Soft computing models to analyze atmospheric pollution issues

ÁNGEL ARROYO, *Departamento de Ingeniería Civil, Universidad de Burgos, Burgos, Spain.*
E-mail: aarroyop@ubu.es

EMILIO CORCHADO, *Departamento de Informática y Automática, Universidad de Salamanca, Salamanca, Spain.*
E-mail: escorchado@usal.es

VERÓNICA TRICIO, *Departamento of Física, Universidad de Burgos, Burgos, Spain.*
E-mail: vtricio@ubu.es

Abstract

Multidisciplinary research into statistical and soft computing models is detailed that analyses data on emissions of atmospheric pollution in urban areas. The research analyses the impact on atmospheric pollution of an extended bank holiday weekend in Spain. Levels of atmospheric pollution are classified in relation to the days of the week, seeking to differentiate between working days and non-working days by taking account of such aspects as industrial activity and traffic levels. The case study is based on data collected by a station at the city of Burgos, which forms part of the pollution measurement station network within the Spanish Autonomous Region of Castile-Leon.

Keywords: Artificial neural networks, soft computing, meteorology, atmospheric pollution, statistical model.

1 Introduction

It has been accepted for some years now that air pollution not only represents a health risk, but that it also, e.g. reduces food production and vegetative growth due to its negative effects on photosynthesis.

Systematic measurements in Spain, which are usually taken within large cities, are fundamental, due to the health risks caused by high levels of atmospheric pollution. European legislation, will in the long term establish how and where such pollutants should be measured.

Thus, all efforts that are directed towards studying these phenomena [8, 25] may improve our understanding and help us to prevent the serious problematic nature of atmospheric pollution.

Several years ago, certain public, regional and municipal bodies adopted the principles of the V Programme Framework of the European Union and those of the Rio Conference concerning the application of community and international directives; in particular, those on environmental air quality in our cities. Previous directives were based on European legislation with which Spain, as a member of the European Community, was expected to comply.

The basis of this study is the application of a series of statistical and soft computing models for studying the evolution of air pollution throughout the week as a function of chosen variables.

Soft computing [4, 20, 27] consists of various technologies which are used to solve inexact and complex problems [33]. It is used to investigate, simulate and analyse complex issues and phenomena in an attempt to solve real-world problems [26].

2   Soft computing and statistical models

Several statistical and soft computing models are used in this study to analyse data taken from cases studies on aerosol pollutants in order to assess their performance.

2.1 Exploratory projection methods

Exploratory projection methods (EPP) [11, 16, 17] is a more recent statistical method aimed at solving the difficult problem of identifying structure in high-dimensional data. It does this by projecting the data onto a low-dimensional subspace in which we search for its structure by eye. However, not all projections will reveal the data's structure equally well. EPP, therefore, defines an index that measures how 'interesting' a given projection is, and it then represents the data in terms of projections that maximize that index. 'Interesting' structure is usually defined with respect to the fact that most projections of high-dimensional data onto arbitrary lines through most multi-dimensional data give almost Gaussian distributions [10]. Therefore, in order to identify 'interesting' features in data, it is necessary to look for those directions onto which the data projections are as far from the Gaussian as possible.

A simple measure of deviation from a Gaussian distribution is based on the higher order moments of the distribution. Kurtosis is based on the normalized fourth moment of the distribution and measures the heaviness of the tails of a distribution. A bimodal distribution will often have a negative Kurtosis and therefore negative Kurtosis can signal that a particular distribution shows evidence of clustering.

