CORRADO DI NATALE[1,2] , EUGENIO MARTINELLI[1], ARNALDO D'AMICO[1,2]

1. University of Rome "Tor Vergata"
Department of Electronic Engineering, Roma, Italy
2. CNR-IMM, Sezione di Roma, Roma, Italy

PRE-PROCESSING AND PATTERN RECOGNITION METHODS FOR ARTIFICIAL OLFACTION SYSTEMS: A REVIEW

Data analysis is a fundamental part of artificial olfaction systems. Nonetheless, while electrical and mechanical components of electronic noses are deeply investigated and optimised for the specific applications, data analysis is sometimes performed without any critical understanding about hypotheses on which it is funded.

Several methodologies were imported into this field and utilised to correlate electronic nose data to the information required for classification purposes. Chemometrics and neural networks are the sources of the most popular methods currently in use.

In this paper some of the most used methodologies of data analysis are reviewed and their use for electronic nose data is discussed. Although the discussion is given considering chemical sensor arrays, the conclusions are generic and valid for a broader range of sensor systems.

1. INTRODUCTION

Artificial olfaction systems are attempts to reproduce the functions of the natural olfaction sense. After some twenty years after the first introduction of electronic noses [1], a sort of standard electronic nose model common to the many available implementations of artificial olfaction may be identified. In this model three major sub-systems are found: sampling, sensor array and data analysis. While sensor arrays are oriented to exploit the analogies with natural receptors (e.g. cross-selectivity and redundancy), the data analysis still does not take advantage of natural data treatment paradigms. The principles of data processing of natural olfaction receptor signals are still rather unknown and the differences in terms of sensor signals make them hardly applicable to process sensors data, although the topic is evolving and first attempts to derive analysis methodologies from natural paradigms have already been proposed [2].

Except these rare attempts to include biological paradigms, electronic nose data analysis is rather based on techniques borrowed from other disciplines and chosen among that vast class of algorithms known as pattern recognition.

Pattern recognition in chemistry appeared at the end of the sixties. Among the first applications the use has to be mentioned of the Learning Machine on behalf of the analytical chemists at the Washington State University. They classified molecules according to molecular structural categories used to form patterns [4].

In the gas sensing area, multivariate methods of pattern recognition are commonly required when sensor arrays, composed of real, non-selective, cross-sensitive sensors, are utilised. Pattern recognition, exploiting the cross-correlation, extracts information contained in the sensor outputs ensemble. Only a few of the manifold of pattern recognition techniques introduced in other disciplines are utilised (ref. 5, 6, and 7 offer a review of the most practised ones).

Formally, pattern recognition may be defined as “the mapping of a pattern from a given pattern space into a class-membership function” [3].

A pattern can be defined as any an ordered set of real numbers. In analytical chemistry, the response of a multi-channel instrument (such as gas chromatography or spectrophotometer) forms a pattern. For a sensor array, the response of the sensors of the array takes the name of sensors pattern.

Class-membership space represents either quantitative or qualitative set of quantities. When quantification is considered, the class-membership function turns out to be a vectorial function defined in a metric space. In the case of class recognition, class-membership is an ensemble of abstract sets to which each measured sample is assumed to belong. In the first case we prefer to talk of multicomponent analysis while the term pattern recognition is usually referred to the second case.

Mathematically, a pattern is represented by a vector belonging to a proper vector space. The space where sensors responses are represented is called sensor space. The simplest of these representations considers one scalar response for each sensor. This response are dubbed sensor feature and feature extraction is the operation extracting from a sensor signal the synthetic descriptors that form the patterns.

When a sensor (independently from its working principle) is exposed to a gas or a mixture of gases, it gives a response depending on the nature of the gas and its concentration. Almost the totality of the examples reported in literature consider as the sensor response either the absolute or the relative change in sensor signal, measured in two steady-state conditions in the absence and in presence of the gas, and when all the transients are ended. This definition leaves not exploited the dynamics behaviour of the sensor signal. Several authors studied the optimisation of feature extraction in order to maximise the array performance, among them Eklöv et al [8] showed a method to optimise the information extraction from an array of MOSFET sensors. In their paper, the dynamical behaviour of sensor signals was represented by seventeen different features, partially correlated one each other. The optimisation was carried out maximising the performances of an array of MOSFET sensors in the quantitative estimation of mixtures of hydrogen and ethanol. In spite of the evident restriction of the studied case this paper shows that the quantification error is minimised when descriptors considering the dynamical behaviour are taken into account. On the other hand, it has to be mentioned that the dynamical behaviour is very sensitive to the fluctuations of the sample delivery system. More recently some papers investigated the sensor signal in a representation space similar to the phase space of classic mechanics. In such space special features like the areas spanned by sensor trajectories and morphological descriptors were introduced as alternative features [9, 10].

Figure 1 shows, as a block scheme, all the conceptual steps of a typical electronic nose experiment. It is worth to note the importance of the cases extracted from the abstract classes and used in the experiment. Unbiased sampling is not always easy in practical applications where the definition of the abstract classes may be vague and the effective class membership may be not fully certain. It is important to keep this problem in mind when the performances of an electronic nose have to be evaluated. In Fig. 1 three main blocks are shown corresponding to pre-processing, exploratory analysis, and classification. In the following, each of these items will be discussed evidencing both theoretical and practical aspects.

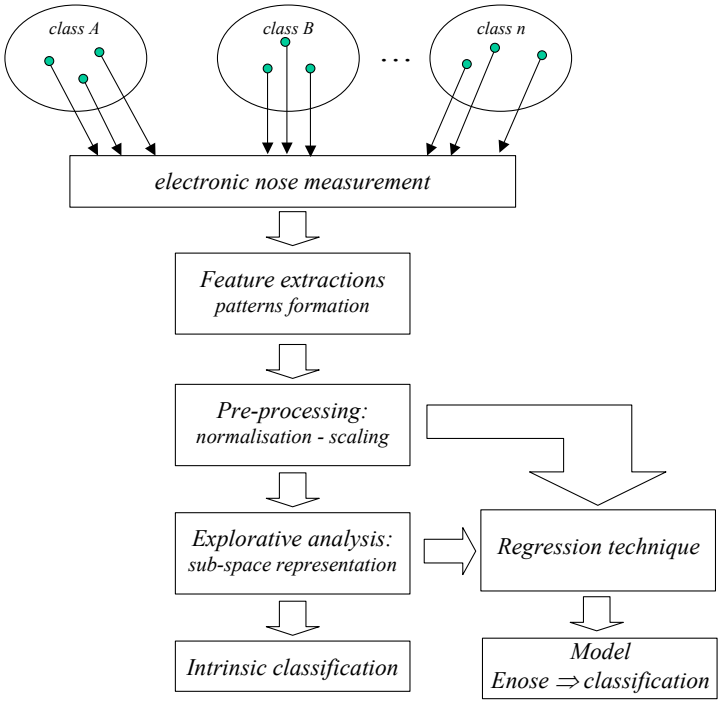

Fig.1. Sketch of a typical electronic nose experiment. Samples are extracted from abstract classes and measured by the instrument. Once features are extracted and pattern formed, pre processing may be applied to standardise the data and to remove the influence of quantitative information. Hence, data may be explored to discover intrinsic classification properties, or used to build a classifier through regression. Either statistical approach or neural networks are used to obtain classifiers.

