RECURSIVE DISCRIMINANT REGRESSION ANALYSIS TO FIND HOMOGENEOUS GROUPS

ESTEBAN GARCÍA-CUESTA
Physics Department, University Carlos III, Av. Universidad 30
Leganés, Madrid 28911, Spain[*]
esteban.garcia@uc3m.es

INÉS M. GALVÁN
Computer Science Department, University Carlos III, Av. Universidad 30
Leganés, Madrid 28911, Spain
inesmaria.galvan@uc3m.es

ANTONIO J. DE CASTRO
Physics Department, University Carlos III, Av. Universidad 30
Leganés, Madrid 28911, Spain
decastro@fis.uc3m.es

The main motivation of this paper is to propose a method to extract the output structure and find the input data manifold that best represents that output structure in a multivariate regression problem. A graph similarity viewpoint is used to develop an algorithm based on LDA, and to find out different output models which are learned as an input subspace. The main novelty of the algorithm is related with finding different structured groups and apply different models to fit better those structures. Finally, the proposed method is applied to a real remote sensing retrieval problem where we want to recover the physical parameters from a spectrum of energy.

Keywords: Dimensionality reduction; local regression; LDA; supervised learning; density clustering.

1. Introduction

In regression or classification problems a dimensionality reduction may be needed whenever there are high dimensional datasets. That reduction searches for the variables or a combination of them which best preserves its intrinsic information. Principal Component Analysis (PCA)[1] is a standard linear dimensionality reduction technique that performs a dimensionality reduction by projecting the original data onto the linear subspace spanned by the first l eigenvectors. It is optimal for Gaussian distributed classes and captures the directions of maximum variance in the data using the correlation or covariance matrix.

An important drawback of the technique is how to choose the number of dimensions. Often, an accumulative variance criteria is used and there also have been proposed other methods based on probabilistic principal components where the number of dimensions are learned.[2] Besides of this, PCA is a very powerful unsupervised dimensionality reduction tool and has been used widely, for instance in computer vision.[3–8] It is unsupervised because it only uses the input data for the analysis. Otherwise, when the input and output data are used during the dimensionality reduction process it is called supervised. Linear discriminant analysis (LDA)[9] is the supervised version of PCA and is also widely used in the dimensionality reduction context.[10]

On the other hand, kernel methods[11] have proved to be extremely powerful in many areas of machine

[*] Departamento de Fìsica Universidad Carlos III de Madrid, Av. Universidad 30, 28911 Leganés-Madrid, Spain.

learning, and the so-called "kernel trick" is by now widely appreciated. Many dimensionality reduction algorithms, as PCA, can be reformulated in terms of Gram matrices, and generalized to nonlinear problems by substituting a kernel function for the inner product. But, the Gram matrices not only can be used in algorithms taking the advantage of the "kernel trick" but also can be interpreted as a similarity pairwise matrix. Spectral clustering[12] is based on this notion of similarity which can be defined by similarity graphs and it is able to find arbitrarily shaped data groups.

In this paper we want to transform a global regression problem in n regression sub-problems using the output data to find out different models. A recursive model division approach has been adopted to find out the different models associated to different local groups. This approach improves the estimation capabilities comparing with a non-recursive one.[13] The recursivity allows to get a better insight into the structures when there exist a hierarchy between the different models. Therefore, a different manifold is learnt during each one of the recursive steps until the number of clusters desired is found. Afterwards, different regression models are applied for each new group of data.

2. Computational Analysis

The goal of LDA[4] is to maximize the between-class measure while minimizing the within-class measure. The objective function for multiple classes can be described by

$$\ell = \max_{A} \frac{\mathrm{tr}(\mathbf{A}^T \mathbf{S}_b \mathbf{A})}{\mathrm{tr}(\mathbf{A}^T \mathbf{S}_t \mathbf{A})} \tag{1}$$

where $\mathbf{S}_b$ is the between scatter matrix, $\mathbf{S}_t$ is the total scatter matrix, tr() denotes the matrix trace, and $\mathbf{A}$ is a matrix with projections functions in its columns.