2.2 Principal component analysis (PCA)

Principal component analysis (PCA) [17] gives the best linear compression of the data in terms of least mean square error and can be implemented by several artificial neural networks [21, 22]. The basic PCA network [12] applied in this study is described by Equation (1) and Equation (2): an N-dimensional input vector at time t, $x(t)$, and an M-dimensional output vector, y, with W_{ij} being the weight linking input j to output i, and η being the learning rate. Its activation and learning may be described as follows:

Feedforward step by Equation (1):

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{1}$$

Feedback step by Equation (2):

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i \tag{2}$$

Change weights by Equation (3):

$$\Delta W_{ij} = \eta e_j y_i \tag{3}$$

This algorithm is equivalent to Oja's Subspace Algorithm [29] by Equation (4):

$$\Delta W_{ij} = \eta e_j y_i = \eta (x_j - \sum_k W_{kj} y_k) y_i \tag{4}$$

EPP [10, 15] projects the data onto a low-dimensional subspace which allows its structure to be examined by eye. As explained before, this is done by means of an index that measures the 'interestingness' of a given projection, the data for which is then represented by projections that maximize the most 'interesting' vectors. 'Interestingness' is usually defined by the fact that most projections of high-dimensional data onto arbitrary vectors through a majority of multi-dimensional data give almost Gaussian distributions [10, 28], which allows a search to be made for those directions onto which the data projections are as far from the Gaussian as possible.

2.3 ε-insensitive Hebbian learning

It has been shown [32] that the non-linear PCA rule by Equation (5):

$$\Delta W_{ij} = \eta \left(x_j f(y_i) - f(y_i) \sum_k W_{kj} f(y_k) \right) \tag{5}$$

Can be derived as an approximation of the best non-linear compression of the data. Initially, therefore, there is a cost function as in Equation (6):

$$J(W) = 1^T E \left\{ \left(\mathbf{x} - W f \left(W^T \mathbf{x} \right) \right)^2 \right\} \tag{6}$$

Which may be minimized to give Equation (5) [19] using the residual in the linear version of Equation (8) to define a cost function of the residual by Equation (7):

$$J = f_1(e) = f_1(x - W_y) \tag{7}$$

Where $f_1 = ||\cdot||^2$ is the (squared) Euclidean norm in the standard linear or non-linear PCA rule. With this choice of $f_1(\cdot)$, the cost function is minimized with respect to any set of samples from the data set on the assumption that the residuals are chosen independently and identically distributed from a standard Gaussian distribution. The minimization of J is equivalent to minimizing the negative log probability of the residual, if e is Gaussian as in Equation (8):

$$p(e) = \frac{1}{Z} \exp(-e^2) \tag{8}$$

Then, p will denote a general cost function associated with this network as in Equation (9):

$$J = -\log p(e) = (e)^2 + k \tag{9}$$

Where K is a constant. A gradient descent is performed on J in Equation (10):

$$\Delta w \propto -\frac{\partial J}{\partial W} = -\frac{\partial J}{\partial e}\frac{\partial e}{\partial W} \approx y(2e)^T \tag{10}$$

In which a less important term has been discarded [10].

In general [29], the minimization of such a cost function may be understood to increase the residual probability, depending on the probability density function (*pdf*) of the residuals. Thus, if the *pdf* of the residuals is known, this knowledge could be used to determine the optimal cost function. [14] investigated this possibility using the (one dimensional) function as in Equation (11):

$$p(e) = \frac{1}{2+\varepsilon}\exp(-|e|_\varepsilon) \tag{11}$$

Where as in Equation (12):

$$|e|_\varepsilon = \begin{cases} 0 & \forall |e| < \varepsilon \\ |e| - \varepsilon & \text{otherwise} \end{cases} \tag{12}$$

With ε being a small scalar.

With this model of the residual *pdf*, the optimal $f_1(\cdot)$ function is the ε-insensitive cost functionas in Equation (13):

$$f_1(e) = |e|_\varepsilon \tag{13}$$

In the case of the negative feedback network, the learning rule is shown in Equation (14):

$$\Delta w \propto -\frac{\partial J}{\partial W} = -\frac{\partial f_1(e)}{\partial e}\frac{\partial e}{\partial W} \tag{14}$$

Which gives Equation (15):

$$\Delta W_{ij} = \begin{cases} 0 & \text{if } |e_j| < \varepsilon \\ \text{otherwise} & \eta y\,(\text{sign}(e)) \end{cases} \tag{15}$$

2.4 Applying maximum likelihood Hebbian learning

The maximum likelihood Hebbian learning (MLHL) [3, 7] algorithm is defined as follows. Let the residual *pdf* after feedback be as in Equation (16):

$$p(e) = \frac{1}{Z}\exp(-|e|^p) \tag{16}$$

Where p is a parameter which relates the *pdf* to the learning rule as in Equation (20).

