EOS Encyclopedia of SENSORS
www.aspbs.com/eos

Pattern Recognition Techniques for Odor Discrimination in Gas Sensor Array

Amine Bermak[1], Sofiane Brahim Belhouari[1], Minghua Shi[1], Dominique Martinez[2]

[1]*Hong Kong University of Science and Technology, EEE Department, Clear Water Bay, Kowloon, Hong Kong*

[2]*LORIA, Campus Scientifique, 54506 Vandoeuvre-Les-Nancy, France*

CONTENTS

1. OVERVIEW

In the past decade there has been a growing interest for the development of olfactory machines and electronic nose systems. This surge has been mainly driven by a variety of real-life applications. Indeed, the problem of classifying and further quantifying chemical substances on a real-time basis is very critical for a wide range of applications in the industrial and civil environments. The ability to monitor and precisely measure leakages of combustible and explosive gases [1] is crucial in preventing the occurrence of accidental explosions. Accordingly, the development of gas sensors for the detection of single gases (such as CO, CH_4, H_2, SO_2, NO_x, O_3 etc.) has seen an increasing interest within the research community. Environmental applications including air quality control and pollution monitoring [2–4] are experiencing a steadily increasing attention as the need to protect the environment has grown over the last years [5]. Quality control and agricultural applications (food, drinks agricultural products quality control, soil contamination, etc.) are also on the rise [6–8]. New and exciting developments have also recently emerged in the area of medical applications (detection of infections, diseases and bacteria) [9–11], mobile robot navigation [12] as well as space applications (monitoring of air quality and gas detection in space shuttle) [13]. The last decade has also witnessed an increasing interest in modeling the biological olfactory systems and building bio-inspired electronic nose [14–18]. Table 1 illustrate an exhaustive description of the most recent as well as traditional applications for gas sensors. The table also reports the gases involved as well as the gas sensors used for each application.

Sensors that can transform different gases into measurable electrical signals are fundamentally required in the upper mentioned application-specific sensing systems. Various odor sensors have been reported in the literature [5, 19, 20]. All sensors are composed of a chemically sensitive material interfaced to a transducer (Fig. 1) [21]. Chemical species, which represent the inputs, are measured by allowing the analyte molecules to interact with the chemically sensitive material of the sensor. This will generate some physical change which is sensed by the transducer and converted to an output signal $x(t)$ at a given time t. This interaction between the measurand (analyte molecule) and the gas sensitive material layer of the sensor can be either reversible (measurand dissociate from the layer when the external concentration is removed) or irreversible (measurand undergo a chemical reaction and the sensitive material layer is consumed) [21]. A large number of gas sensitive materials have been used in the past [5, 21] which resulted in a very large variety of gas sensors including semiconductor, piezoelectric, optical, catalytic as well as electrochemical gas sensors [5]. Microelectronic gas sensors in particular, have received an increased attention in recent years, due to their numerous advantages including small size, high sensitivities in detecting very low concentrations, possibility of on-line operation and low-cost fabrication, making them attractive for consumer applications. Unfortunately, all gas sensors suffer from a number of shortcomings such as lack of selectivity, nonlinearities of the sensor's response and long-term drift. Pattern recognition algorithms combined with a gas sensor array have been traditionally used to address these issues

ISBN: 1-58883-056-X/$50.00

Encyclopedia of Sensors
Edited by C. A. Grimes, E. C. Dickey, and M. V. Pishko
Volume X: Pages (1–17)

Table 1. Review of the most recent and traditional gas sensors applications illustrating the gases involved and the gas sensors used. NA stands for Not Available. TGS stands for the commercial Figaro sensors.