2. DATA PRE-PROCESSING: SCALING AND NORMALISATION

An usual procedure in pattern recognition is the scaling of the data. Instead of using raw data, two main scaling procedures are widely used: zero-centred and autoscaling.

Zero-centred data means that each sensor is shifted across the zero value, so that the mean of the responses is zero. Zero-centred scaling may be important when the assumption of a known statistic distribution of the data is used. For instance, in case of a normal distribution, zero-centred data are completely described only by the covariance matrix.

Autoscaling means to scale each sensor to zero-mean and unitary-variance. This operation equalises the dynamics of the sensor responses avoiding that a sensor with a larger response range, may hide the contribution of other sensors dynamically limited. Further, autoscaling makes the sensor responses dimensionless; this feature becomes necessary when sensors whose signals are expressed in different units are joined in the same array. This is the case of hybrid arrays (different sensor technologies in the same array) and when electronic noses are fused with other instruments, e.g. the fusion of electronic noses and electronic tongues [11].

3. NORMALISATION

Horner and Hierold [12] have shown that the application of simple normalisation of sensors data can greatly help in removing the quantitative information and putting in evidence the qualitative aspects of the data.

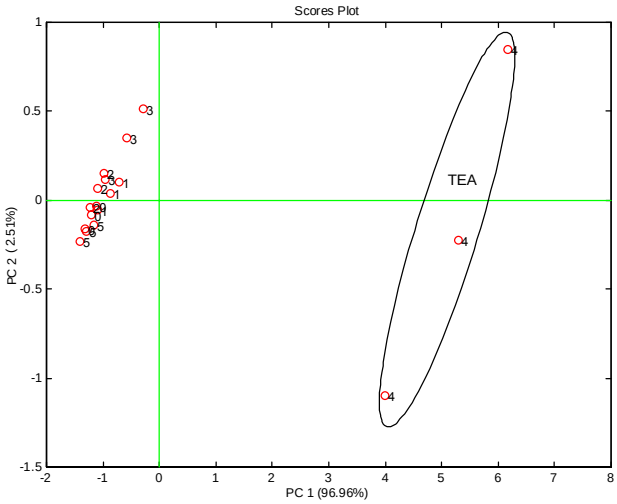

Fig.2. Examples of data characterised by strong concentration effects. Data are related to a quartz microbalance array exposed to six volatile compounds, each measured three times at different concentration. Only triethylamine (TEA in the plot) data emerge (experimental details in Ref. [10])

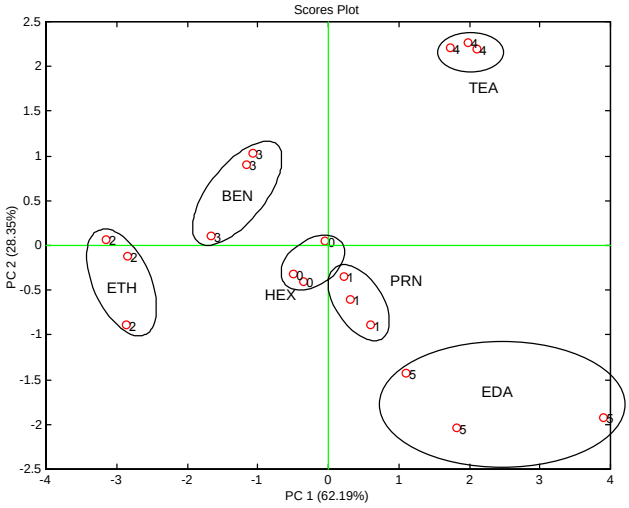

Fig.3. Data of Fig. 2 after the application of linear normalisation of Eq. 2. Classes are now clearly separated. It is worth to note the condensation of TEA data (class 4) into a very restricted region. Meaning of the classes: HEX: hexane; PRN: propanal; ETH: ethanol; BEN: benzene; TEA: triethylamine; EDA: ethylendiamine.

Figures 2 and 3 show the situation in the case of a quartz microbalance (QMB) array exposed to a certain number of different compounds measured at various concentration levels [13]. The cross-selectivity of the sensors make their responses ambiguous, so that different samples, due to

a combination of qualitative and quantitative aspects may give rise to similar sensor responses. In Fig. 2 the confusion among the data puts in evidence the ambiguity of the sensor response.

A simple way to disentangle the information is found when the relationship between sensor responses and concentrations of analytes is linear, such as:

$$\Delta f_i = K_{ij} c_j \,, \tag{1}$$

where: Δf_i - the response of the i-th QMB sensor,
K_{ij} - is the sensitivity of the i-th sensor towards the j-th compounds,
c_j - the concentration of the j-th compound.

The normalisation consists in dividing each sensor response by the sum of all the sensor responses to the same sample, so that the concentration information disappears.

$$\Delta f_i \Rightarrow \frac{K_{ij} c_j}{\sum_m K_{mj} c_j} = \frac{K_{ij}}{\sum_m K_{mj}} \,. \tag{2}$$

Figure 3 shows the PCA score plot of the same data of Fig. 2 after the application of Eq. 2. The application of linear normalisation to an array of linear sensors should produce, on the PCA score plot, one point for each compound, independent of its concentration, and achieving the highest possible recognition. Deviations from ideal behaviour, as shown in Fig. 3, are due to the presence of measurement errors, and to the non-linear relationship between sensor response and concentration.

In the previously quoted paper, Horner and Hierold, treated also the case of power law, valid for instance for metal-oxide semiconductor gas sensors. Equation 2 can be extended to sensors described by a power-law ($z = c^{\alpha}$) simply linearising, through the logarithm, the sensor response.

Normalisation is, in practice, useful to counteract possible fluctuations in the sample concentration. These fluctuations are mostly due to sample temperature fluctuations, and to instabilities of the sampling system that may lead to variations of the dilution factor of the sample with the carrier gas. Of course, normalisation is of limited effiicency because the mentioned assumptions strictly hold for simple gases and they are faded to fail when mixtures of compounds are measured. Furthermore, in complex mixtures, temperature fluctuations do not result in a general concentration shift, but since individual compounds have different boiling temperatures, each component of a mixture changes differently so that both quantitative (concentration shift) and qualitative (pattern distortion) variations occur.

4. EXPLORATORY TECHNIQUES

Given a set of data related to a number of measurements, exploratory techniques aim at studying the intrinsic characteristics of the data in order to discover their internal properties.

Exploratory analysis evidences the attitude of an electronic nose to be utilised for a given application, leaving to the supervised classification the task to build a model to be used to predict the class membership of unknown samples.

Two main groups of exploratory analysis may be identified: representation techniques and clustering techniques.

Representation techniques are a group of algorithms aimed at providing a representation of the data in a space of dimensions lower than the original sensor space. The most popular of these methods are Principal Component Analysis, Self Organising Map, and Sammon's mapping. Each of these techniques is based on specific hypotheses about the nature of the data and the sensor space. Each of them tends to preserve some particular characteristic of the data. Table 1 lists the main characteristics of the above mentioned methods. It is worth to remark that the simplest, in terms of calculus and interpretation of results, is that based on the strongest assumption about the statistical distribution of the data. On the other hand, when assumptions about data distribution are removed, a neural network is necessary for data representation.

Table 1. The table reassumes the main methodologies to represent, on a suitable sub-space, multidimensional data sets.