There is a general cost function associated with this network in Equation (17):

$$J = E(-\log p(e)) = E(|e|^2 + K) \tag{17}$$

Where K is a constant that is independent of W and the expectation is taken over the input data set. A gradient descent is performed on J as in Equation (18):

$$\Delta W \infty - \frac{\partial J}{\partial W}_{w(t-1)} = -\frac{\partial J}{\partial e}\frac{\partial e}{\partial w}_{w(t-1)} \approx$$
$$\approx E\{y|e|^{p-1} \, \mathrm{sig}(e))^T|_{w(t-1)}\} \tag{18}$$

Where T denotes the transpose of a vector, and the operation of taking powers of the norm of e is on an element wise basis as it is taken from a derivative of a scalar with respect to a vector. If the conditions of stochastic approximation [18] are satisfied, it may be approximated with a difference equation. The function to be approximated is clearly sufficiently smooth, the learning rate is acceptable and the rule is as follows by Equation (19):

$$\Delta W_{ij} = \eta \cdot y_i \cdot \mathrm{sign}(e_j)|e_j|^{p-1} \tag{19}$$

Researchers [16, 17] have shown that it is less important to obtain the exact distribution when searching for a specific source than it is to arrive at a distribution that is approximately correct. Therefore, the network operation is as follows: initially there is a feedforward step as in Equation (1) and a feedback step as in Equation (2) followed by a weights change as in Equation (20):

$$\Delta W_{ij} = \eta \cdot y_i \cdot \mathrm{sign}(e_j)|e_j|^{p-1} \tag{20}$$

By maximizing the likelihood of the residual with respect to the actual distribution, MLHL is matching the learning rule to the residual.

2.5 Cooperative maximum likelihood Hebbian learning

An extended version of the MLHL model is the cooperative maximum likelihood Hebbian learning (CMLHL) [6] model. CMLHL is based on MLHL [3, 7] adding lateral connections which have been derived from the Rectified Gaussian Distribution [7]. The resultant net can find the independent factors of a data set but does so in a way that captures some type of global ordering in the data set.

Consider an N-dimensional input vector (x), an M-dimensional output vector (y) and a weight matrix W, where the element W_{ij} represents the relationship between input x_j and output y_i, then as shown in [7, 13], the CMLHL can be carried out as a four-step procedure:

Feed-forward step, outputs are calculated in Equation (21):

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \tag{21}$$

Lateral activation passing step by Equation (22):

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \tag{22}$$

Feedback step by Equation (23):

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \tag{23}$$

Weights update step, learn the neural network by Equation (24):

$$\Delta W_{ij} = \eta \cdot y_i \cdot \text{sign}(e_j)|e_j|^{p-1} \tag{24}$$

Where t is the temperature, $[\cdot\cdot]^+$ is necessary to ensure that the **y**-values remain in the positive quadrant, η is the learning rate, τ is the 'strength' of the lateral connections, b the bias parameter, p a parameter related to the energy function [6, 7, 13] and A is a symmetric matrix used to modify the response to the data [28]. The effect of this matrix is based on the relation between the distances separating the output neurons.

2.6 ISOMAP

Methods for non-linear dimensionality reduction have proven successful in many applications, although the weakness of a method such as multidimensional scaling (MDS) [23] is that they are based on Euclidean distances and do not take the distribution of the neighbouring datapoints into account. ISOMAP non-linear dimensionality reduction [2] resolves this problem by attempting to preserve pairwise geodesic (or curvilinear) distance between datapoints. Geodesic distance is the distance between two points measured over the manifold. ISOMAP defines the geodesic distance as the sum of edge weights along the shortest path between two nodes (computed using Dijkstra's algorithm [23], e.g.). The doubly centered geodesic distance matrix K in ISOMAP is of the form Equation (25):

$$K = \frac{1}{2}HD^2H \tag{25}$$

Where $D^2 = D_{ij}^2$ means the element wise square of the geodesic distance matrix $D = [D_{ij}]$, and H is the centring matrix, given by Equation (26):

$$H = I_n - \frac{1}{N}e_N e_N^T \tag{26}$$

In which $e_N = [1\ldots1]^T \in R^N$

The top N eigenvectors of the geodesic distance matrix represent the coordinates in the new n-dimensional Euclidean space.

