Genetic Algorithm for Linear Feature Extraction

Alberto J. Pérez-Jiménez & Juan Carlos Pérez-Cortés[1]
Universidad Politécnica de Valencia
Spain

1. Introduction

Feature extraction is a commonly used technique applied before classification when a number of measures, or features, have been taken from a set of objects in a typical statistical pattern recognition task. The goal is to define a mapping from the original representation space into a new space where the classes are more easily separable. This will reduce the classifier complexity, increasing in most cases classifier accuracy. Feature extraction methods can be divided into linear and non-linear, depending on the nature of the mapping function (Lerner et al., 1998). They can also be classified as supervised or unsupervised, depending on whether the class information is taken into account or not. Feature extraction can also be used for exploratory data analysis, where the aim is not to improve classification accuracy, but to visualise high dimensional data by mapping it into the plane or the 3-dimensional space.

The best known linear methods are Principal Component Analysis, or PCA (unsupervised) (Fukunaga, 1990), Linear Discriminant Analysis or LDA (supervised) (Fukunaga, 1990; Aladjem, 1991; Siedlecki et al., 1988), and Independent Component Analysis or ICA (unsupervised) (Cardoso, 1993). Schematically, PCA preserves as much variance of the data as possible, LDA attempts to group patterns of the same class, while separating them from the other classes, and ICA obtains a new set of features by extracting the less correlated (in a broad sense) directions in the data set. On the other hand, well-known non-linear methods are: Sammon's Mapping (unsupervised) (Sammon, 1969; Siedlecki et al. 1988), non-linear discriminant analysis or NDA (supervised) (Mao & Jain, 1995), Kohonen's self-organising map (unsupervised) (Kohonen, 1990) and evolutionary extraction (supervised) (Liu & Motoda, 1998). Sammon's mapping tries to keep the distances among the observations using hill-climbing or neural network methods (Mao & Jain, 1995; Sammon, 1969), NDA obtains new features from the coefficients of the second hidden layers of a multi-layer perceptron (MLP) (Mao & Jain, 1995) and Kohonen Maps project data in an attempt to preserve the topology. Finally, evolutionary extraction uses a genetic algorithm to find combinations of original features in order to improve classifier accuracy. These new features are obtained by multiplying, dividing, adding or subtracting the original features.

In the linear methods, the mapping function is known and simple; therefore, the task is reduced to finding the coefficients of the linear transformation by maximising or minimising

[1] Work partially supported by the Spanish "Ministerio de Educación y Ciencia" under grant DPI2006-15542-C04-02.

Source: Vision Systems: Segmentation and Pattern Recognition, ISBN 987-3-902613-05-9,
Edited by: Goro Obinata and Ashish Dutta, pp.546, I-Tech, Vienna, Austria, June 2007

a criterion. If a proper criterion function is selected, many standard linear algebra methods can be applied. However, in many cases a linear mapping may not be powerful enough to obtain good results, making it necessary to consider non-linear mappings.

Non-linear mappings present different functional forms and this often makes their application more problem-dependent. Furthermore, since closed-form optimisation methods for many non-linear functions are not known or, are in general less stable and powerful than their linear counterparts when they do exist, non-parametric estimation techniques such as neural networks or iterative optimisation procedures such as hill-climbing or genetic algorithms are commonly used.

In this paper, a new linear supervised feature extraction method referred to as GLP (genetic linear projections) is proposed. The goal of this method is to find the coefficients of a set of linear projections by maximising a certain criterion function. The success confidence rate in the new feature space, a criterion that is directly related to the estimated accuracy of a Nearest Neighbour classifier, is proposed as the function to maximise. Because no closed-form solution exists to maximise this criterion, a well-known numerical optimisation method, genetic algorithms (GA) (Holland, 1975; Goldberg, 1989), has been employed.

In Section 2, we describe the GLP algorithm. In Section 3, we present a comparison between a linear method (PCA), a non-linear method (NDA) and the proposed GLP algorithm over several data sets in terms of both feature extraction and data projection purposes. Finally, we present some conclusions and further works in section 4.

2. Genetic Linear Projection (GLP)

2.1 Linear feature extraction

In linear feature extraction, new features are obtained by means of linear projections (LP). A LP is defined as follow

$$\mathrm{LP}(x) = c_1x_1 + c_2x_2 + \dots c_dx_d \tag{1}$$

where x is a d-dimensional vector with components x_i and c_i are the projection coefficients representing the projection axis. By representing the coefficients as a vector, $c=\{c_1, c_2, \dots c_d\}^T$, the application of a LP can be redefined as

$$\mathrm{LP}(x) = c^Tx \tag{2}$$

Each LP defines a new feature to represent x. To define m new features, *w*e need m LPs that can be arranged as a $m \times d$ matrix (C). By defining the transformation matrix C in this way, a new representation of x, $y=\{y_1, y_2, \dots y_m\}^T$, can be obtained by means of

$$y = Cx \tag{3}$$

Ideally C should be selected in order to minimise the Bayes error (Duda & Hart, 1973) in the new space. Moreover, this expression depends on the *a posteriori* probability of classes, and in general, is not straightforward to obtain. Even when this expression exists, usually no tractable expression for the gradient can be obtained. For this reason, linear feature extraction methods often employ other less suitable, but simpler, class separability measures in order to use closed-form solutions, or they employ gradient-based numerical optimisation methods in order to obtain C.

