



# Mapping classifiers and datasets

Olcay Taner Yıldız

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

## ARTICLE INFO

### Keywords:

Classifiers

Datasets

No free lunch theorem

PCA

Isomap

## 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.

© 2010 Elsevier Ltd. All rights reserved.

## 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 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 (Wolpert, 1995).

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) (Rencher, 1995) projects data points  $x_i \in \mathbb{R}^d$  onto lower dimensional coordinates  $y_j \in \mathbb{R}^p$  for best information preservation. The linear projection is given by

$$\mathbf{Y} = \mathbf{X}\mathbf{W}, \quad (1)$$

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 nonlinear structures that are hidden in high dimensional data (Tenenbaum, de Silva, & Langford, 2000).

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

E-mail address: [olcaytaner@isikun.edu.tr](mailto:olcaytaner@isikun.edu.tr)

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 (Blake & Merz, 2000) and 3 are from Delve (Hinton, 1996) 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)  $k$ -nn:  $k$ -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  $D$  ( $mlp1$ ),  $K$  ( $mlp2$ ),  $(D + K)/2$  ( $mlp3$ ),  $D + K$  ( $mlp4$ ),  $2(D + K)$  ( $mlp5$ ).

**Table 1**  
Datasets.

Dataset	Class	Instance
Australian	2	690
Balance	3	625
Breast	2	699
Bupa	2	345
Car	4	1728
Cmc	3	1473
Credit	2	690
Cylinder	2	540
Dermatology	6	366
Ecoli	8	336
Flags	8	194
Flare	3	323
Glass	6	214
Haberman	2	306
Heart	2	270
Hepatitis	2	155
Horse	2	368
Iris	3	150
Ionosphere	2	351
Monks	2	432
Mushroom	2	8124
Nursery	5	12960
Optdigits	10	3823
Pageblock	5	5473
Pendigits	10	7494
Pima	2	768
Ringnorm	2	7400
Segment	7	2310
Spambase	2	4601
Tae	3	151
Thyroid	4	2800
Tictactoe	2	958
Titanic	2	2201
Twonorm	2	7400
Vote	2	435
Wine	3	178
Yeast	10	1484
Zoo	7	101

- (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 (Quinlan, 1993).
- (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 (Loh & Shih, 1997).
- (12)–(14)  $svm$ : support vector machines (SVM) with 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 (Chang & Lin, 2001).

##### 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, and 2/3 as the training set, *train-all*. The training set, *train-all*, is then resampled using  $5 \times 2$  cross-validation (cv) (Dietterich, 1998) where two-fold cv is done five times (with stratification) and the roles swapped at each fold to generate ten training and validation folds,  $tra_i$ ,  $val_i$ ,  $i = 1, \dots, 10$ .  $tra_i$  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.

#### 3.2. Meta-datasets

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

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

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

Fig. 1 shows the plot of classifiers and datasets after PCA and Isomap. If we look at Fig. 1(a), after both PCA and Isomap, we see that multilayer 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 Fig. 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 ( $pim, hab, zoo, eco$ ), ( $mon, bup, cyl$ ), ( $cmc, flg, gla$ ), ( $tac$ ), ( $yea$ ). 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 prob-

Download English Version:

<https://daneshyari.com/en/article/386030>

Download Persian Version:

<https://daneshyari.com/article/386030>

[Daneshyari.com](https://daneshyari.com)