The complete isometric feature mapping, or ISOMAP, algorithm has three steps:

(1) Construct neighbourhood graph. Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than e *(e-Isomap)*, or if i is one of the K nearest neighbuors of j *(K-Isomap)*. Set edge lengths equal to $d_X(i,j)$.
(2) Compute shortest paths. Initialize $d_G(i,j) = d_X(i,j)$ if i, j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 5\ 1,2,\ldots,N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j),\ d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j))\}$ will contain the shortest path distances between all pairs of points in G.
(3) Construct d-dimensional embedding. Let λp be the p-th eigenvalue (in decreasing order) of the matrix $\lambda\ (D_G)$, and v_p^i be the i-th component of the p-th eigenvector.

ISOMAP complements, and may be combined with linear extensions of PCA based on higher order statistics, such as independent component analysis (ICA) [17]. It may also lead to a better

understanding of how the brain comes to represent the dynamic appearance of objects, where psychophysical studies of apparent motion point to the central role of geodesic transformations on non-linear manifolds.

2.7 Curvilinear component analysis

Curvilinear [9] is a non-linear projection method that preserves distance relationships in both input and output spaces. Curvilinear component analysis (CCA) is a useful method for redundant and non-linear data structure representation and can be used in dimensionality reduction. CCA is useful with highly non-linear data, where PCA or any other linear method fails to give suitable information [9].

CCA brings some improvements to other methods like Sammon's Mapping [24], although when unfolding a non-linear structure, Sammon's Mapping cannot reproduce all distances. One way to get round this problem consists in favouring local topology: CCA tries to reproduce short distances firstly, long distances being secondary. Formally, this reasoning led to the following error function (without normalization) by Equation (27):

$$E_{\text{CCA}} = \sum_{i,j=1}^{N} \left(d_{i,j}^n - d_{i,j}^p\right)^2 F_\lambda\left(d_{i,j}^p\right) \tag{27}$$

By comparison with E_{Sammon}, E_{CCA} has an additional weighting function F depending on $d_{i,j}^p$ and on parameter λ. The F factor is a decreasing function of its argument, so it is used to favour local topology preservation. For example, F could be a step function of (λ-d).

Another improvement brought by CCA to Sammon's mapping is the use of vector quantization (VQ). VQ decreases the computational load of the distance calculation (there $0.5 \times N(N-1)$ distance to compute). Unfortunately, the use of VQ implies the use of an interpolation in order to project vectors that are not prototypes of the codebook.

3 Case study. Analysis of atmospheric pollution throughout the week using soft computing models

This study presents interesting results related to the evolution of different pollution parameters using the records of an air quality control station (made available by the Department of the Environment-Directorate of Environmental Quality of the Government of the Spanish Autonomous Region of Castile-Leon) [30, 31]. The aforementioned station is situated in the urban area of the city of Burgos. The study was conducted over approximately half a year in 2007. After analysing all the weeks in the period, a week with a national bank holiday weekend in December was selected because many people in Spain travel over this period, greatly decreasing traffic in large cities and almost completely curtails industrial activity.

In this study, the following variables were analysed: SO_2, NO, NO_2, PM10. These are standard variables used to monitor urban air quality. The variables express the concentration of the pollutant in $\mu g/m^3$. Regulations have been established for the admitted margins of tolerance, values limit and periods of time established. In the case of NO and NO_2, the values limit come fixed on the Spanish Royal decree (R.D.) 1073/2003, transposed of the board 1999/30/CE. The limits of PM10 and SO_2 are the demanded for the R.D. 1073/2002, transposed of the board 1999/30/CE.