2.2 Criteria

In this work, we propose to obtain *C* by optimising a criterion function that is directly related to the Bayes error. The estimated error rate, *Ê*, of a *k*-Nearest Neighbour classifier (*k*-NN) can be a good option. Under certain convergence conditions, the error rate of a *k*-NN classifier offers an optimistic, but very close estimation of the Bayes error (Devijver & Kittler, 1982). The *Ê* can be easily calculated by error count over a test set by the expression,

$$\hat{E} = \frac{e}{n} \tag{4}$$

where *n* is the size of the test set, and *e* is the number of observations that are not correctly classified by the *k*-NN classifier. The estimated success rate of a classifier, *Â*, is directly related to *Ê* and can be calculated as *Â=1-Ê*.

Another interesting criterion can be defined using the conditional probability of an observation *x* belonging to a class *wᵢ*, $P(w_i \mid x)$. Most statistic classifiers can provide an estimation of this value that can be used as a confidence measure for the classified observations. In a *k*-NN classifier, a maximum likelihood estimation of $P(w_i \mid x)$, $\hat{P}(w_i \mid x)$, can be obtained as

$$\hat{P}(w_i \mid x) = \frac{k_i}{k} \tag{5}$$

where *k* is the number of neighbours employed by the *k*-NN classifier, and k_i is the number of neighbours of class w_i. We formulate the *estimated success confidence rate*, $\hat{C}_a$, of a classifier as

$$\hat{C}_a = \frac{1}{n}\sum_{x \in X} \delta(w_{\theta'_x}, w_{\theta_x})\hat{P}(w_{\theta_x}, x) \tag{6}$$

where *n* is the number of observations of a sample *X*, w_{θ_x} is the real class of $x \in X$, $w_{\theta'_x}$ is the class assigned to *x* by the employed classifier, and $\delta(w_i, w_j)$ is defined as

$$\delta(w_i, w_j) = \begin{cases} 1 & if \quad w_i = w_j \\ 0 & if \quad w_i \neq w_j \end{cases} \tag{7}$$

In the case the value $\hat{P}(w_i \mid x)$ is always *1*, the definition of $\hat{C}_a$ equals the value of *Â*. The criterion can be seen as a confidence measure of the estimated success rate of a classifier.

When projecting data, the use of $\hat{C}_a$ as the optimisation criterion has advantages with respect to *Â* (or *Ê*). Two projections with the same *Â* value can have different values of $\hat{C}_a$. In this situation, the *k*-NN classifier implemented in the feature space with a better $\hat{C}_a$ value is expected to show more confidence in its decisions. For this reason, we propose the success confidence rate, $\hat{C}_a$, as the criterion to estimate the linear transformation matrix *C*.

2.3 Genetic optimisation

Since no closed-form method is known to optimise the proposed criterion, and since there is no tractable expression for its gradient, random numerical optimisation methods must be used.

The number of parameters to be estimated by the optimisation method is $m \times d$, with m being the number of LPs or new features to obtain, and with d being the dimensionality of the original data. If we want to project high-dimensional data, the number of parameters to estimate will be large. For this reason, we propose a GA as an appropriate paradigm to carry out the optimisation.

GAs have proven to be specially useful in large search spaces (Goldberg, 1989). We have used a GA with the following properties:

- An individual is composed of m chromosomes representing the m LPs to search. Each chromosome contains d genes, and each gene contains a binary string of b bits that encodes a coefficient of the LP in fixed point format.
- The fitness function is defined as the computed success confidence rate, $\hat{C}_a$, of a *k*-NN classifier trained with the projected data obtained from the LPs coded in the individual.
- The genetic selection scheme uses a rank-based strategy (Mitchell, 1996). In this strategy, the probability of being selected is computed from the rank position of the individuals. In our case, this method gave a faster convergence than a fitness-proportionate method.
- The following settings are used for the rest of the parameters: crossover probability is *0.6*, mutation probability is *0.001*, population size is *100* and the maximum number of generations is *300*.

Finally, since estimating the success confidence rate of a k-NN classifier is a time-consuming task, a fast neighbour search by means of kd-trees (Friedman et al., 1977) was implemented to reduce the computational cost. Additionally, a micro-grain parallel GA (Shyh-Chang et al., 1994) was implemented, allowing the use of several computers to compute individual fitness functions, obtaining a linear speedup.

We refer to the described method as Genetic Linear Projections (GLP).

3. Comparative study

3.1 Methodology

In this section, the GLP method is compared with the well-known PCA (linear, unsupervised), and the NDA by means of neural networks (non-linear, supervised). The comparison addresses both, feature extraction and data projection (mapping) applications. The three methods are applied to sixteen data sets in order to obtain different numbers of new features (see Table 1). Since the results obtained by NDA and GLP are not deterministic, for this methods five runs are performed with each parameter combination.