Method	***Hypotheses***	***Represented Feature***	***Algorithm***
Principal Component Analysis	Gaussian Distribution	Maximum likelihood	Linear projection onto the basis of the canonical axis of the data covariance matrix. Direct calculus
Self Organising Map	Definition of a Metric in the sensor space (normally Euclidean)	Topology conservation of data structure.	Quantisation of the sensor space, where each volume contains an equal probability of data occurrence. Iterative calculus (neural network)
Sammons Mapping	Definition of a Metric in the sensor space (normally Euclidean)	Data similarities	Optimisation of an error function describing the difference between the distance matrices calculated in the sensor space and the representation space. Iterative calculus

Clustering techniques are mostly based on the concept of similarity expressed through the definition of a metric (distances calculus rule) in the sensor space. The most trivial and common choice is to express the similarity as an Euclidean distance. Other definitions, such as the Mahalanobis distance, are also used [14].

4.1. Principal component analysis

The scope of the Principal Component Analysis (PCA) is a *reliable* portrayal of a data set in a sub-space of reduced dimensionality. *Reliable* here means that data are reproduced preserving their statistical properties.

PCA is part of a large set of linear transformations reassumed as: $\mathbf{S} = \mathbf{WX}$. Here $\mathbf{X}$ is the original data set, $\mathbf{W}$ is the transformation matrix, and $\mathbf{S}$ are the data points in the representation space.

PCA has been firstly introduced in the thirties in psychology to describe the behaviour of humans [14]. It has then been deeply developed in chemometrics, where it has been introduced at the beginning to analyse spectroscopic and chromatographic data, which are characterised by a higher correlation among the spectra channels [15].

The possibility of a reliable representation of an electronic nose data set in a sub-space of reduced dimensionality lies in the fact that the chemical sensors always exhibit a certain degree

of correlation among them. The principal component analysis consists in finding an orthogonal basis where the correlation among sensors disappears.

Correlation means that given a sensor space of dimension N the effective dimension of the sub-space occupied by the data is less than N. This dimension can also be calculated taking advantage of the algorithms developed to describe dynamic systems. In particular, it is worth to mention the correlation distance that allows evaluating the fractional dimensionality of a data-set [16]. Correlation distance provides an independent way to evaluate the expected reduction of dimensions.

PCA is calculated considering only the second momentum of the probability distribution of the data (covariance matrix). Only for normally distributed data the covariance matrix ($\mathbf{X}^T\mathbf{X}$) completely describes zero-centred data. From a geometric point of view, any covariance matrix, since it is a symmetric matrix, is associated to a hyper-ellipsoid in the N dimensional space. PCA corresponds to a coordinates rotation to represent the associated hyper-ellipsoid in its canonical form. In practice, the novel coordinate basis is coincident with the hyper-ellipsoid principal axis.

The reduction of an ellipsoid to its canonical form is a typical linear algebra operation, it is performed through the eigenvectors of the associated matrix.

Since that PCA can be calculated with the following rule. Let us consider a matrix $\mathbf{X}$ of data, let $\mathbf{C}=\mathbf{X}^T\mathbf{X}$ be the covariance matrix of $\mathbf{X}$. The i-th principal component of $\mathbf{X}$ is $\mathbf{X}^T\lambda(i)$, where $\lambda(i)$ is the i-th normalised eigenvector of C corresponding to the i-th largest eigenvalue. A sketch of the geometric meaning is shown in Fig. 4.

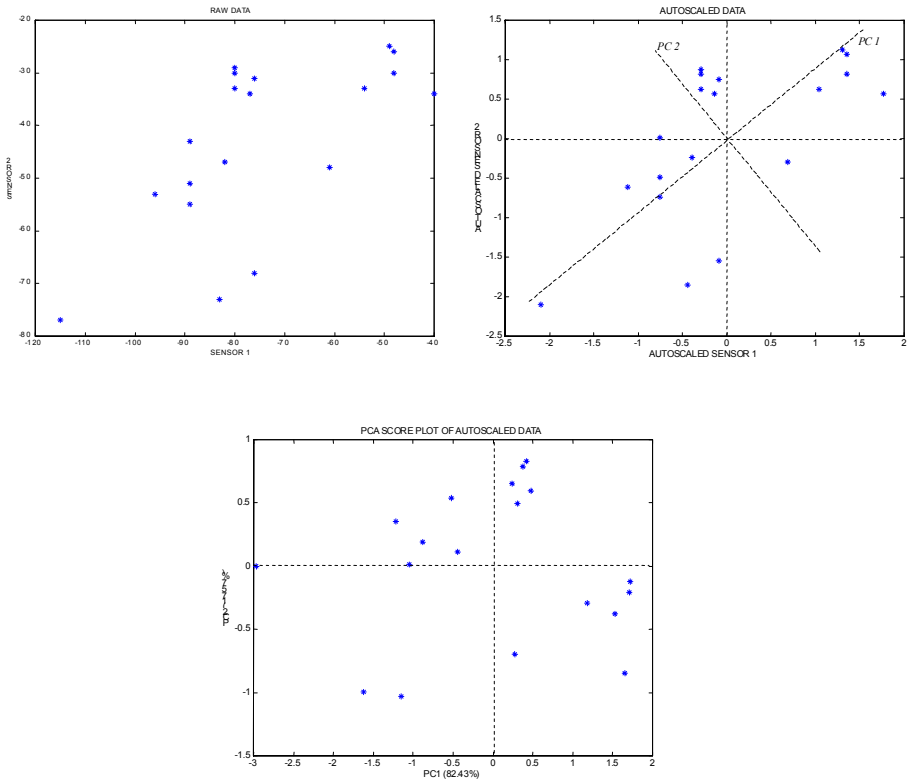

Fig. 4. Geometrical meaning of PCA. Two sensors are considered. The three plots show from the upper to the lower: the raw data set, the autoscaled data and the data plotted in the principal components basis. The principal components, in the original space, are shown in the second plot.

The eigenvalues happen to be directly proportional to the variance explained by their correspondent eigenvector, so that considering the relative values of the eigenvalues $\lambda(i)$ it is possible to reduce the representation to only those components carrying most of the information.

Given a matrix of data, PCA provides two main kinds of information called scores and loadings. The scores are concerned with the measurements, and they are defined as the coordinates of each vector measurement (a row of matrix **X**) in the principal components base.

The loadings are concerned with the sensors. They measure the contribution of each sensor to the PCA basis. A large loading, for a sensor, means that the principal component is aligned along the sensor direction.

It is important to note that highest eigenvalues correspond to components defining the directions of highest correlation among the sensors, while the components characterised by smaller eigenvalues are related to uncorrelated directions. Since sensor noises are uncorrelated, the representation of the data using only the most meaningful components removes the noise of the sensors. In this way PCA is used to remove noise from spectra [15].

When applied to electronic nose data the presence of various sources of correlated disturbances has to be considered. As an example, sample temperature fluctuations induce correlated disturbances which are described by the principal components of highest order. When these disturbances are important, the first principal component has to be eliminated in order to emphasize the relevant data properties. A set of algorithms called Minor Component Analysis (MCA) was introduced to take into account these phenomena mainly in image analysis [17].

The hypothesis of normal distribution is a strong limitation that should be kept always in mind when PCA is used. In electronic nose experiments, samples are usually extracted from more than one class, and not always the totality of measurements results in a normally distributed data set. Nonetheless, PCA is frequently used to analyse electronic nose data. Due to the high correlation normally shown by electronic nose sensors, PCA allows a visual display of electronic nose data in either 2D or 3D plots. Higher order methods were proposed and studied to solve pattern recognition problems in other application fields. It is worth mentioning here the Independent Component Analysis (ICA) that has been applied successfully in image and sound analysis problems [18]. Recently ICA was also applied to process electronic nose data resulting as a powerful pre-processor of data [19].