We are going to analyze LDA from a graph viewpoint likewise in Ref. 14. Assuming that the input data is centered $\overline{\mathbf{X}} = \mathbf{X} - \mu$, then

$$\begin{aligned} \mathbf{S}_b &= \sum_{k=1}^{c} n_k \left(\frac{1}{n_k} \sum_{j=1}^{n_k} \overline{\mathbf{x}}_j^k \right) \left(\frac{1}{n_k} \sum_{j=1}^{n_k} \overline{\mathbf{x}}_j^k \right)^T \\ &= \sum_{k=1}^{c} \overline{\mathbf{X}}^k \mathbf{R}^k (\overline{\mathbf{X}}^k)^T = \overline{\mathbf{X}} \mathbf{R} \overline{\mathbf{X}}^T \end{aligned} \tag{2}$$

where n_k is the number of elements of class k, c is the number of classes, and $\mathbf{R}^k$ is a $n_k \times n_k$ structured matrix with all elements equal to $1/n_k$. If we assume that the classes k are ordered in the diagonal part of the R matrix then,

$$\mathbf{R} = \begin{pmatrix} R^1 & & & \\ & R^2 & & \\ & & \ddots & \\ & & & R^c \end{pmatrix}.$$

In our regression application, we do not know what is that output structure, and we have to deal with that unknown information. Since we do not know c, we do not know the number of elements in each class n_k either, and therefore the R matrix is unknown.

We propose to use the output structure information instead of classes for this purpose. The output structure information is defined as the similarity Gram matrix. In regression frameworks, this structure is very appealing because is going to allow to do clustering in the output structure based on the pairwise similarity, and therefore preserving homogeneity in the models. To introduce this approach we are going to use the graph Laplacians properties and a kernel approach.

The relation between spectral clustering based on graph Laplacians and the kernel approach relies on the fact that the smallest eigenvectors of graph Laplacians can also be interpreted as the largest eigenvectors of kernel matrices (Gram matrices).

The unnormalized graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} \in \Re^{n \times n}$ is a diagonal matrix such that each entry is the sum of the rows of $\mathbf{W}$, i.e. $d_{ii} = \sum_j w_{ij}$, and $\mathbf{W}$ is the similarity matrix. An overview of its properties can be found in Ref. 15. One important property for spectral clustering is the following: (*Number of connected components and the spectrum of* L) *Let* G *be an undirected graph with non-negative weights. Then the multiplicity* k *of the eigenvalue* 0 *of* L *equals the number of connected components* $A_1, \ldots, A_k$ *in the graph. The eigenspace of eigenvalue* 0 *is spanned by the indicator vectors* $1_{A_1}, \ldots, 1_{A_k}$ *of those components.*

Therefore if we consider a graph of k connected components and without loss of generality, we can assume that the vertices are ordered according to the connected components they belong to, and the matrix $\mathbf{W}$ and $\mathbf{L}$ have a block diagonal form as $\mathbf{R}$.

Because we want to use the output structure information to cluster the data, we calculate the kernel similarity output matrix as $\mathbf{K}_{yy} = \Phi(\mathbf{Y})\Phi(\mathbf{Y})^T$ where $\Phi(\mathbf{Y})$ is only defined in the feature space $\mathbf{F}$, and $\mathbf{Y} \in \Re^{s \times n}$ is the output matrix data where s is the number of features and n the number of samples.

The main freedom of using the kernel approach lies in choosing the kernel function $K(x,y)$, or otherwise specifying the kernel matrix $\mathbf{K}_{ij}$. Some widely used kernels are the linear, polynomial and Gaussian kernels, given by: $K(x,y) = x \cdot y, K(x,y) = (1 + x \cdot y)^p, K(x,y) = e^{\frac{-|x-y|^2}{2\sigma^2}}$. For this study we have used a linear kernel.