PCA obtains an eigenvector matrix (Φ) and an eigenvalue diagonal matrix (Δ) from the covariance matrix of the original data by means of a closed-form method. The columns of Φ correspond to orthonormal linear projections (eigenvectors) in the directions of maximal scatter. The values in the diagonal of Δ (eigenvalues) allow us to sort these directions depending on the scatter. To reduce a *d*-dimensional original space to an *m*-dimensional space, with $m<d$, we only have to keep the *m* eigenvectors with the largest eigenvalues.

The NDA method is based on training a two-hidden layer neural network. This is accomplished using the backpropagation algorithm with momentum, obtaining the new features from the response of the units of the second hidden layer. The number of units of the second hidden layer must be selected to equal the number of desired new features.

In order to detect possible overfitting problems with the three methods, each data set is split into a training set (70%) and a test set (30%). The methods are applied to the training sets, testing the performance of the obtained projections in the test sets. In order to estimate the success confidence rate, $\hat{C}_a$, a leaving-one-out procedure is employed in the training set for small data sets (less than *5000* patterns). A hold-out procedure is used with bigger data sets.

In the case of feature extraction, the performance of the methods is compared in terms of the success rate improvement obtained, as well as in terms of the reduction obtained in the number of features. Because the estimate of the success rate is obtained by error count (Duda & Hart, 1973), the 95% confidence intervals are provided to correctly compare the results.

For data projection purposes, the performance of these methods is first compared by means of visual judgement over the *2*-dimensional projections obtained from the data sets, and then by means of the success rate of a *k*-NN classifier computed for each data set in the original and projected spaces. This quantitative criterion gives us an idea of how well the class structure is preserved by the projections (Mao & Jain, 1995).

3.2 Corpora

The corpora are selected from well-known data sets from the UCI repository (Blake & Merz, 1998). A self-designed synthetic data set, *cookies*, is also used. This data set has been created to represent a well-known case in which PCA does not work well because the maximal scatter axes are not the most significant. This corpus consists of two *10*-dimensional normal distributions with covariance matrices

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.0001 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix},$$

and means μ_1=(+*0.1, 0, 0, …*), μ_1=(−*0.1, 0, 0, …*). Each class has *1000* patterns. These distributions represent two hyperspheres that are flattened (like cookies) in the dimension that separates them.

Table 1 summarises the features (size, dimensionality, number classes, ...) of every data set used.

Corpora	Size	Dim.	Classes	*k*	New features
german	1000	24	2	23	[1,2 - 20]
glass	214	9	6	3	[1,2 - 8]
cookies	2000	10	2	21	[1,2 - 10]
ionosphere	351	34	2	1	[1,2 - 30]
iris	151	4	3	15	[1,2,4]
digits	3000	196	10	3	[1,2 - 100]
bupa	345	6	2	23	[1,2 - 6]
pima	768	8	2	19	[1,2 - 8]
segment	2310	19	7	1	[1,2 - 15]
sonar	208	60	2	1	[1,2 - 60]
vehicle	846	18	4	3	[1,2 - 15]
wine	178	13	3	15	[1,2 - 10]
waveform	5000	21	3	27	[1,2 - 20]
page blocks	5473	10	5	3	[1,2 - 10]
sat	6435	36	6	5	[1,2 - 35]
musk	6598	166	2	3	[1,2 - 100]

Table 1. Data set features: size, dimensionality, number of classes, optimum *k* value for the *k*-NN classifier and a list of the new number of features searched for the different methods (values with dashes mean that several numbers in the interval have been searched for).

