$ class and $K > 2$ class problems and reduce dimension separately. Three of our base classifiers (decision trees, svms and mlps) behave differently when we have more than two classes in the dataset. Two-class versions of mlp are more similar to svms. Svms are mainly two-class classifiers, if there are more than two classes, one resorts to one-vs-one or one-vs-all or other approaches (In our implementation we used one-vs-one approach). Mlps use K output units for $K > 2$ class discrimination whereas for two-class discrimination one output unit suffices.

There are decision tree algorithms which make m-ary splits but most of them including *c45* and *ldt* use binary splits. In that case, one node may be sufficient to separate two classes but at least $K - 1$ nodes are needed to separate $K > 2$ classes, where one must optimally divide class groups not only single class. The similarity between *c45* and *ldt* (univariate and multivariate) trees increase when K is increased from two (Figure 2). We

also see that as we go from $K = 2$ to $K > 2$, svm with quadratic kernel is now more similar to other svms and mlps are more distinguishable.

Not only the class size, but also the sample size is a factor in classifier similarities. As the sample size increases, the amount of training and validation data increases. These result in a decrease in generalization error and better performance on the test set. With larger training sets, we expect classifiers to have smaller variance and therefore get closer to each other. Therefore, we divide the datasets into two groups as small size datasets ($N < 1000$) and large size datasets ($N > 1000$). Figure 3 shows the plot of classifiers for small size and large size datasets after PCA and Isomap. As the sample size increases, we expect k-nn variants and mlp variants (with the exception of mlp2) to get near to each other as seen in the figures. Whereas for svms, radial basis svm and linear svm get similar but svm with the quadratic kernel is still far.

We then checked to see if we can group datasets using not all the classifiers but variants of a single algorithm. For this, we divide the classifiers into three as k-nn, mlp and svm classifiers and reduce dimension separately. The plots of the datasets for knn, mlp and svm base classifiers after PCA and Isomap can be seen in Figures 4, 5 and 6 respectively. Except for some changes, we see more or less the same datasets grouped together; this indicates that the similarity does not depend to much on the algorithm but rather in some intrinsic properties of the dataset.

4. Discussion

It has been proposed (10) to use k-Nearest Neighbor algorithm to identify the datasets that are most similar to the one at hand. The distance between datasets is assessed using a relatively small set of data characteristics, which were selected to represent properties that affect algorithm performance.

Intrinsic properties of the datasets and their relations with classification performance have been used by (11). They propose 12 complexity measures for two class supervised classification problems that characterize the difficulty of a classification problem. The metrics they propose focus on the geometrical properties of the class boundary. In another work (12), datasets are characterized using *meta-attributes* which use general, statistical and information theoretic measures. Such measures can also be used together with posterior probability estimates of example classifiers to be able to find similarities between datasets.

There does not seem to be much difference between PCA and Isomap results in that both seem to find similar clustering of data points (classifiers/datasets).

The benefit of finding similarity between datasets or between classifiers is threefold: First, if we know which datasets are similar and which datasets are different, one can devise a more informative experiment in testing algorithms.

Ensemble methods require that the base-classifiers be accurate on different instances, specializing in sub-domains of the dataset. Similarity between classifiers can be used as a diversity measure and those that are too close need not be both included in the ensemble. For example, we see that *1nn* and *3nn* are very close but *svr* and *sv2* are not.

Automatic systems that can recommend good classifiers would be very useful in data mining applications where users need not be experts in machine learning (13). A similarity calculation strategy as we do in this paper would be useful in such a case.

References

[1] D. H. Wolpert, The relationship between pca, the statistical physics framework, the bayesian framework, and the vc framework, in: D. H. Wolpert (Ed.), The Mathematics of Generalization, Addison Wesley, MA, 1995, pp. 117–214.

[2] A. C. Rencher, Methods of Multivariate Analysis, Wiley and Sons, 1995.

[3] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

[4] C. Blake, C. Merz, UCI repository of machine learning databases (2000). URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[5] G. H. Hinton, Delve project, data for evaluating learning in valid experiments (1996).
URL `http://www.cs.utoronto.ca/~delve/`

[6] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Meteo, CA, 1993.

[7] W. Y. Loh, Y. S. Shih, Split selection methods for classification trees, Statistica Sinica 7 (1997) 815–840.

[8] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines (2001).
URL http://www.csie.ntu.edu.tw/∼cjlin/libsvm

[9] T. G. Dietterich, Approximate statistical tests for comparing supervised classification learning classifiers, Neural Computation 10 (1998) 1895–1923.

[10] P. B. Brazdil, J. P. da Costa, Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results, Machine Learning 50 (2003) 251–277.

[11] T. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Transactions on Pattern Analysis and Intelligence 24 (3) (2002) 289–300.

[12] R. Henery, Methods for comparison, in: D. Michie, D. Spiegelhalter, C. Taylor (Eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwood, 1994, pp. 107–124.

[13] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2000.