The general characteristics of the site where the measurement station used in the study was situated are as follows: Burgos, a city in the north-centre of Spain with a population of around
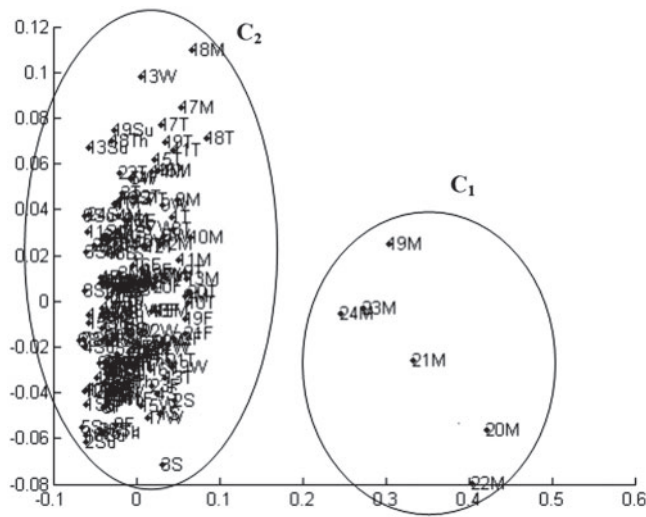
F<small>IG</small>. 1. PCA—projections of hourly pollution parameters.

170,000 inhabitants and a total municipal area of approximately $107 \, km^2$. The city of Burgos is 854 masl (meters above sea level) at latitude (N) 42°20′ and longitude (W) 3°42′. The measurement station is located within the city and may be classified as an urban station.

This study examines the performance of several statistical and soft computing methods when analysing the above-mentioned pollution variables, in order to track the evolution of air pollution over a significant time period. Thus, the main aim is to identify the extent to which air quality is affected on days with low industrial activity and reduced traffic.

3.1 Experiments and results

As already mentioned, the aim of this study is to analyse the difference between working days and non-working days using four variables with information on pollution. The analysis was undertaken over 6 months.

The study, which forms part of a more ambitious project [1, 5], is based on a file containing meteorological and pollution data sets recorded at 15-min intervals: a daily total of 96 records for all data in 2007, referring only to four variables. On this occasion, for presentation purpose, hourly averages are taken. Then, there is one sample per hour; 24 samples per day and 168 samples per week.

The information represented at each point is visually labelled from Figure 1 to Figure 5, which shows the time in 24 h format and a weekday initial (e.g. 6M means 6 am—Monday). All data were normalized for the study.

The graphical results obtained in this study are presented and analysed as follows.

This method identifies two main clusters (C_1 and C_2). Cluster C_2 groups most weekday samples and it is difficult to analyse its structure. C_1 is a group of scattered points that correspond in this case to Monday (M), which has the samples with the highest pollution throughout the entire week.

This method provides improved results with respect to PCA (Figure 1). Cluster C_1 represents the samples showing the highest pollution on Monday (M) evening, in the same way as was seen with
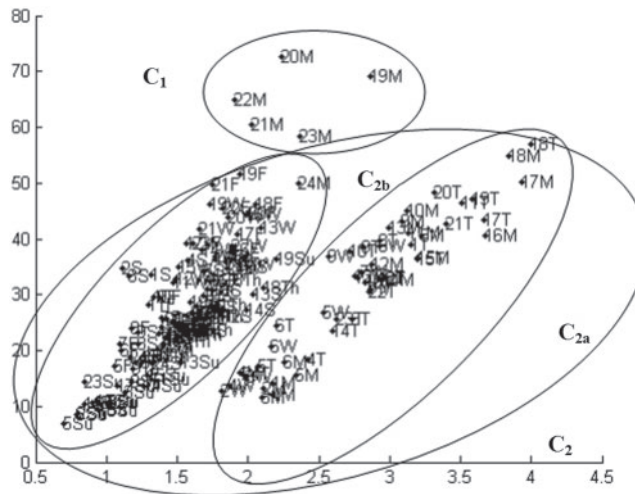
F_IG. 2. MLHL—projections of hourly pollution parameters.