Corpora	Original		PCA		NDA		GLP	
german	0.7278 0.7285	(24)	0.7307 0.7086	(15)	***0.9882*** ***0.7183***	***(6)***	***0.8198*** ***0.7461***	***(6)***
glass	0.7308 0.5690	(9)	***0.7244*** ***0.6897***	***(4)***	0.7485 0.7255	(8)	0.8160 0.7843	(8)
cookies	0.4993 0.4728	(10)	0.4993 0.4729	(10)	***1.0000*** ***1.0000***	***(1)***	1.0000 1.0000	(2)
ionosphere	0.8833 0.8198	(34)	0.9083 0.8378	(8)	***1.0000*** ***0.9174***	***(1)***	0.9876 0.8899	(2)
iris	0.9725 0.9268	(4)	0.9725 0.9268	(4)	0.9818 0.9750	(4)	0.9909 0.9500	(4)
digits	0.9522 0.9504	(196)	***0.9603*** ***0.9559***	***(40)***	0.9948 0.9122	(15)	0.9581 0.9578	(100)
bupa	0.6901 0.5922	(6)	0.6901 0.5922	(6)	0.8092 0.7229	(6)	0.7557 0.6627	(6)
pima	0.7623 0.7014	(8)	0.7605 0.7330	(6)	0.8680 0.7352	(8)	***0.8272*** ***0.7589***	***(4)***
segment	0.9599 0.9607	(19)	0.9599 0.9607	(15)	***0.9876*** ***0.9757***	***(6)***	0.9826 0.9729	(10)
sonar	0.8392 0.8462	(60)	0.8531 0.8462	(10)	***1.0000*** ***0.7458***	***(1)***	1.0000 0.7966	(10)
vehicle	0.7141 0.7179	(18)	0.7059 0.7265	(15)	***0.9397*** ***0.8308***	***(6)***	0.8362 0.7594	(10)
wine	0.9590 0.9464	(13)	0.9836 0.9464	(6)	***1.0000*** ***0.9811***	***(1)***	1.0000 1.0000	(2)
waveform	0.8396 0.8531	(21)	***0.8541*** ***0.8720***	***(2)***	0.9429 0.8375	(10)	0.8649 0.8578	(15)
page blocks	0.9611 0.9685	(10)	0.9616 0.9685	(8)	***0.9746*** ***0.9685***	***(4)***	0.9650 0.9710	(8)
sat	0.9092 0.8965	(36)	0.9116 0.8944	(15)	***0.9555*** ***0.8944***	***(10)***	0.9107 0.8991	(25)
musk	0.9676 0.9641	(166)	0.9673 0.9656	(90)	***1.0000*** ***0.9946***	***(4)***	0.9783 0.9759	(15)

Table 2. The best success rates obtained by PCA, GLP and NDA on the training set (top) and the test set (bottom). The results on the original feature space are also shown. The values in brackets represent the number of features. Values in boldface represent the methods that obtain the highest reduction, maintaining or improving the original correct classification rate.

3.3 Results

Tables 2 and 3 present the best runs obtained for the three methods. The success rate is presented instead of the success confidence rate (the criterion used to optimise) because we are interested in the final classifier performance. Analysing the results and considering their 95% confidence intervals (see Table 3) it can be observe that a significant classifier improvement was only obtained by NDA in three data sets (*cookies*, *vehicle* and *musk*), and by GLP in one data set (*cookies*). In all the other cases, only feature reduction was achieved, i.e., the classifier obtained similar results to the original space but with fewer features.

By analysing all the runs (not only the best ones), and taking into account the confidence intervals, it can be observed that the three methods obtained a similar reduction with the exception of PCA, which obtained a worse reduction in five data sets (*cookies*, *ionosphere*, *segment*, *sonar* and *vehicle*). For instance, Figure 1 shows the results for the *vehicle*, *cookies* and *segment* data sets, for different numbers of features. It also shows that a similar reduction was obtained by NDA and GLP in these cases, while PCA yielded significantly worse results.

It is interesting to note that, although only one linear projection was enough to separate the classes of *cookies* data set, PCA and GLP had problems. PCA was not able to do it because the maximal scatter direction was not the optimal in this case. GLP failed because random optimisation methods have problems finding very isolated solutions. Nevertheless GLP was able to find a good solution with two or more linear projections while PCA continued to fail.

In the data projection context, looking at the success rate obtained by the classifiers when projecting data sets into a 2-dimensional space (see Table 4), it can be observed that NDA and GLP outperformed PCA in most of the data sets. NDA obtained better results than GLP for high dimensional data sets (i.e. the *digits* data set, see Figure 2). Visual analysis of obtained projections confirmed these results showing that GLP and NDA produced less overlapping views than PCA (see Figures 2, 3 and 4).

Finally, with respect to the time complexity of methods, although the estimation of the success confidence rate, $\hat{C}_a$, was optimised by using *kd-trees*, and although the GA was parallelised to speed up the algorithm, the *off line* cost of GLP was higher than the cost for the other two methods. The method with the lowest cost is LDA because the transformations are obtained by means of a closed-form method and there is no need for several runs as in NDA or GLP.

Regarding the *on line* costs, GLP and LDA generate linear transformations and have an application cost that is lower than the application of the non-linear transformations generated by the neural network on NDA.

4. Conclusions

From the results obtained, we can conclude that although NDA obtains good results with non-linear projections in all data sets, similar results can be obtained using GLP in most of them. This indicates to us that, in practice, linear projections can obtain results just as good as non-linear projections in most cases. Even though PCA employs linear projections as well, it performs worse in some data sets probably because it is an unsupervised method. Classical linear, supervised feature extraction methods like LDA have important limitations: first, the number of new features is limited by the number of classes; and second, numerical problems arise when working with high dimensional or small data sets, restricting its use.