Applications	Smells/gases involved	Examples of gas sensors	Ref.
Safety	H_2, CO, NO_2, NH_3	TGS-109-203-311-800-812	[22–26]
• Leak detection	CH_4, $CHCl_3$, H_2S,	TGS-813-815-823-832-842	[27–31]
• Poisonous gases	C_2H_5OH, C_4H_{10},	TGS-2600-2610-2611-2620	[32–36]
• Dangerous and	CH_3SH, $(CH_3)_3N$,	ZnO/Al_2O_3,SnO_2/Pd, ZnO	[37–41]
combustible gases	C_2H_6O, C_3H_8, C_6H_6	SnO_2-/ZnO-/CdS-Pt-/La_2O_3	[42–46]
	CF_3CH_2F, CH_4O	WO_3/Au, MOS, MOSFET	[47–49]
Environmental control		Gradient sensor microarray	
• Air quality control	CO, NO_x, SO_2, CH_4	TGS, MOS, MOSFET	[2–4, 50, 51]
• Combustion atmosph.	C_6H_6, C_4H_{10}, C_3H_8	Tin oxide doped Pd/Al/Pt,	[52–54]
• Contaminated soils		and different film thickness	
Quality control	Coffee, vinegar,	TGS-813-880-822-825-812	[6–8, 55, 56]
• Food quality control	perfume, aroma,	SnO_2-/Sb_2O_3-/ZnO-/NiO	[57–61]
• Perfume and aroma	diary products	Polymer, MOSFET, Quartz-res.	[62–66]
• Monitoring cultivation	wine, etc.	Thin-film WO_3/Pd-/-/Bi-/Sb	[67–70]
Medical		Quartz microbalance (QMB),	
• Infection detection	NA	Metal oxide semiconductor	[9–11]
• Bacterial detection		Electrochemical sensors	
Mobile robot navigation	CO, NO_x, SO_2, CH_4	TGS-2600, 2610, 2611, 2181	[12]
Space	NA	Polymers and quartz	[13]

[6]. In fact, an array of different gas sensors is used to generate a unique signature for each odor. A gas sensor array permits to improve the selectivity of the single gas sensor, and shows the ability to classify different odors and to quantify components concentrations. The aim of the pattern recognition techniques is to find a relationship between the sensors outputs and the odor class (or concentration). To achieve this, first some features have to be extracted from the sensors responses and then the functional relationship between the feature vectors and the class labels has to be derived by a learning procedure. Therefore, an electronic nose consists of an array of gas sensors, signal preprocessing, and a pattern recognition algorithm, as illustrated in Fig. 1.

The pattern recognition problem for real life applications of electronic noses remains challenging due to the temporal variability of the instrument, the large intra-class variability compared to a small inter-class separation and the small amount of data available. This is typically the case in the gas sensor area where it is very costly and time-consuming to obtain a large, reliable and representative set of examples. Reliable and robust pattern recognition algorithms are therefore needed to address the learning problem in electronic noses. In this chapter, we provide an overview of the most relevant pattern recognition techniques as well as biologically inspired processing for odor discrimination in gas sensor array. The algorithmic part of an odor discrimination system consists of three steps: (i) signal conditioning and feature extraction, (ii) feature reduction and (iii) classification. The role of the first step is to segment the pattern of interest from the background, remove noise, normalize the pattern, and any other operation that contributes in defining a compact representation of the pattern. Feature reduction should provide a small number of informative features in order to make the learning task simpler. Classification tasks address the problem of identifying unknown sample as one from a set of recognizable gases.

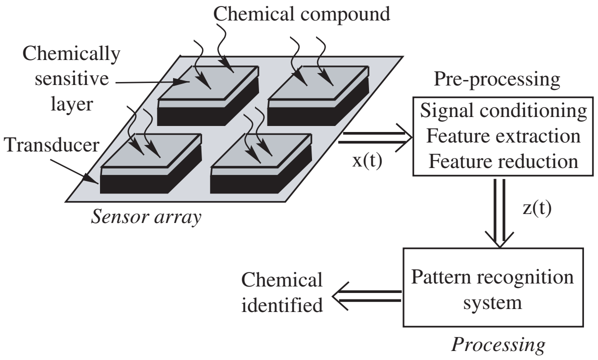

Figure 1. Electronic nose block diagram.

2. SIGNAL CONDITIONING AND FEATURE EXTRACTION

This data preprocessing stage operates on the gas sensor responses in a way that improves the overall pattern analysis performance. It can be achieved by extracting parameters that are descriptive of the sensor array responses. Thus, the raw data are transformed into a characteristic feature vector. Although signal preprocessing will depend on the application, a series of steps are commonly carried out.