If we compare the hypothetical diagonal matrix $\mathbf{K}_{yy}$ that contains the Gram structure of the output, and the one obtained by the computational analysis of LDA at Eq. 2 it can be observed that $\mathbf{R}$ matrix can be decomposed as $\mathbf{R}^k = \mathbf{K}_{yy}^k \cdot \mathbf{D}^k$ where $\mathbf{D}^k$ is a diagonal block matrix containing the normalization values for each block k. Then $\mathbf{S}_b = \overline{\mathbf{X}}(\mathbf{K}_{yy} \cdot \mathbf{D})\overline{\mathbf{X}}^T$, and the within scatter matrix is reformulated as $S_w = S_t - S_b = \overline{\mathbf{X}}(I - (\mathbf{K}_{yy} \cdot \mathbf{D}))\overline{\mathbf{X}}^T$. It can be derived that the new total scatter matrix is equal to the total covariance matrix in the input space $S_t = \overline{\mathbf{X}}(\mathbf{K}_{yy} \cdot \mathbf{D})\overline{\mathbf{X}}^T + \overline{\mathbf{X}}(I - (\mathbf{K}_{yy} \cdot \mathbf{D}))\overline{\mathbf{X}}^T = \overline{\mathbf{X}}\,\overline{\mathbf{X}}^T$.

With this approach the optimization problem defined in Eq. (1) can be transformed

$$\ell = \max_A \frac{\mathrm{tr}(\mathbf{A}^T\overline{\mathbf{X}}\mathbf{K}_{yy}\overline{\mathbf{X}}^T\mathbf{A})}{\mathrm{tr}(\mathbf{A}^T\overline{\mathbf{X}}\,\overline{\mathbf{X}}^T\mathbf{A})}, \tag{3}$$

where the new ratio of the between and within scatter matrices is the straightforward result of applying the Gram matrix instead of a matrix containing the proportional part of the number of elements associated to each class k.

3. Recursive Discriminant Regression Analysis

A recursive partition approach has been adopted in order to find the different models in the new low dimensional space $\mathbf{A}$. As it is shown in the Fig. 1, this approach split the samples into two different subsets iteratively using the projection of the samples ($\mathbf{P} = \mathbf{A}^T\overline{\mathbf{X}}$) and a density based clustering algorithm.

Then a new estimation model (M1/Model 1) is calculated for the subset with the lowest deviation $\mathbf{X1}$, and the other subset $\mathbf{X2} = \{\mathbf{X} - \mathbf{X1}\}$ is used for the next iteration of the analysis. Afterwards, the process is repeated using $\mathbf{X4} = \{\mathbf{X2} - \mathbf{X3}\}$ being M3 a new model and so on. The main advantage of this recursivity is that it allows to separate those samples that have been already classified as homogeneous from the other ones, and to get a better insight into the last ones during the next analysis.

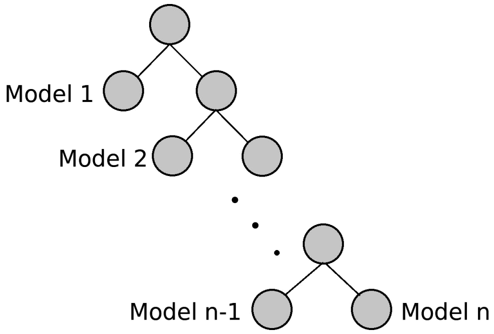


Fig. 1. Recursive groups models.

In the next, we explain the proposed algorithm. Given a set of input/output data $\mathbf{X} \in \Re^{p \times n}$, and $\mathbf{Y} \in \Re^{s \times n}$ with n samples, and each one of the samples with input dimension p, and output dimension s, the algorithmic procedure is:

(1) Construct the output Gram matrix $\mathbf{K}_{yy} \in \Re^{n \times n}$.

(2) Solve the generalized eigenproblem of (3) using the following steps:

Calculate the SVD (singular value decomposition) of $\overline{\mathbf{X}}$, $\overline{\mathbf{X}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathbf{T}}$, where $\mathbf{s}_1, \ldots, \mathbf{s}_r$ are the singular values associated to the left and right eigenvectors $\mathbf{U}$, and $\mathbf{V}$.