The proposed GLP method does not have these limitations. The main drawback of the GLP method is it computational cost; however, this is an *off line* process. Once the

Corpora	Original		PCA		NDA		GLP	
german	0.7278	0.6925 0.7598	0.7307	0.6955 0.7626	***0.9882***	***0.9776*** ***0.9951***	***0.8198***	***0.7895*** ***0.8478***
	0.7285	0.6760 0.7794	0.7086	0.6551 0.7607	0.7183	0.6620 0.7670	0.7461	0.6935 0.7949
glass	0.7308	0.6529 0.8008	0.7244	0.6529 0.8008	0.7485	0.6743 0.8187	0.8160	0.7474 0.8771
	0.5690	0.4482 0.7006	0.6897	0.5594 0.7976	0.7255	0.6091 0.8370	0.7843	0.6603 0.8749
cookies	0.4993	0.4728 0.5258	0.4993	0.4728 0.5258	***1.0000***	***0.9974*** ***1.0000***	***1.0000***	***0.9974*** ***1.0000***
	0.4728	0.4328 0.5142	0.4729	0.4328 0.5142	***1.0000***	***0.9939*** ***1.0000***	***1.0000***	***0.9939*** ***1.0000***
ionosphere	0.8833	0.8391 0.9227	0.9083	0.8672 0.9429	***1.0000***	***0.9851*** ***1.0000***	***0.9876***	***0.9646*** ***0.9975***
	0.8198	0.7319 0.8874	0.8378	0.7535 0.9028	0.9174	0.8435 0.9601	0.8899	0.8089 0.9395
iris	0.9725	0.9329 0.9977	0.9725	0.9329 0.9977	0.9818	0.9481 0.9998	0.9909	0.9655 1.0000
	0.9268	0.8173 0.9860	0.9268	0.8173 0.9860	0.9750	0.8823 0.9994	0.9500	0.8485 0.9946
digits	0.9522	0.9424 0.9611	0.9603	0.9512 0.9684	***0.9948***	***0.9906*** ***0.9974***	0.9581	0.9486 0.9663
	0.9504	0.9337 0.9633	0.9559	0.9400 0.9681	***0.9122***	***0.8918*** ***0.9299***	0.9578	0.9425 0.9700
bupa	0.6901	0.6305 0.7506	0.6901	0.6305 0.7506	***0.8092***	***0.7537*** ***0.8567***	0.7557	0.7003 0.8119
	0.5922	0.4910 0.6880	0.5922	0.4910 0.6880	0.7229	0.6316 0.8112	0.6627	0.5703 0.7594
pima	0.7623	0.7252 0.7988	0.7605	0.7233 0.7971	***0.8680***	***0.8382*** ***0.8970***	0.8272	0.7941 0.8596
	0.7014	0.6408 0.7625	0.7330	0.6728 0.7906	0.7352	0.6728 0.7906	0.7589	0.7004 0.8145
segment	0.9599	0.9490 0.9688	0.9599	0.9490 0.9688	***0.9876***	***0.9810*** ***0.9924***	***0.9826***	***0.9751*** ***0.9885***
	0.9607	0.9438 0.9742	0.9607	0.9438 0.9742	0.9757	0.9610 0.9856	0.9729	0.9575 0.9834
sonar	0.8392	0.7716 0.8967	0.8531	0.7872 0.9081	***1.0000***	***0.9749*** ***1.0000***	***1.0000***	***0.9749*** ***1.0000***
	0.8462	0.7233 0.9198	0.8462	0.7233 0.9198	0.7458	0.6150 0.8447	0.7966	0.6682 0.8834
vehicle	0.7141	0.6763 0.7506	0.7059	0.6676 0.7425	***0.9397***	***0.9168*** ***0.9570***	***0.8362***	***0.8038*** ***0.8651***
	0.7179	0.6597 0.7738	0.7265	0.6680 0.7812	***0.8308***	***0.7823*** ***0.8777***	0.7594	0.7055 0.8139
wine	0.9590	0.9084 0.9868	0.9836	0.9430 0.9980	1.0000	0.9707 1.0000	1.0000	0.9707 1.0000
	0.9464	0.8434 0.9882	0.9464	0.8434 0.9882	0.9811	0.8993 0.9995	1.0000	0.9328 1.0000
waveform	0.8396	0.8271 0.8517	0.8541	0.8419 0.8655	***0.9429***	***0.9346*** ***0.9503***	***0.8649***	***0.8531*** ***0.8760***
	0.8531	0.8344 0.8709	0.8720	0.8540 0.8885	0.8375	0.8177 0.8557	0.8578	0.8393 0.8753
page blocks	0.9611	0.9545 0.9670	0.9616	0.9551 0.9675	***0.9746***	***0.9692*** ***0.9794***	0.9650	0.9587 0.9706
	0.9685	0.9593 0.9768	0.9685	0.9593 0.9768	0.9685	0.9593 0.9768	0.9710	0.9621 0.9789
sat	0.9092	0.9004 0.9174	0.9116	0.9030 0.9198	***0.9555***	***0.9492*** ***0.9614***	0.9107	0.9020 0.9189
	0.8965	0.8825 0.9101	0.8944	0.8803 0.9082	0.8944	0.8803 0.9082	0.8991	0.8852 0.9125
musk	0.9676	0.9620 0.9724	0.9673	0.9618 0.9722	***1.0000***	***0.9992*** ***1.0000***	***0.9783***	***0.9737*** ***0.9823***
	0.9641	0.9550 0.9719	0.9656	0.9566 0.9732	***0.9946***	***0.9901*** ***0.9972***	0.9759	0.9680 0.9821

Table 3. The bests success rates obtained by PCA, GLP and NDA on the training set (top) and the test set (bottom). The results on the original feature space are also shown. Small values represent the 95% confidence intervals for the correct classification rate. Values in boldface represent values that are significantly different from the original.

transformations are computed, the cost of applying them to new data is lower than applying the neural network trained by the NDA method. Moreover, the process of training an NDA neural network is not straightforward in many cases, having convergence problems.