Nonetheless, a sub-set belonging to one class may be very likely normally distributed. In this case a PCA calculated on one class could not work in describing data belonging to another class. In this way, the membership of data to each class can be evaluated. This aspect is used by classification method called SIMCA (Soft Independent Modelling of Class Analogy), it is a clever exploitation of the limitations of PCA to build a classification methodology [20].

Non linear PCA algorithms have also been developed to provide a representation along principle curves rather than principal directions [21]. Also neural networks were proposed to solve the problem of faithful representation of multidimensional data in representation space of lower dimensions [17].

Another limitation to the use of PCA comes from the fact that being a linear projection it may introduce mistakes. Indeed, in the projection, data separated in the original space, may result with similar score, phenomena likely to that producing constellation in the starred sky.

A final consideration about PCA is concerned with its use as a pre-processor of non-linear methods such as neural networks [22]. The assumption of normal distribution of the data limits conceptually all the following steps of analysis to be adherent to this hypothesis, and also if

positive results are sometimes achieved they have to be considered as serendipitous events and the procedure theoretically wrong.

4.2. Self organizing map

Self-Organising Map (SOM) is a neural network based on the concept of vectorial quantisation, which is a method to approximate the probability density function of vectorial variables by a finite set of codebook (or reference) vectors [23]. The scope of the SOM is to create a globally ordered map of sensors features onto a layered neural network. SOM defines a sort of "elastic net" of nodes (each representing a codebook vector) fitted to the sensor space to approximate its density function. In this way the map can represent any sensor data distribution disregarding its complexity.

SOM was introduced, at the beginning of the eighties, by T. Kohonen, and due to its great flexibility it has been utilised in many different fields, from speech recognition to process control in industrial plants [24].

In 1994, the SOM was proposed to the sensor community as a method for the extraction, from a sample data set, of an amount of information on the behaviour of a multisensor system [25]. Nevertheless, with the exception of few cases [26, 27], SOM has not been intensively utilised on behalf of researchers in the sensor field. The reason for this is that, although SOM is based on an intuitive algorithm, the interpretation of its results is not straightforward, and it requires careful reading of the network parameters.

SOM is a network formed by N neurons arranged as the nodes of a planar grid, so that each neuron has four neighbours. Each neuron is identified by a vector **r**, whose components are the node coordinates in the grid. The neurons are logic elements with two possible states; they have m input (vector **z**) and one output. An input is a real value vector, while the output is either active (value 1) or inactive (value 0) . Each neuron is characterised by a m-component codebook vector $\mathbf{w}_r$, which represents the neuron in the input space.

This logical structure accepts inputs from a sensor array: due to each input the codebook vectors of neurons are modified by a learning algorithm. It aims at constructing the whole set of codebook vectors $\{\mathbf{w}_r\}$ of the grid as a discrete approximation of all the supplied **z** vectors.
Once a new input is provided, the learning algorithm prescribes two stages:
1- Response. Determination of the winner neuron from the condition:

$$\left\|\mathbf{z}-\mathbf{w}_s\right\| \le \left\|\mathbf{z}-\mathbf{w}_r\right\| \quad \text{for all } r. \tag{3}$$

Namely the neuron whose codebook vector is closest to the input **z** is selected as the winner.
2- Adaptation. Variation of the codebook vectors of all the neurons according to:

$$\mathbf{w}_r^{new} = \mathbf{w}_r^{old} + \alpha\, h_{rs}\left(\mathbf{z}-\mathbf{w}_r^{old}\right) \quad \text{for all } r, \tag{4}$$

where: α - a convergence parameter, h_{rs} - the neighbour function.
As an example, h_{rs} may have the following form:

$$h_{rs} = \exp\left(-\frac{\left\|\mathbf{r}-\mathbf{s}\right\|^2}{2\sigma^2}\right). \tag{5}$$

The function h_{rs} defines an area, around the s neuron, selecting the neurons participating in the adaptation stage. The parameter σ is a length scale defining the proximity of neuron *s*.

From a practical point of view a calibration data-set is used to train the SOM and, at each learning step a datum, randomly selected, is presented to the SOM. The process takes place until the network converges, namely the codebook vectors do not change more than a tolerance value. Once the network reaches convergence, the codebook vectors of the SOM neurons contain the model of the phenomena of which the calibration data set is a sample.

The first information that can be extracted from the codebook vectors is the reciprocal distance between them. This information may be represented onto the SOM grid attributing to the segments, joining adjacent neurons, a grey level whose intensity is proportional to the distance between the connected codebook vectors [28]. This representation gives pictorial information about the clustering of the data. Indeed, light lines define clusters while dark lines indicate gaps between the neurons. By fixing a threshold on the distance, clusters may be defined [29].

It is worthwhile to discuss the relation between SOM and PCA. PCA is a linear projection of the data set onto a space of reduced dimensions, typically a plane. The projection is unique for the whole domain of the data; while the SOM is rather a set of local linear models. With respect to PCA, SOM can provide a more accurate display of the data avoiding the overlapping effect due to linear projection [13].

In sensor arrays there is the necessity of evaluating the contribution that each sensor brings to the whole array. Loadings are frequently used for this scope. PCA provides an information that is a sort of mean over the whole domain of the data. SOM gives the opportunity to evaluate the behaviour of each sensor studying the components of the codebook vectors. Since each sensor is a coordinate of the SOM input space it is also a component of the codebook vectors. The behaviour of sensors in the experiment is then stored as a component of the codebook vectors. This information may also be graphically displayed onto the SOM grid as either a coloured map or a 3D surface. Both the choices give the possibility of monitoring the influence of a sensor on the whole domain of the data and the comparison of plots reveals whether or not two or more sensors are correlated one to each other.

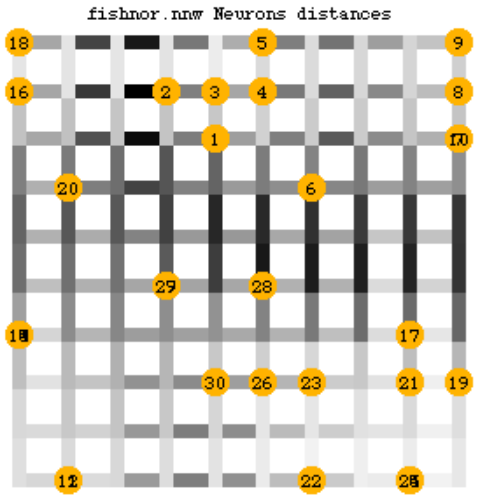

Fig. 5. Example of a SOM trained with electronic nose data. Data are related to the measure of fish stored in ice for several days. The distance between codebook vectors in the original space is represented in grey scale. The presence of three regions is evidenced, by the clear colour. (experimental details in Ref. 27).

In Fig. 5 an example of a SOM neurons lattice is shown. Data are related to an experiment where the days of storage on ice of codfish fillets were evaluated by a quartz microbalances based electronic nose. Details about the experiment can be found in Ref. 30. Labelled neurons indicate those units that, after training, correspond to the input data. Neurons are connected by a segment whose grey intensity is directly proportional to the distance between the codebook vectors of the connected neurons. Classes can be defined by fixing a threshold on the distance between codebook vectors. Neurons whose codebook vectors are closer than the threshold are considered as belonging to the same clusters, vice versa they belong to different clusters if their codebook vectors differ more than the threshold value. Figure 6 shows the clusters formed at a threshold value of 0.4 (distances are normalised in the range [0, 1]). Each cluster corresponds to a different freshness degree; storage days one and two are in class 1, storage day three is in class two and data taken after the fourth storage day are mapped in class three.