PCA (Figure 1). Cluster C_2 in Figure 1 corresponds to clusters C_{2a} and C_{2b} in (Figure 2). The samples for cluster C_{2a} have lower pollution levels than the samples for cluster C_1, These samples belong to Tuesday (T) and Wednesday (W). Cluster C_{2b} is undoubtedly the one with the most samples. In this cluster, the points represent lower levels of pollution, corresponding to Thursday (Th), Friday (F), Saturday (S) and Sunday (Su). Thursday (Th) and Friday (F) were national bank holidays that correspond to Constitution Day and the Day of the Immaculate Virgin, respectively. On these days, industrial activity is greatly curtailed and many people traditionally travel long distances to visit relatives.

The difference with MLHL (Figure 2) is that CMLH is able to identify a new subcluster of samples. Cluster C_{2a} in (Figure 2) corresponds to clusters C_{2a1} and C_{2a2} in (Figure 3). Samples in C_{2a1} have higher levels of pollution than samples in C_{2a2}. Finally, Cluster C_{2b} contains the same samples as cluster C_{2b} in (Figure 2), but this time with sparser results. Applying CMLHL has made it possible to identify in a clearer and sparser way the internal structure based on pollution levels. C_{2b} contains the records with the lowest pollution levels recorded over the national bank holiday, Thursday (T), Friday (F), Saturday (S) and Sunday (Su).

This time the results are very similar to those obtained by applying MLHL (Figure 2). Again, cluster C_1 and cluster C_2 with the subclusters C_{2a} and C_{2b} are presented. The clusters in Figure 4 contain the same records as in Figure 2.

Once again, similar graphical results are obtained. CCA provides the same sample clusters as those that are obtained by applying MLHL and ISOMAP (Figures 2 and 4).

4 Conclusions and future works

The aim of this study is to demonstrate the validity of different statistical and neural models for the analysis of air pollution variables over the week in a typical European city such as Burgos, using the information available on pollution. After applying nine different statistical and soft computing models to the data sets (only five results are shown here), it has been shown that a degree of internal
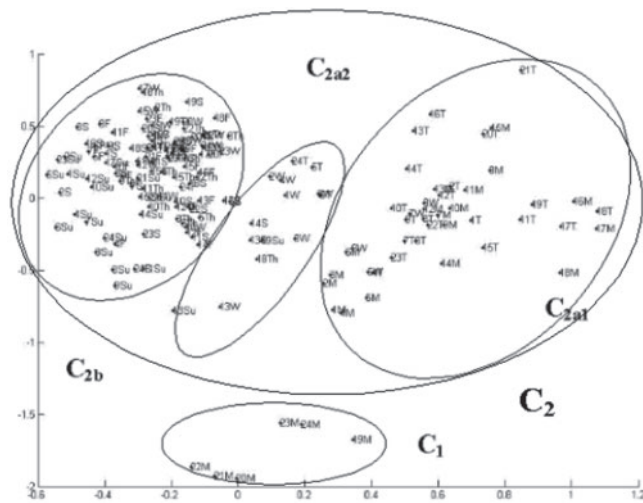
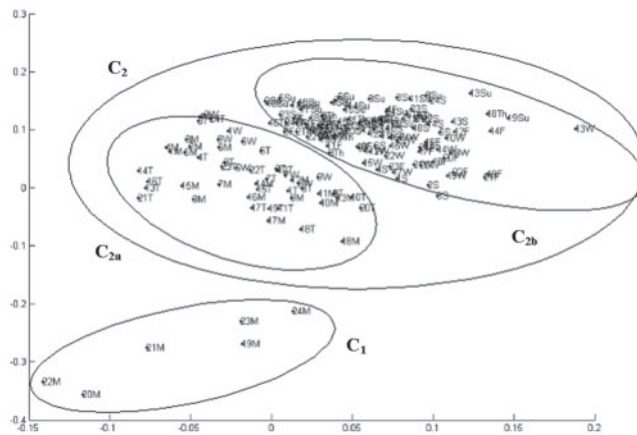FIG. 3. CMLHL—projections of hourly pollution parameters.