From the point of view of data projection, it can be concluded that NDA projections outperform our GLP method when the intrinsic dimensionality is high. In these cases, the NDA projection is able to obtain a good view of the class structure even in a 2-dimensional projection. Nevertheless, we consider that NDA has one important drawback. Because non-linear transformations are used, an important distortion of the original space occurs, especially when projecting into a 2-dimensional space in an attempt to preserve the class structure (see Figure 3). In this situation, a synthetic view of the configuration of real clusters is obtained. The GLP method uses linear transformations, thereby producing less distorted and more meaningful views of the original space (distortion can appear because the new axes are not necessarily orthogonal). The PCA method is linear and unsupervised; therefore, the projections computed do not always show a good view of the class structure if the discriminant axes are not the ones with the highest variance.

Corpora	Original	PCA	NDA	GLP
german	0.7278	0.7178	***0.9075***	0.7885
glass	0.7308	0.6731	0.6454	***0.6626***
cookies	0.4993	0.3986	***1.0000***	***0.9959***
ionosphere	0.8833	0.7125	***0.9901***	0.9769
iris	0.9725	0.9266	0.9709	***0.9745***
digits	0.9522	0.4364	***0.8508***	0.6336
bupa	0.6901	0.5331	***0.7756***	0.7206
pima	0.7623	0.7130	***0.8586***	0.7825
segment	0.9599	0.6402	***0.9412***	0.9135
sonar	0.8392	0.5664	***0.9879***	0.9289
vehicle	0.7141	0.4935	***0.7921***	0.7438
wine	0.9590	0.9508	***0.9968***	***0.9936***
waveform	0.8396	0.8541	***0.8909***	0.8514
page blocks	0.9611	0.9369	***0.9641***	0.9530
sat	0.9092	0.8322	***0.8756***	0.8380
musk	0.9676	0.8913	***0.9993***	0.9296

Table 4. Mean values for the correct classification rate obtained over the training sets when looking for two new features (exploratory analysis). The bests results for each data set are in boldface.

5. References

Aladjem, M. (1991). Parametric and nonparametric linear mappings of multidimensional data. *Pattern Recognition*, 24, 6, pp 534–551.

Blake, C. & Merz, C. (1998). UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. University of California, Irvine.

Cardoso, J. and Comon, P. (1996). Independent component analysis, a survey of algebraic methods. *Proceedings of ISCAS-96*, Vol. 2, pp 93–96.

Devijver, P. and Kittler, J. (1982), *Pattern Recognition: A Statistical Aproach*. Prentice Hall, London.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York.

Friedman, J.; Bentley, J. & Finkel, R. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathemathical Software*, 3, pp 209–226.

Fukunaga, K. (1990). *Statistical Pattern Recognition*. Academic Press, second edition edition.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Tabor. The University of Michigan Press.

Kohonen, T. (1990). The self-organizing map. *Proceedings IEEE*, 78, 9, pp 1464–1480.

Lerner, B.; Guterman, H.; Aladjem, M.; Dinstein, I. & Romen, Y. (1998). On pattern classification with sammon's nonlinear mapping (an experimental study). *Pattern Recognition*, 31, 4, pp 371 – 381.

Liu, H. & Motoda, H. (eds.). (1998). *Feature Extraction, Construction and Selection. A Data Mining Perspective*. Kluwer Academic Publishers.

Mao, J. & Jain, A. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6, 2.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA.

Sammon, J. (1969). A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18, 5.

Shyh-Chang, L.; Punch, W. & Goodman, E. (1994). Coarse-grain parallel genetic algorithm: categorization and a new approach. *6th IEEE Symposium on Parallel and Distributed Processing*, pp 28 – 37.

Siedlecki, W.; Siedlecka, K. & Sklansky, J. (1988). An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, 21, 5, pp 411 – 429.