Then we can transform the Eq. (3) substituting $\overline{\mathbf{X}}$ by its SVD decomposition, as $\max_A \frac{\mathrm{tr}(\mathbf{A}^T\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{K}_{\mathbf{yy}}\mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{A})}{\mathrm{tr}(\mathbf{A}^T\mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{A})}$, and being $\mathbf{B} = \mathbf{S}\mathbf{U}^T\mathbf{A}$ we get the next optimization problem, $\max_B \frac{\mathrm{tr}(\mathbf{B}^T\mathbf{V}^T\mathbf{K}_{yy}\mathbf{V}\mathbf{B})}{\mathrm{tr}(\mathbf{B}^T\mathbf{B})}$ that can be solved as an eigenproblem on $\mathbf{B}$ and then compute $\mathbf{A}$ as $\mathbf{A} = \mathbf{U}\mathbf{S}^{-1}\mathbf{B}$.

(3) Compute the projected data on the new axes as, $\mathbf{P} = \mathbf{A}^T\overline{\mathbf{X}}$.

(4) Apply a density based clustering algorithm on the first l components of the projected data $\mathbf{P}$ obtaining a cluster $\mathbf{X1} \in \Re^{\mathbf{p} \times \mathbf{q}}$ associated to a model M1, and other data subset $\mathbf{X2} = \{\mathbf{X} - \mathbf{X1}\} \in \Re^{p \times n-q}$. This step looks for a two clustering division adjusting gradually the density algorithm parameters pursuing that purpose.

(5) While the number of found clusters is minor than desired return to step 1 using $\mathbf{X2} \in \Re^{p\times n-q}$ and $\mathbf{Y2} \in \Re^{s\times n-q}$ subsets as inputs.

(6) Apply a regression model (M_i) to each one of the obtained groups $\mathbf{X}_i$.

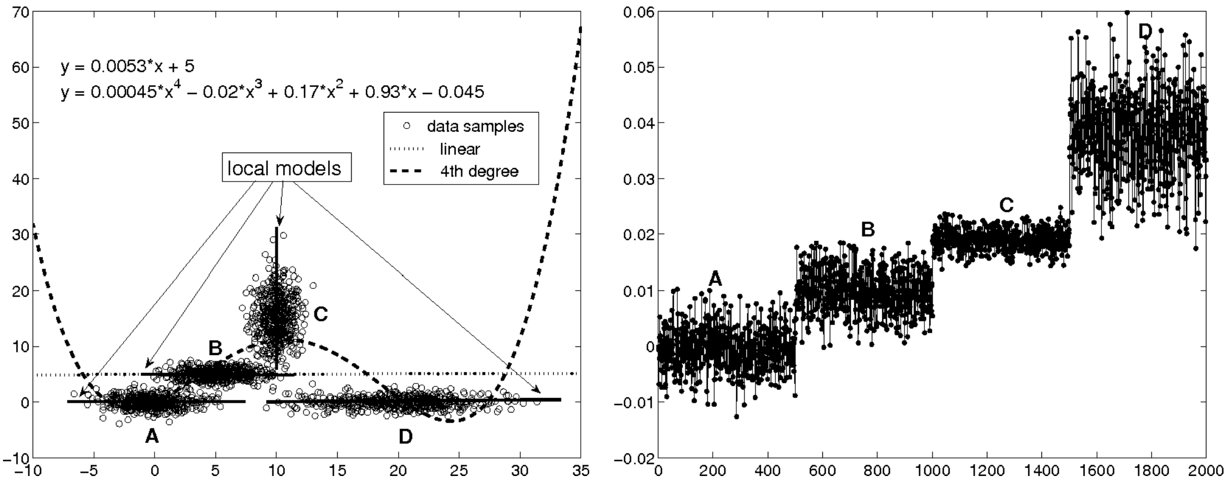
3.1. *Artificial example*

We have created an artificial example which is composed by four different groups of random values drawn from a normal distribution. The data sets have been centered in $(0, 0)$, $(5, 5)$, $(10, 15)$, $(20, 0)$, and their standard deviation are $(2, 1)$, $(2, 1)$, $(1, 4)$, $(4, 1)$ as it is shown in Fig. 2. The polynomial fitting of degrees 1 and 4 has been added for comparison purposes. We want to point out that the linear fitting is very vague and has a poor accuracy. As the polynomial degree is increased a better fitting can be found but also a more complex model has to be build which eventually could lead to overfitting. The adjustment of degree 4 is still vague and although improves the linear model it is very difficult to find a good model for the chosen dataset.