2.1. Signals Handling

The common method used to characterize gas sensors is to use a periodic "*exposure-cleaning*" operation. Gases are injected over time in a periodical manner and each gas injection is followed by a cleaning phase which can be done in some cases by injecting dry air. Figure 2 shows a typical response of the sensor for different concentrations.

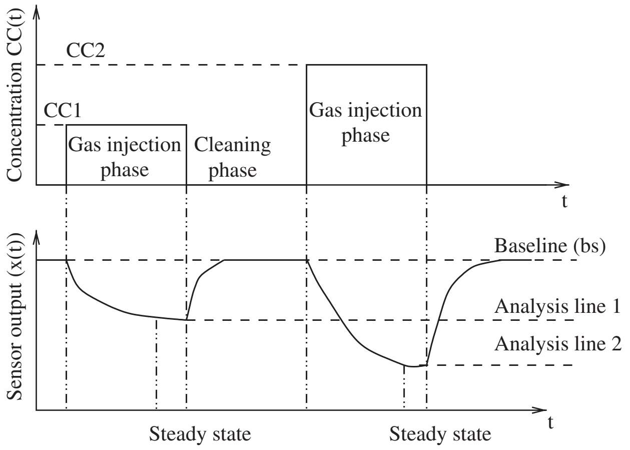

Figure 2. Typical response of a gas sensor.

After the cleaning phase, the signal is settled at the base line value which is the reference measure before gas injection. Most e-noses use as raw signals the temporal responses when switching from the baseline to the analysis line. Usually these signals appear contaminated with diverse type of noises. Therefore, appropriate filters are used to provide a maximum noise rejection. Among the most serious limitation of actual gas sensors is the drift problem, which shows significant temporal variations of the sensor response when exposed to identical atmospheres. Drift problem can be explained as a random temporal variation of the sensor response when exposed to the same gases under identical conditions. It can affect both the baseline (additive) and the sensitivity of the sensor (multiplicative). Figure 3 illustrates an example of an additive drift problem in which we have reported the real response of the sensor as function of the concentration of gases periodically injected into a gas chamber in which the sensor is being placed. We can note that the baseline response of the sensor is shifted which complicates the classification problem.

Baseline manipulation procedures transform the sensor response relative to its baseline to reduce the effects of sensor drift. Three baseline methods are commonly used: difference, relative and fractional [71, 72]. For classification purposes, it is customary to include data normalization across the sensor array in order to reduce the pattern dispersion induced by concentration changes. The most widely used methods in odor discrimination systems are: vector normalization, vector auto-scaling and dimension auto-scaling [71].

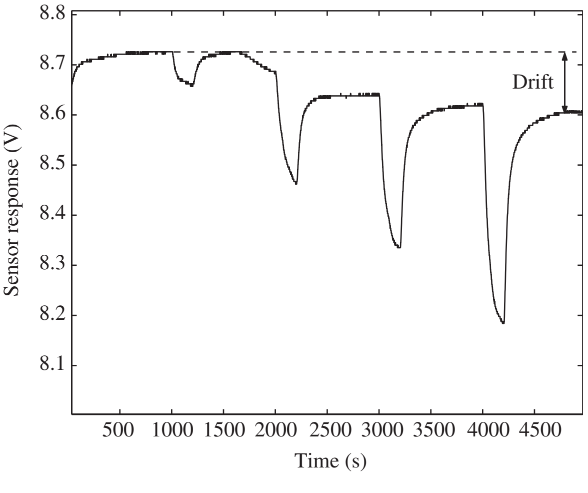

Figure 3. Additive drift affecting the sensor baseline.

2.2. Steady State or Transient Information

Numerous preprocessing techniques have been proposed in the literature [71, 72] to generate descriptive parameters from the sensor array responses. The most common procedure uses the steady state of the sensors' response as a feature vector and ignores the transient response [73]. A number of compression algorithms have been proposed to extract additional information from the transient response, resulting in improved selectivity and increased recognition accuracy [74]. In [75], the transient response has been modeled as the sum of exponential functions and their estimated parameters form the feature vector. However, the extraction of the features is more complex than the classical method. Solutions employing model fitting [76] or parameter extraction technique [74] normally require complicated analysis of the whole dataset. Therefore the need of complex architectures and algorithms due to the large number of parameters involved. A sub-sampling technique of the complete transient response has been tested with significant success for the task of odor classification [71]. A new method based on wavelet transformation has been described in [77]. The feature vector is obtained from the coefficients approximation of multi-resolution analysis with wavelet descriptors. The performed measurements have shown that the use of the wavelet improves notably the recognition system capability. Data extraction using both the steady state and dynamic response of the sensor has been widely used for gas sensors applications.