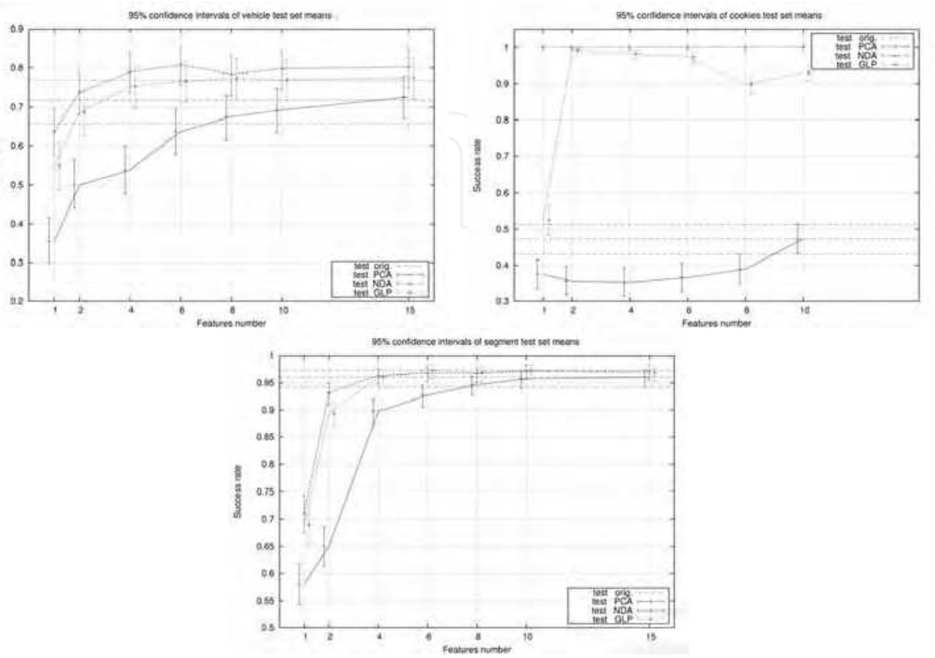

Fig. 1. Correct classification rate results for *vehicle* (top), *cookies* (middle) and *segment* (bottom) data set. The 95% confidence intervals are shown.

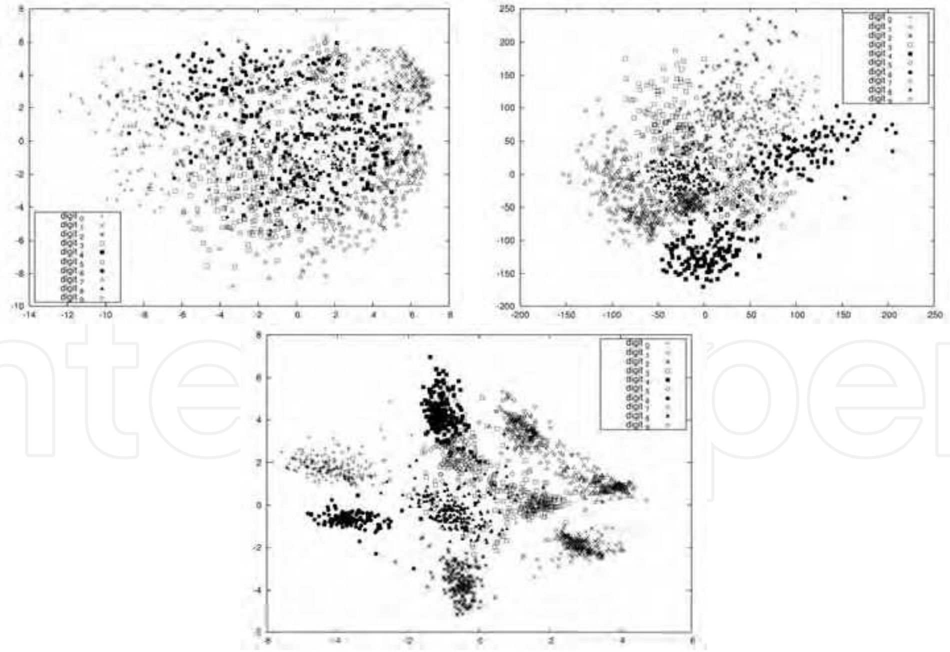

Fig. 2. Projections obtained for the *digits* data set by PCA (top), GLP (middle) and NDA (bottom).