The Fig. 2(b) shows the projection (**P**) of the same dataset onto the subspace (**A**) found by the proposed algorithm. It can be seen how the different groups have their own center of mass. In a first step the group $\{D\}$ can be discovered obtaining two groups: $\{A, B, C\}$ and $\{D\}$, and a lineal model is assign to the group $\{D\}$. During the next step of the recursivity the group $\{C\}$ is discovered and it is also modeled by a linear approximator having the dataset $\{A, B\}$ as the last group to analyze. Doing the same process the groups $\{A\}$ and $\{B\}$ are discovered assigning to them their corresponding lineal model. Finally we obtain four lineal models which fit the different data samples ranges. In the Fig. 2(a) the different local models are drawn with solid lines. These models fit very well each one of the different groups and therefore are able to find a good fitting to the complete dataset.

4. Application to a Combustion Temperature Estimation Scenario

We have applied the explained algorithm to a combustion scenario which is related with the inversion of the radiative transfer equation (RTE). This inversion is a challenging mathematical problem, it is ill-posed and it also has multicollinearity problems. Given the measurements of energy at different wavenumbers represented by $\mathbf{X} \in \Re^{p\times n}$, and the output data associated to the temperature profile represented by $\mathbf{Y} \in \Re^{s\times n}$, we would like to learn a mapping f such that $\mathbf{Y} = f(\mathbf{X})$. Recall that each column of $\mathbf{X}$ and $\mathbf{Y}$ corresponds to a different observation. For instance, each sample $\mathbf{x}_i$ is a spectrum of radiance in the infrared range of $2110\,\mathrm{cm}^{-1} - 2410\,\mathrm{cm}^{-1}$ with $p = 2341$. Likewise, each sample $\mathbf{y}_i$ is the corresponding temperature profile with $s = 200$.



(a)

(b)

Fig. 2. (a) Global model versus Local models fitting for four different groups of normal distributed data. (b) Projection of dataset samples.

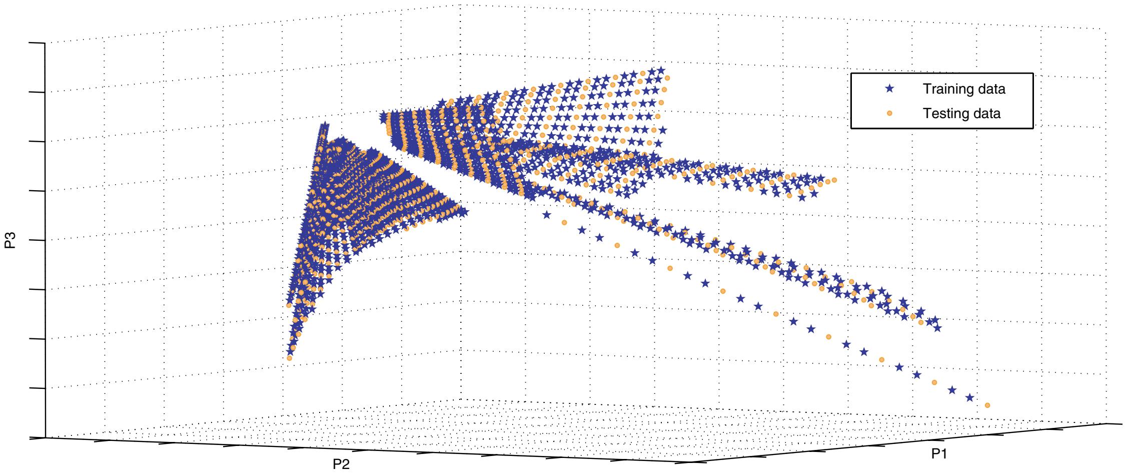


Fig. 3. Training and testing data projected into the first three components.

The dataset used in this study has been generated with a simulator based on the experimental database HITRAN[16] and the parameter ranges used to generate this dataset are based on typical combustion environment conditions (see Ref. 17 for more detailed information).