FIG. 4. ISOMAP—projections of hourly pollution parameters.

structure can be identified. PCA provides an initial approximation to the internal structure of the data, but it does not offer interesting information on hourly pollution readings. Other soft computing methods applied are able to provide an improved response. MLHL, ISOMAP and CCA provide a good response and very similar results among them. This good response is due to the fact that these methods offer a higher subdivision of the clusters; they group pollution values and classify them depending on the days of the week. CMLHL offers the best visual results, providing a very fine response. This is due to the fact that it is capable of differentiating new subclusters of data. Clearly, it was possible to fit each of the clusters to average hourly values of pollution. It is possible to affirm that the cluster with the lowest levels of pollution corresponds to the national bank holiday: Thursday (T), Friday (F), Saturday (S) and Sunday (Su).
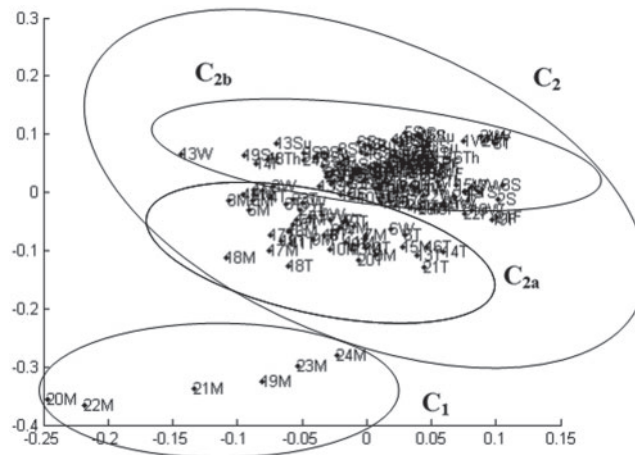
F<small>IG</small>. 5. CCA—projections of hourly pollution parameters.

In this study, only pollutant variables are considered, but further clarification would be desirable on the important influence of atmospheric conditions on the evolution of atmospheric pollution. Future work will be based on the analysis of more complex data sets that combine pollution and meteorological data in the city of Burgos, using soft computing methods, in order to identify the relationship between pollution and meteorological conditions throughout the week and over different time periods.

Funding

References

[1] A. Arroyo, E. Corchado and V. Tricio. Atmospheric pollution analysis by unsupervised learning. *Intelligent Data Engineering and Automated Learning*, **2009**, 767–772, 2009.

[2] H. P. Chan, D. Y. Yeung and Y. Xiong. Super – resolution through neighbor embedding. *IEEE Computer Society Conference on computer vision and pattern recognition*, **1**, 275–282, 2004.

[3] E. Corchado and C. Fyfe. Connectionist techniques for the identification and suppression of interfering underlying factors. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, **17**, 1447–1466, 2003.

[4] E. Corchado and A. Herrero. Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing*, **11**, 2042–2056, 2011.

[5] E. Corchado, A. Arroyo and V. Tricio. Soft computing models to identify typical meteorological days. *Logic Journal of the IGPL*, 2010; [Epub ahead of print; doi:10.1093/jigpal/jzq035, 21 July 2010, http://jigpal.oxfordjournals.org/content/early/].

[6] E. Corchado, Y. Han and C. Fyfe. Structuring global responses of local filters using lateral connections. *Journal of Experimental and Theoretical Artificial Intelligence*, **15**, 473–487, 2003.

[7] E. Corchado, D. MacDonald and C. Fyfe. Maximum and minimum likelihood Hebbian learning for exploratory projection pursuit. *Data Mining and Knowledge Discovery*, **8**, 203–225, 2004.

[8] O. Cuesta-Santos, A. Collazo, A. Wallo, R. Labrador, M. Gonzalez and P. Ortiz. Deposition of atmospheric nitrogen compounds in humid tropical Cuba. *Scientific World Journal*, **1**, 238–244, 2001.

[9] P. Demartines and D. J. Hérault. Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks* **8**, 148–154, 1997.

[10] P. Diaconis and D. Freedman.Asymptotics of graphical projection pursuit. *The Annals of Statistics*, **12**, 793–815, 1984.