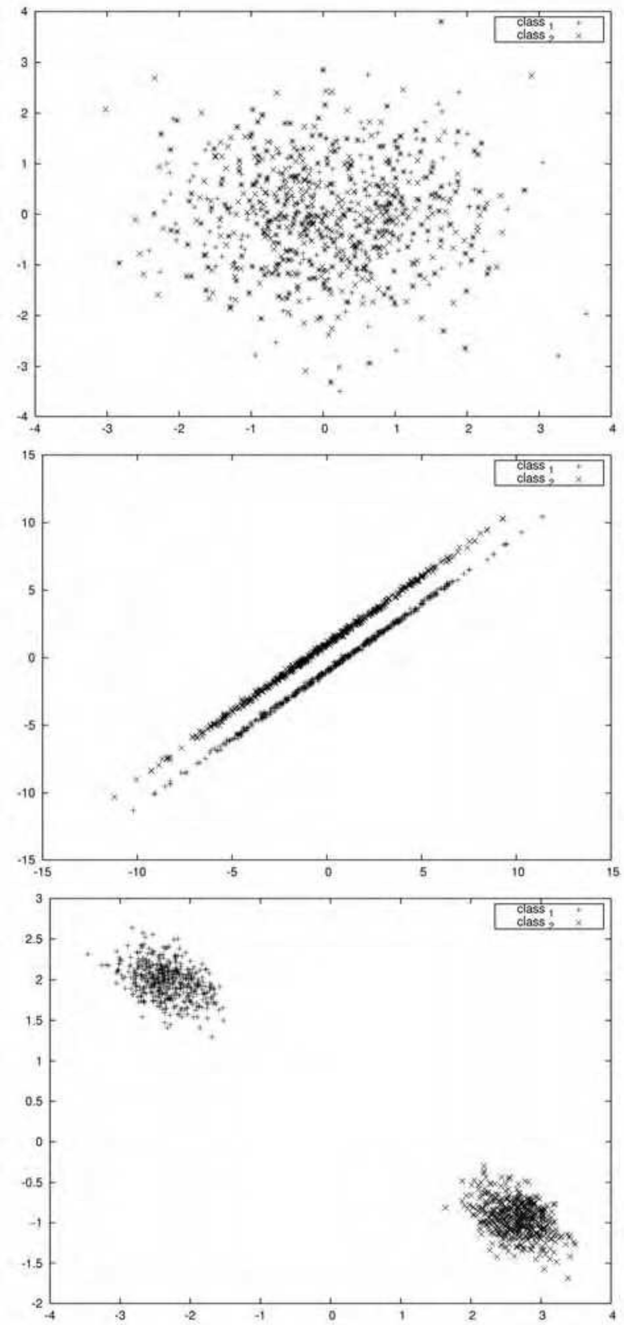

Fig. 3. Projections obtained for the *cookies* data set by PCA (top), GLP (middle) and NDA (bottom).

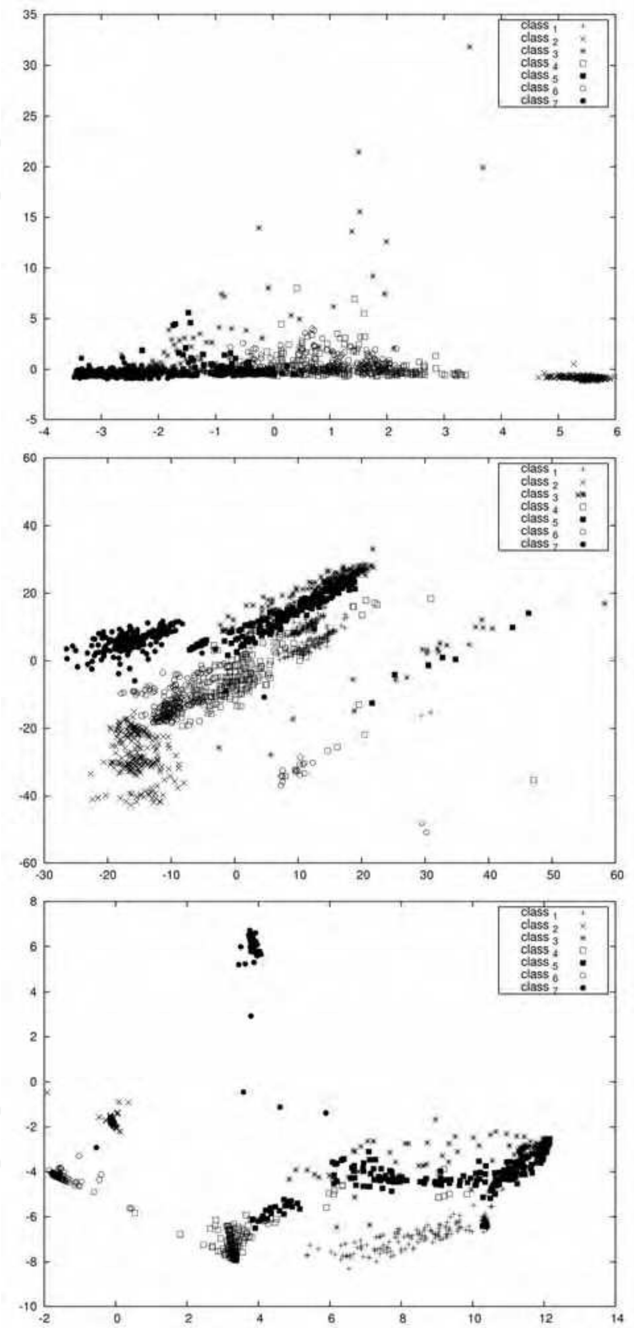

Fig. 4. Projections obtained for the *segment* data set by PCA (top), GLP (middle) and NDA (bottom).

Vision Systems: Segmentation and Pattern Recognition
Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-05-9
Hard cover, 536 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

Research in computer vision has exponentially increased in the last two decades due to the availability of cheap cameras and fast processors. This increase has also been accompanied by a blurring of the boundaries between the different applications of vision, making it truly interdisciplinary. In this book we have attempted to put together state-of-the-art research and developments in segmentation and pattern recognition. The first nine chapters on segmentation deal with advanced algorithms and models, and various applications of segmentation in robot path planning, human face tracking, etc. The later chapters are devoted to pattern recognition and covers diverse topics ranging from biological image analysis, remote sensing, text recognition, advanced filter design for data analysis, etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821