The Fig. 3 shows the first three dimensions of training $\mathbf{P}_{\mathrm{train}}$ and testing $\mathbf{P}_{\mathrm{test}}$ projections. It can be seen in this figure that the test data its almost perfectly embedded into the learnt manifold $\mathbf{A}$. Therefore we have chosen a simple classification technique as k-nearest neighbors with $k = 1$. Then, if two samples are close in the lower subspace they are considered to belong to the same model. That is, each testing sample is associated with the model of its nearest neighbor of the training dataset.

The different clusters do not have a uniform structure and they are characterize by areas of high density, separated by others that are empty or are noisy. In order to cluster this type of data, we have used the well known density clustering algorithm DBSCAN[18] due to its simplicity and speed.

The proposed recursive approach allows to get the different structures during the different iterations of the analysis. Table 1 shows the standard deviations for each group during the first six iterations. It is also indicated the different models which are associated to the lowest standard deviation.

To measure the success of the classification process we have split the data intro training and testing samples and used a 1-nearest neighbor classifier. The classification ratio of success obtained has been of 98.8% which can be considered as very high.

To show the improvements between a global model and the approach suggested in this study we have done six different experiments. Each experiment corresponds with different data sets associated to each one of the clusters discovered. Then we have applied a regression model to estimate each one of these data sets. Because the similarity criteria used

Table 1. Standard deviation of the different clusters found during the recursive partition process.

Iteration	Cluster no.	Standard deviation	Model no.	Iteration	Cluster no.	Standard deviation	Model no.
1	1	0.014	M1	4	1	0.012	M4
	2	0.016			2	0.018	
2	1	0.013	M2	5	1	0.018	M5
	2	0.016			2	0.020	M6
3	1	0.010	M3				
	2	0.017					

Table 2. Mean Absolute Error per sample(MAEs) of temperature, and its standard deviation (SD).

Method	Cluster no.	Temperature test (MAEs/SD)K	Temperature train(MAEs/SD)K
Global Model	Cluster 1(MLP)	2.45/1.40	0.75/0.50
	Cluster 2(MLP)	2.18/1.62	0.76/0.52
	Cluster 3(MLP)	1.49/0.93	0.88/0.68
	Cluster 4(MLP)	1.21/0.90	0.77/0.51
	Cluster 5(MLP)	1.10/1.54	0.76/0.47
	Cluster 6(MLP)	0.80/1.33	0.78/0.45
Model 1 (M1)	Cluster 1(linear)	0.12/0.25	4E-6/4E-6
Model 2 (M2)	Cluster 2(linear)	0.17/0.33	5E-6/4E-6
Model 3 (M3)	Cluster 3(linear)	0.10/0.17	9E-6/8E-6
Model 4 (M4)	Cluster 4(linear)	0.17/0.32	1E-4/9E-5
Model 5 (M5)	Cluster 5(linear)	0.19/0.87	6E-4/5E-4
Model 6 (M6)	Cluster 6(linear)	1.21/0.97	1.15/0.80

in the algorithm has been linear we use a linear model as estimator. In the global approach a multilayer perceptron (MLP) has been used as estimator. The architecture of the MLP is one hidden layer, and the number of neurons is fixed using a greedy approach (30 neurons). This results are indicated in the Table 2 which shows the Mean Absolute Error profile per sample (MAEs). The obtained MAEs in temperature is computed as MAEs $= \frac{1}{z}\frac{1}{n}\sum_{k=1}^{z}\sum_{j=1}^{n}|\mathbf{y}_{kj} - \hat{\mathbf{y}}_{kj}|$ where z is the discretized length, and n the number of samples. The MAEs gives an idea of the physical error.

The table is divided in two rows, and each one in another six which correspond to the number of clusters discovered by our algorithm. The first row from above shows the results of the global model per each one of the clusters, and the second row shows separately the error for each one of the models.

In data experiments, the MAEs is below 1% relative error (1.21 K. for the worst case) which is an acceptable level of accuracy for most of the practical applications in the context of combustion temperature retrieval.[19] Also the standard deviation error for every model is lower because the discovered groups of data are more homogeneous.