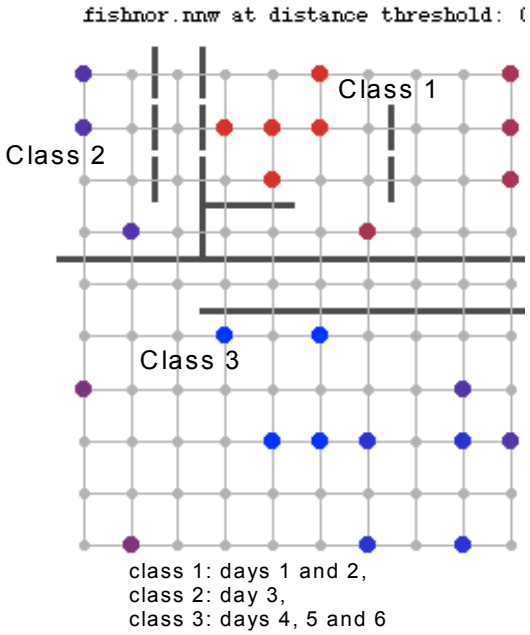

Fig. 6. The figure shows the same SOM of Fig. 5 after fixing a threshold on the distance between codebook vectors. Vector couples separated more than the threshold are considered as belonging to different clusters. Boundaries between clusters are then formed and classes identified. The relation between classes and storage days is also shown.

5. SAMMON'S MAPPING

Sammons mapping is a non-linear mapping introduced by Sammons at the end of the sixties as a solution to the problem of optimisation of the signals space in telecommunications [31]. It consists in a 2D representation of vectors belonging to an *N*-dimensional space with the constraint of preserving the reciprocal distances between the vectors expressed through the distance matrix: $d_{ij} = \left\| \mathbf{x}_i - \mathbf{x}_j \right\|$. The principle is to place the data points onto a plane in such a way that all their mutual Euclidean distances have, as close as possible, the corresponding value d_{ij}.

Given a data matrix **X**, the most convenient way to calculate the Sammon's mapping is to use an iterative optimisation where the new set of data **x** is such that the following error function is minimised by some steepest descent iterative process [32]:

$$E = \frac{1}{\sum_i \sum_{j<1} d_{ij}^*} \sum_i \sum_{j<i} \frac{\left(d_{ij} - d_{ij}^*\right)^2}{d_{ij}^*}. \tag{6}$$

Sammon's mapping provides an easy and fast way to graphically display the clustering properties of a data set.

6. CLUSTER ANALYSIS

In cluster analysis no assumptions are made about the number of classes and their structure (statistical distribution). The grouping is done on the basis of similarities, namely distances.

A number of different clustering techniques were developed [33], here *agglomerative hierarchical clustering* is discussed.

The starting point of the method is the similarities matrix of the data set previously defined. Distance can be defined in several ways, the most common class of distances is the generalised distance which contains the Euclidean distance as a particular case:

$$d(x,y) = \left(\sum_i |x_i - y_i|^{1/m} \right), \; m = 2 \Rightarrow \text{Euclidean distance.} \tag{7}$$

Cluster analysis starts from individual data points, thus at the beginning there are as many clusters as data points. Most similar objects are first grouped, and these initial groups are merged according to their similarities. As the similarity decreases, all subgroups are fused into a single cluster. The results are displayed in a two-dimensional diagram known as a *dendrogram*. It illustrates the mergers which occur at successive levels.

Like Sammon's mapping, cluster analysis is a simple way to study the intrinsic occurrence of classes in the data set showing the capability of an electronic nose to automatically recognise different classes.

7. SUPERVISED CLASSIFICATION

In supervised classification the classes are a priori defined both as kind and number. This information has to be acquired from other considerations about the application under study.

Once classes are defined, supervised classification may be described as the search of a model of the following kind:

$$\mathbf{c} = f(\mathbf{s}), \tag{8}$$

where: $\mathbf{c}$ - a vector describing the class assignment, $\mathbf{s}$ - the vector of features of the sensors in the array, f - a generic function.

To solve pattern recognition problems class memberships must be encoded in a numerical form that allows treating the problem by numerical methods. The most common way to express class memberships is the so-called "one-of-many" code. In this codification, the dimension of **c** is equal to the number of classes. The component corresponding to the class at which a sample belongs is settled to 1, leaving 0 to the others.

In electronic nose applications various sources of measurement errors may occur, and as a consequence, Eq. 8 is written in a more realistic form as:

$$\mathbf{c} = f(\mathbf{s}) + \mathbf{e}\,, \tag{9}$$

where the vector **e** contains everything is not related to the classification scheme expressed by the vector **c**.

Equation 9 is formally similar to the general problem of regression where the scope is the determination of the function *f*, in terms of functional form and parameters. Statistics provides the tools to estimate, from an experimental data set, the parameters of the function *f*, in order to approximate the measured experimental data. The classical approach is the Least Squares Method. Many practical algorithms were proposed as practical solution. Among them chemometrics and the neural networks are those widely used.

It is important to reflect about the applicability in typical electronic nose experiments of the least squares. Least squares method is based on the assumption that the variables are normally distributed and that the quantity **e** of Eq. 9 is a random variable normally distributed with zero-mean. The assumption of zero-mean means that all the variables to which the sensor responses are sensitive, except those related to the classification of the samples, may fluctuate but not biasing the measurement. Usually this does not hold for electronic nose data where, except the sensors noise, the contributions to **e** (sensors drift, sample temperature variations, sample dilution changes,…) are not zero-mean quantities.

Nevertheless, solutions based on least squares can be used to establish classification models, but it is important to be aware of the fundamental limitations of the methods.

Before to discuss practical solutions, it is necessary to detail the general frame.

In pattern classification it is very important to estimate the expected error rate after the classifier model has been assessed on a calibration data set. The expected error rate evaluates the efficiency of the model giving the probability of misclassifying future samples. The estimation of error rate on the same set used for calibration induces large optimistically biased estimate of the performances. This effect is often called "over-fitting". The importance of over-fitting grows with the order of the regression function. In particular, it is important when highly non-linear functions are used (e.g. in neural networks). In these cases the model may be almost perfect to estimate the data on which the model is estimated failing completely in generalising other data.

Over-fitting may be more important for classification scopes, where the samples are extracted from sets that may realistically be not well defined, so that the assignment of samples to classes may be affected by errors due to vagueness of the classification scheme. In this situation the possibility to generate models not able to predict with sufficient accuracy unknown samples is high.

The straightforward solution for the estimation of error rate is to split the data set in two independent sets, and use one for calibration and the other to test the classifier and estimate the error rate. This method cannot be used when small sample sizes are available. In many practical cases, and electronic nose experiments are often among these, the data sets are small because of

the difficulties to get a highly populated set. In these cases, all the data are necessary to estimate the classification model. Moreover, how to split samples is a non-trivial problem because the division should be done keeping the distributions of the two sets as close as possible in order to avoid biasing evaluation of performances.