[11] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data-analysis. *IEEE Transactions on Computers*, **23**, 881–890, 1974.

[12] C. Fyfe and R. Baddeley. Non-linear data structure extraction using simple Hebbian networks. *Biological Cybernetics*, **72**, 541–533, 1995.

[13] C. Fyfe and E. Corchado. Maximum likelihood Hebbian rules. *Proceedings of the 10th European Symposium on Artificial Neural Networks (ESANN 2002)*, 143–148, 2002.

[14] C. Fyfe and D. MacDonald. ε-insensitive Hebbian learning. *Neurocomputing*, **47**, 57–35, 2002.

[15] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Education Psychology*, **24**, 417–444, 1933.

[16] A. Hyvärinen. Complexity pursuit: separating interesting components from time series. *Neural Computation*, **13**, 898–883, 2001.

[17] A. Hyvärinen, J. Karhunen and E. Oja. *Independent Component Analysis.* Wiley, 2002.

[18] R. L. Kashyap, C. C. Blaydon and K. S. Fu. Stochastic approximation. In *A Prelude to Neural Networks, Adaptive and Learning Systems*, Jerry M. Mendel ed., Prentice Hall, 1994.

[19] P. L. Lai, D. Charles and C. Fyfe. Seeking independence using biologically inspired artificial neural networks. In *Developments in Artificial Neural Network Theory: Independent Component Analysis and Blind Source Separation*, M. A. Girolami ed., Springer, 2000.

[20] L. Lima, P. Novais, R. Costa, J. B. Cruz and J. Neves. Group decision making and Quality-of-Information in e-Health systems. *Logic Journal of the IGPL*, 2010; [Epub ahead of print; doi:10.1093/jigpal/jzq029, 3 July 2010, http://jigpal.oxfordjournals.org/content/early/].

[21] E. Oja. Neural networks, principal components and subspaces. *International Journal of Neural Systems*, **1**, 68–61, 1989.

[22] E. Oja, H. Ogawa and J. Wangviwattana. Principal components analysis by homogeneous neural networks, part 1, the weighted subspace criterion. *IEICE Transaction on Information and Systems*, **E75D**, 375–366, 1992.

[23] S. T. Roweis and Saul L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, **290**, 2323–2326, 2000.

[24] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computer*, **18**, 401–409, 1969.

[25] R. San José, J. L. Pérez and R. M. Gonzalez. Air quality real-time operational system for large industrial plants: an application of MM5-CMAQ modelling system. *Twelfth International conference on modelling, monitoring and management of air pollution, Rhodes, Greece*, Vol. 14, pp. 245–253, 2004.

[26] J. Sedano, E. Corchado, L. Curiel, J. R. Villar and P. M. Bravo. Modelling a pneumatic drill process by a two-steps AI Model. *International Journal of Computer Mathematics*, **86**, 1769–1777, 2009.

[27] J. Sedano, L. Curiel, E. Corchado, E. de la Cal and J. R. Villar. A soft computing method for detecting lifetime building thermal insulation failures. *Integrated Computer-Aided Engineering*, **17**, 103–115, 2010.

[28] S. Seung, N. D. Socci and D. Lee. The rectified Gaussian distribution. *Advances in Neural Information Processing Systems*, **10**, 350–356, 1998.

[29] J. Smola and B. Scholkopf. A tutorial on support vector regression. *Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series*, Royal Holloway, 1998.

[30] V. Tricio, R. Viloria and A. Minguito. Ozone measurements in urban and semi-rural sites at Burgos (Spain). *Geophysical Research Abstracts*, **5**, EAE03-A-14249, 2003.

[31] V. Tricio, R. Viloria and A. Minguito. Temporary evolution of ozone air quality in province of Burgos (Spain): regression and cluster techniques. *Geophysical Research Abstracts*, **9**, 10951, Sref-ID: 1607-7962/gra/EGU07-A-10951, 2007.

[32] L. Xu. Least mean square error reconstruction for self-organizing nets. *Neural Networks*, **6**, 648–627, 1993.

[33] L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, **37**, 77–84, 1994.