5. Conclusions

In this paper we have presented a novel algorithm that tries to use the Gram matrix output structure to discover an input manifold that best represents that structure. We have used a graph similarity viewpoint to develop the algorithm which also introduces recursivity to find hierarchical relations between the clusters and improves the quality of the obtained models. We have tested the algorithm in an specific remote sensing application where we want to retrieve the temperature profile of a combustion from its spectrum of energy. The results obtained after clustering the data using our proposed algorithm improves the results obtained by a single model. Therefore, we suggest that using the output structure of the data can be a very powerful idea to discover some interesting information in the input data for regression problems.

Acknowledgments

The authors wish to acknowledge the Spanish Ministry of Education for financial support under the projects TRA2005-08892-C02-01, and M*::UC3M-TIN2008-06491-C04-04. E. García would also like to thank Margarita Gallardo.

References

1. I. T. Jollife, Principal component analysis, *Springer Series in Statistics Springer-Verlag (Chap. 8)*, (2nd edn.) New York (2002).
2. E. Lopez-Rubio and J. M. Ortiz-de-Lazcano-Lobato, Dynamic competitive probabilistic principal components analysis, *International Journal of Neural Systems* **19**(2) (2009) 91–103.
3. M. Turk and A. P. Pentland, Face recognition using eigenfaces, *IEEE Conference on Computer Vision and Pattern Recognition* (1991) 586–591.

4. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis-enhanced cosine radial basis function neural network for robust epilepsy and seizure detection, *IEEE Transactions on Biomedical Engineering* **55**(2) (2008) 512–518.
5. H. Yin and I. Hussain, Independent component analysis and non-gaussianity for blind image deconvolution deblurring, *Integrated Computer-Aided Engineering* **15**(3) (2009) 219–288.
6. J. Xu, A. Roy and M. H. Chowdhury, Noise separation in analog integrated circuits using independent component analysis technique, *Integrated Computer-Aided Engineering* **15**(2) (2008) 163–180.
7. Q. Wu and J. Ben-Arie, View invariant head recognition by hybrid PCA based reconstruction, *Integrated Computer-Aided Engineering* **15**(2) (2008) 97–108.
8. F. Cong, I. Kalyakin, T. Huttunen-Scott, H. Li, H. Lyytinen and T. Ristaniemi, Single-trial based independent component analysis on mismatch negativity in children, *International Journal of Neural Systems* **20**(4) (2010) 279–292.
9. K. Fukunaga, Introduction to statistical pattern recognition, *Academic Press*, New York (1990).
10. A. Samant and H. Adeli, Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis, *Computer-Aided Civil and Infrastructure Engineering* **13**(4) (2000) 241–250.
11. B. Schölkopf and A. Smola, Learning with kernels: Support vector machines, regularization, optimization, and beyond, *MIT Press*, Cambridge MA (2002).
12. A. Y. Ng, M. I. Jordan and Y. Weiss, On spectral clustering: Analysis and an algorithm, *NIPS* (2001) 849–856.
13. E. García-Cuesta, I. M. Galván and A. J. de Castro, Discriminant regression analysis to find homogeneous structures, *IDEAL* (2009) 191–199.
14. D. Cai, X. He and J. Han, SRDA: An efficient algorithm for large-scale discriminant analysis, *IEEE Transactions on Knowledge and Data Engineering* **20**(1) (2008) 1–12.
15. B. Mohar, Some applications of laplace eigenvalues of graphs, *Graph Symmetry: Algebraic Methods and Applications*, 497 of NATO ASI Series C (1997) 225–275.
16. L. S. Rothman *et al.*, The HITRAN molecular spectroscopic database: Edition of 2000 including updates through 2001, *J. Quant. Spectrosc. Radiat. Transfer* **82** (2003) 5–44.
17. E. García-Cuesta, I. M. Galván and A. J. de Castro, Multilayer perceptron as inverse model in a ground-based remote sensing temperature retrieval problem, *Engineering Applications of Artificial Intelligence* **21** (2008) 26–34.
18. M. Ester, H.-P. Kriegel, J. Sander and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *KDD* (1996) 226–231.
19. G. Lu, Y. Yan and M. Colechin, A digital imaging based multifuncional flame monitoring system, *IEEE T. Instrum. Meas.* **53** (2004) 1152–1158.