A more reliable validation of a model is achieved using the "leave-one-out" technique [34]. "Leave-one-out" repeats n times for n measures the model building, each time leaving one measure out for testing and the rest for training. The average test error rate over n trials is the estimated error rate.

In case of small data sets, the method of bootstrap has been proven to be more efficient than "leave-one-out". Both "leave-one-out" and bootstrap are kinds of resampling methods. Bootstrap method generates new samples (called "bootstrap samples") by drawing, with replacement, a number N of samples from the original samples [35]. Different methods to generate the bootstrap samples are available, a comparison among the four more popular is discussed in ref. [36], where the efficiency of the methods is compared on a classification problem.

8. LINEAR DISCRIMINATION

The simplest way to estimate a supervised model is to consider that the descriptor of each class can be represented as a linear combination of the sensor responses; considering *N* sensors and *M* classes the expressions can be written as:

$$\begin{cases} c_1 = \sum_{j=1}^{N} k_{1j} s_j + e, \\ \dots \\ c_M = \sum_{j=1}^{N} k_{Mj} s_j + e. \end{cases} \tag{10}$$

Geometrically, this means to section the sensor space with straight lines, each bisecting the space. The result is a partitioning of the space in volumes, each defining one class.

Considering a set of *P* experimental data, the previous set of equations can be written in compact matrix form as:

$$\mathbf{c}_{M\times P} = \mathbf{K}_{M\times N}\mathbf{s}_{N\times P} + \mathbf{E}_{M\times P}\,. \tag{11}$$

The matrix $\mathbf{K}_{M\times N}$ containing the model parameters, can then be directly estimated considering the Gauss-Markov theorem to solve least squares solution of generic linear problems written in matrix form [37]:

$$\mathbf{K}_{M\times N} = \mathbf{c}_{M\times P}\mathbf{s}^{+}_{P\times N}\,, \tag{12}$$

where the matrix $\mathbf{s}^{+}_{P\times N}$ is the generalised inverse, or pseudo-inverse, of the matrix $\mathbf{s}_{N\times P}$. The operation of pseudo-inversion generalises the inversion of square matrices to rectangular matrices. This solution is often called in literature as Multiple Linear Regression (MLR).

Once the model is assessed, it allows assigning any unknown samples to one class. Due to the presence of the above mentioned error matrix ($\mathbf{E}_{N\times P}$), the model provides a numerical estimation of the "one-of-many" encoding of class assignment. In practice, something of different from 0 and 1 is obtained. The estimated class assignment vector is called "classification score" and the sample is assigned to the class represented by the component with the higher value. This gives the possibility to evaluate also a sort of goodness of the classification considering either the ratio between the first and the second value of the components of the estimated **c** or the difference between the highest value and 1 (target value).

The components of the matrix $\mathbf{K}_{M\times N}$ define the importance of each sensor in the classification of each class. This information can also be used as the loadings of PCA or the component planes of SOM, to design and optimise the sensor array composition.

The pseudo-inversion, like the inversion of square matrices, is influenced by the partial correlation among the sensors. Chemometrics offers methods to solve problems with colinear sensors, such as Principal Component Regression (PCR) and Partial Least Squares (PLS) [15]. PLS is often used to solve classification problems. In this case, PLS offers not only a more robust solution of the classification problem, but by plotting the latent variables it is possible to graphically represent the class separation. It is worth to mention that the PLS latent variables differ from the PCA factors being evaluated as the eigenvectors of the matrix $\mathbf{S^T C C^T S}$. Geometrically, the PCA components are rotated in order to maximise their correlation with the components of the matrix **C**. Furthermore, PLS loadings can be used, as the PCA loadings, to study the contribution of each sensor to the solution of the classification problem.

In linear discrimination only classes separable by straight lines are correctly classified. Classification improves if non-linear boundaries between classes are used. Figure 7 shows an example where a parabolic function achieves the separation where a linear boundary fails.

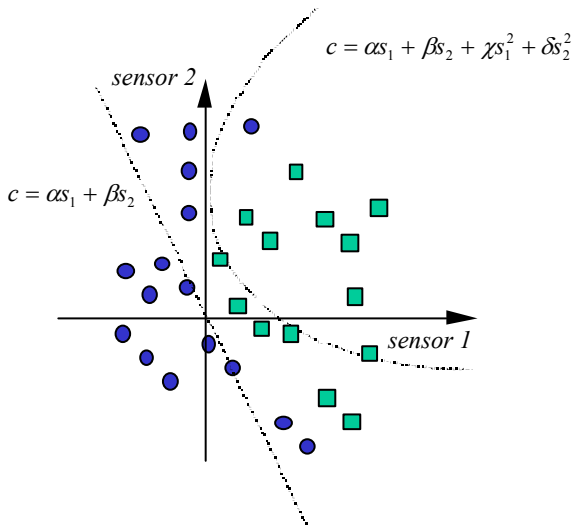

Fig. 7. Geometrical meaning of linear and quadratic discriminant function. As for the polynomial fitting of functions, the increase of the order of the polynomia brings a better fit.

A simple non-linear discriminant analysis can be obtained by a simple modification of the method previously discussed.

As an example, let us consider the following quadratic form such as:

$$\begin{cases} c_1 = \sum_{j=1}^{N} k_{1j} s_j + \sum_{j=1}^{N} h_{1j} s_j^2 + e \\ \cdots \\ c_M = \sum_{j=1}^{N} k_{Mj} s_j + \sum_{j=1}^{N} h_{Mj} s_j^2 + e \end{cases} \tag{13}$$

This system can be written in the compact form of Eq. 7 defining a suitable sensor matrix. Considering P measures, let us define a matrix **T** from the sensor responses, such as:

$$\mathbf{T}_{P\times 2N} = \begin{bmatrix} s_{11} & s_{1N} & s_{11}^2 & s_{1N}^2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ s_{P1} & s_{PN} & s_{P1}^2 & s_{PN}^2 \end{bmatrix} \tag{14}$$

In the same way, the parameters k_{ij} and h_{ij} are joined to form a unique parameter matrix **H**. With these definitions a linear problem may be written like that of Eq. 7. The matrix **H** can then be estimated either by direct pseudo-inversion or by PLS.

It is worth to note that increasing the order of the function the colinearity of the sensor matrix increases and the use of chemometrics methodology (e.g. PLS) becomes more advantageous.

Increasing the order of the discriminant function may solve highly complex classes distribution. The extreme solution is to use a method where the choice of the function is not required. Neural networks offer the possibility to solve the classification problem disregarding the functional form. It is well known that optimised neural networks may reproduce any kind of non-linear function. Neural networks derive all the knowledge from the experimental data, so that, increasing the size of the calibration data-set increases the accuracy of the neural network based classifier.

9. NEURAL NETWORKS

The introduction of alternative techniques based on artificial neural networks (ANN) seemed to open the way for the unravelling of the problems concerning the solution of non-linear problems such as that described in Eq. 9.

ANN began to be developed in the 50's aiming at providing models to study and to reproduce some aspects of the brain working principles. In particular, the ability to learn from experience, to apply the knowledge to predict events and to adapt the knowledge have captured the attention, at the beginning, of physiologists and psychologists. An important leap ahead occurred when also physicists, mathematicians and electronic engineers started to be involved in these researches.

Since that time it became clear that besides the understanding of the working principle of brain, a new set of tools for calculus was going to be available. In a few years several neural models, specialised to solve calculus problems, were proposed and applied in many different fields [38].

A neural network is characterised by three basic elements: the process units (neurons), the network architecture, and a learning rule providing the law according to which the network can learn from experience.

The most used network for the solution of regression problems is the multilayered feed-forward network [39]. The network is based on a neuron model called generalised perceptron. In this model the neuron is a processor unit with many inputs and one output. It is divided in two parts: the first part is a weighted sum of the inputs and the second one is a non-linear transformation of the sum. The weights are optimised according to a learning rule as a function of the networks experience. Several non-linear functions can be used for the non-linear transformation of a perceptron. Among others, the sigmoid and hyperbolic tangent are more frequently used.

In multilayered feed forward neural networks the neurons are arranged in layers, the neurons in the same layers are not connected to each other but all the neurons of one layer feed the neurons of the following layer. The first layer is called input layer, in case of a sensor array application it receives the outputs of the sensors. The last layer is named the output layer and the intermediates are called hidden layers.

The basic learning rule adopted for this network is the back propagation [39]. In order to train the network to learn to solve a particular problem it is necessary to provide the network with a number of sample inputs with their corresponding outputs. The back propagation algorithm works adjusting the neuron weights in order to minimise the difference between the actual network output and the output given by the sample. It is a two step algorithm: in the first the network output is worked out and in the second the weights are adjusted to minimise the error. The sample set is processed until the error reaches a value below a prefixed tolerance.

Table 2. Fundamental steps of the basic back-propagation algorithm applied to a three layer network. W is the weight matrix, *net* vectors are the output of each layer, and T is a generic perceptron transfer fucntion. In the backward propagation errors (δ) are calculated, and weights adjusted to minimise it. S is a free parameter called momentum.

forward	*backward*
1. $net_j = \sum_i W_{ij} a_i$	1. $\delta_k = (t_k - a_k) a_k (1 - a_k)$
2. $a_j = T(net_j)$	2. $\Delta W_{jk} = S \delta_k a_j$
3. $net_k = \sum_j W_{jk} a_j$	3. $\delta_j = a_j (1 - a_k) \sum_k \delta_k W_{jk}$
4. $a_k = T(net_k)$	4. $\Delta W_{ij} = S \delta_j a_j$

The algorithm is sketched in Table 2 in a very basic form for a three layer network. The matrices W_{ij} and W_{jk} are the weights connecting input-hidden layers and hidden-output layer respectively, a_i, a_j and a_k are the outputs of each layer, a_t is the "true" output and S is a convergence parameter called momentum. An alternative to back-propagation is to consider a global error function, having as parameters the weights of the network, and to apply an optimisation procedure such as a gradient descent algorithm to find out the set of weights minimising the global error [40].

Back propagation networks have been utilised, since the first beginnings, in many electronic nose applications [41].

One of the major limitations in the use of this tool is given by the large calibration data set that is necessary to correctly calibrate the network. The number of data is related to the dimensions involved in the problems, to the number of sensors, to the number of classes, and to the complexity of the task that the network is in charge to perform. The number of neurons in the hidden layers is connected with this last aspect.

The number of parameters to be estimated is, for a three layer network, *(m*h)+(h*n)*, where, *m*, *h*, and *n* is the number of neurons in the input, hidden, and output layers respectively. For a reliable estimation, the number of training data cannot be less than this value; a rule-of-thumb, declared by a commercial software, fixes to five times the number of weights the minimum size of a training set to train with a back-propagation algorithm [42].

Due to the high non-linearity of neural networks the possibility to over-fit the data is extremely high. This must to be avoided by testing the network on an independent data set.

As discussed above, electronic nose data often suffers of a poor size. In this situation the problem of having a sufficient size of the training sample and an independent sample on which to test the network collide. Resampling methods may help in this situation. "Leave-one-out" has been shown to produce unbiased estimates of error rate with neural network classifiers in small sample size situation [43]. Bootstrap methods were shown to be usable for neural network classifier validations and with better performances with respect to "leave-one-out" [44].

An application of bootstrap to analyse electronic nose data with neural networks can be found in Ref. 45.

10. CONCLUSIONS

Pattern recognition applied to several disciplines and practical problems produced a huge number of algorithms and techniques that are, in principle, applicable to classify electronic nose data. Several of these techniques were actually utilised and frequently appeared in literature. On the other hand, there is not yet an analytic approach based on natural olfaction paradigms. This makes the field almost totally dependent on the developments achieved in other fields. A comparison among the various techniques has also been attempted by some authors, but since the variables determining electronic nose performances are very numerous (choice of sensors, samples, sampling systems,…), the results achieved are scarcely indicative to give a sure direction valid in general.

Nonetheless some of these techniques became a classic of this field (such as scaling and normalisation, PCA and discriminant analysis) and nonetheless, they are often used without a deep understanding of the hypotheses on which they are based and the potentiality of these methods is not yet fully exploited. As for the other components of electronic noses (e.g. sampling system) a rethinking of the assumptions, the implementations, and the interpretation of the methods and solution adopted is highly advisable for a meaningful improvement in the field.

REFERENCES

1. Persaud K., Dodd G.: *Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose*. Nature, vol. 299, 1982, pp. 352-355.
2. Pearce T.: *Computational parallels between the biological olfactory pathway and its analogue the electronic nose*. Biosystems vol. 41, 1997, pp. 43-67.
3. Di Natale C., Davide F., D'Amico A.: *Pattern recognition in gas sensing*. Sensors and Actuators B vol. 23, 1995, pp. 111-118

4. Kowalski B.R., Wold S.: *In Handbook of statistics* vol. 2, P.R. Krishnaiah and L.N. Kanal eds., North Holland Publ., Amsterdam, The Netherlands, 1982, pp. 673 – 697.
5. Hierlemann A., Schweizer-Berberich M., Weimar U., Pfau A., Göpel W.: *Pattern recognition and multicomponent analysis*. Sensors update vol. 2, W. Göpel, J. Hesse, H. Baltes (eds.), VCH, Weinheim, Germany, 1995.
6. Hines E.L., Llobet E., Gardner J.W.: *Electronic noses: a review of signal processing techniques*. IEE Proc. - Circuits Devices Syst. vol. 146, 1999, pp. 297-310.
7. Jurs P.C., Bakken G.A., McClelland H.E.: *Computational methods for the analysis of chemical sensor array data from volatile analytes*. Chemical Review 2000 100 pp. 2649-2678.
8. Eklöv T., Mårtensson P., Lundström I.: *Selection of variables for interpreting multivariate gas sensor data*. Analytica Chimica Acta vol. 381, 1997, pp. 221-232.
9. Martinelli E., Falconi C., D'Amico A., Di Natale C.: *Feature extraction chemical sensor phase space*. Sensors and Actuators B vol. 95, 2003, pp. 132-139.
10. Martinelli E., Pennazza G., Di Natale C., D'Amico A.: *Chemical sensors clustering with the dynamic moments approach*. Sensors And Actuators B vol. 101, 2004, pp. 346-352.
11. Di Natale C., Paolesse R., Macagnano A., Mantini A., D'Amico A., Legin A., Lvova L., Rudnitskaya A., Vlasov Y.: *Electronic nose and electronic tongue integration for improved classification of clinical and food samples*. Sensors and Actuators B vol. 64, 2000, pp. 15-21.
12. Horner G., Hierold C.: *Gas analysis by partial model building*. Sensors and Actuators B, vol. 2, 1990, pp. 173-184.
13. Di Natale C., Paolesse R., Macagnano A., Troitsky V.I., Berzina T.S., D'Amico A.: *Pattern recognition approach to the study of the interactions between metalloporphyrin Langmuir-Blodgett films and volatile organic compounds*. Analytica Chimica Acta vol. 384, 1999, pp. 249-259.
14. Fukunaga K.: *Introduction to statistical pattern recognition*. Academic Press, New York, USA. 1992.
15. Massart D.L., Vandegiste B.G., Deming S.N., Michotte Y., Kaufmann L.; *Data handling in science and technology*. Chemometrics: a textbook, vol. 2, Elsevier. Amsterdam, The Netherlands. 1988.
16. Grossberg R., Procaccia I.: *Measuring the strangeness of strange attractors*. Physica D, vol. 9, 1983, pp.189-208.
17. Oja E.: *Principal components, minor components, and linear networks*. Neural Networks vol. 5, 1992, pp. 927-935.
18. Cardoso J.F.: *Blind signal separation: statistical principles*. Proc. of IEEE, vol .90, 1998, pp. 2009-2025.
19. Di Natale C., Martinelli E., D'Amico A.: *Counteraction of environmental disturbances of electronic nose data by Independent Component Analysis*. Sensors and Actuators B vol. 82, 2002, pp. 158-165.
20. Kvalheim O.M., Oygard K., Grahl-Nielsen O.: *SIMCA multivariate data analysis of blue mussel components in environmental pollution studies*. Analytica Chimica Acta vol. 150, 1983, pp. 145-152.
21. Hastie T., W. Stuetzle: *Principal curves*. Journal of the American Statistical Association, vol. 84, 1989, pp. 502-516.
22. Pardo M., Sberveglieri G., Gardini S., Dalcanale E.: *A hierarchical classification scheme for an Electronic Nose*, Sensors and Actuators B vol. 69, 2000, pp. 359-365.
23. Kramer H., Matthewes M.: *A linear coding for transmitting a set of correlated signals*. IRE Trans. Inf. Theory, vol. IT-2,1956, pp.41-46.
24. Kohonen T.: *Self Organising Map*. 1995 Springer Verlag, Berlin Germany.
25. Davide F., Di Natale C., D'Amico A.: S*elf organizing multisensor system for odour classification, internal catergorization, adaptation, and drift rejection*. Sensors and Actuators B, vol. 18, 1994, pp. 244 4.
26. Di Natale C., Davide F., D'Amico A., Hierleman A., Schweizer M., Mitrovics J., Weimar U., Göpel W.: *A composed neural network for the recognition of gas mixtures*. Sensors and Actuators B, vol. 25, 1995, pp. 808.
27. Kraus G., Hierleman A., Gauglitz G., Göpel W.: *Analysis of complex gas mixtures by pattern recognition with polymer base quartz m icrobalance sensor arrays*. Technical Digest of Transducers '95 Conference, Stockholm Sweden. 25-29 Jun. 1995, pp. 675-678.
28. Kain A.K. Mao J.: *Artificial neural network for nonlinear projection of multivariate data*. Proc. Of 11-th Int. Conf. On Pattern Recognition, 1992 IEEE Comp. Soc. Press, Los Alamitos, CA, USA, pp. 41.
29. Di Natale C., Macagnano A., D'Amico A., Davide F.: *Electronic nose modeling and data analysis by self organizing neural networks*. Measurement Science and Technology. Vol. 8, 1997, pp. 1-8.
30. Di Natale C., Brunink J.A.J., Bungaro F., Davide F., D'Amico A., Paolesse R., Boschi T., Faccio M., Ferri G.: *Recognition of fish storage time by a metalloporphyrins coated QMB sensor array*. Measurement Science and Technology vol. 7, 1996, pp. 1103-1114.

31. Sammons J.W.: *A non linear mapping for data structure analysis*. IEEE Trans. Comp. vol. C-18, 1969, pp. 401.
32. Dennis J.E., Schnabel R.B.: *Numerical methods for unconstrained optimization and non-linear equations*. Prentice Hall Series in computational mathematics, New York, USA, 1983.
33. Johnson R.A. and Wichern D.W.: *Applied multivariate statistical analysis*. Prentice Hall, Englewood Cliffs, NJ, USA, 1982.
34. Lachenbruck P.A., Mickey R.M.: *Estimation of Error Rates In Discriminant Analysis*. Technometrics vol. 10, 1968, pp. 1-11.
35. Efron B.: *Bootstrap methods: another look at the jack-knife*. Annals of Statistics, vol. 7, 1979, pp. 1-26.
36. Hamamoto Y., Uchimura S., Tomita S.: *A bootstrap technique for nearest neighbor classifier design*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, 1997, pp. 73-79.
37. Campbell S.L., Meyer C.D.: *Generalized inverses of linear transformations*, Pitman, London, UK, 1979.
38. Maren A.J. (editor): *Handbook of neural computing applications*, J. Wiley and sons, London, UK, 1991.
39. Rumelhart D.E., McClelland J.L.; Parallel Distributed Processing: *Explorations in the microstructure of cognition*. vol.1: Learning internal representations by error propagation, MIT Press, Cambridge, MA, USA, 1986.
40. Hertz J., Krogh A., Palmer R.G.: *Introduction to the theory of Neural Computation.* vol.1, Addison Wesley, New York, USA, 1991.
41. Gardner J.W.,. Hines E.L and Wilkinson M.: Measurement Science and Techn., 1990, pp. 446-451.
42. Neural Computing: A technology handbook for Professional II/plus© users, NeuralWare Inc., Pittsburgh, USA, 1993.
43. Martin C.E., Rogers S.K., Ruck D.W.: *Neural network Bayes error estimation*. Proc. IEEE Int. Conf. Neural Network, 1994 pp. 305-308.
44. Ueda N., Nakano R.: Mixture density estiamtion via EM algorithm with deterministic annealing, Proc. IEEE Int. Conf. Neural Network, vol. 1, 1995, pp. 101-105.
45. Di Natale C., Mantini A., Macagnano A., Antuzzi D., Paolesse R., D'Amico A.: *Electronic nose analysis of urine samples containing blood*. Physiological measurements vol. 20, 1999, pp. 1-8.

PRZEGLĄD METOD OBRÓBKI WSTĘPNEJ I ROZPOZNAWANIA OBRAZÓW W SZTUCZNYCH SYSTEMACH OLFAKCYJNYCH.

Streszczenie

Analiza danych stanowi podstawową część sztucznych systemów olfakcyjnych. Pomimo tego, podczas gdy elementy elektryczne i mechaniczne nosów elektronicznych są badane dogłębnie i optymalizowane dla potrzeb konkretnych zastosowań, analizę danych przeprowadza się niekiedy bez jakiegokolwiek krytycznego rozpatrywania hipotez, na których jest oparta.

W badaniach tej dziedziny wiedzy zastosowano kilka metodologii dla skorelowania danych o nosie elektronicznym z informacją potrzebną dla celów klasyfikacji. Chemometria i sieci neuronowe stanowią źródło najbardziej popularnych metod będących obecnie w użyciu.

W niniejszej pracy dokonano przeglądu metod najczęściej stosowanych w analizie danych i omawia się ich stosowanie do danych z nosów elektronicznych. Chociaż omówienie dotyczy zespołów czujników chemicznych, wnioski dotyczą i są słuszne dla rodzajów i szerszego rzędu systemów czujników.