In [37, 48, 59], the authors described techniques for extracting and using both the steady-state, the slope as well as the transient response information from the sensor's response.

In [68], the authors used dynamic signal extraction techniques and optimal array configuration to improve the classification performance. In [78], a temporal window is used to extract the dynamic information of a gas sensor. In [79], two simple measures that attempt to capture some of the transient information, have been proposed. The first one is based on the use of the transient integrals with respect to time. While the second measure uses the steady state value and the voltage dynamic slope of the time-dependent response. In general, preprocessing the transient response is shown to convey useful recognition information to the classifier. It is important to note that whether the transient information or the steady state information is used, it is typically obtained by fixing the detection parameters of the sensor such as the heating temperature to the optimal value. Recently, there has been a number of studies which investigate the possibilities of modulating these parameters in order to increase the feature space particularly when using a limited number of sensors. In [17], Raman and Gutierrez-Osuna have created virtual sensors from single temperature modulated MOS sensors by varying the operating temperature. The temperature modulated technique is particularly interesting for metal oxide sensors as their selectivity is greatly influenced by the operating temperature [80]. A survey on temperature modulation can be found in [81].

3. FEATURE REDUCTION

The initial feature vector is often projected onto a lower dimensional space in order to avoid problems associated with high dimensionality and redundancy. The use of dimensionality reduction depends on the relationship between the size of the training set and the number of features. If the number of training examples is very large, then the classification error does not increase as the number of features increases. However, if the number of training examples is small relative to the number of features, a dimensionality reduction technique is needed to guarantee an acceptable classification accuracy. Since only a limited number of examples are typically available, there is an optimal number of feature dimensions beyond which the classifier performance starts to degrade. Mapping methods can create new features based on transformations or combinations of the original feature. While, the feature selection algorithms select the best subset of the input feature set [82].

3.1. Mapping Methods

The goal of mapping methods is to find a low dimensional vector z that preserves most of the information in the original feature vector $\boldsymbol{x}$:

$$\boldsymbol{f} : \boldsymbol{x} \in R^d \rightarrow z \in R^q (q < d) \tag{1}$$

Hence, a projection method is needed for mapping data to a lower dimension space. The aim of such a technique is to determine $\boldsymbol{f}$, which optimizes a given criterion. In the case of dimensionality reduction techniques belonging to the linear transformation family, the mapping is given by:

$$z = \boldsymbol{T}\boldsymbol{x}$$

Most feature extraction techniques for electronic nose applications have been based on linear techniques, mainly Principal Components Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA). Advanced techniques now encompass Independent Component Analysis (ICA) and Blind Source Separation (BSS) or Neuroscale.

- Principal Components Analysis: PCA is a linear transformation that preserves as much data variance as possible. PCA chooses $\boldsymbol{T}$ that minimizes the mean squared distance between original data and those reconstructed from reduced data. It is shown that:

  $$\boldsymbol{T}_{\text{PCA}} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1/2}$$

  where $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ are, respectively, the eigenvectors matrix and the diagonal eigenvalues matrix of the data covariance matrix. PCA was very widely used for gas sensors applications [67, 83, 84]. Later in this section, PCA will be compared with other feature reduction techniques.
- Linear Discriminant Analysis: LDA provides a linear projection of the data with $(c - 1)$ dimensions, by taking account of the scatter of data within each class and across classes. Projection directions are those that maximize the inter-class separation of the projected data. The LDA transformation matrix is given by:

  $$\boldsymbol{T}_{\text{LDA}} = \boldsymbol{S}_w\boldsymbol{\Lambda}_w^{-1/2}\boldsymbol{S}_B$$

  where $\boldsymbol{S}_w$ and $\boldsymbol{\Lambda}_w$ are, respectively, the eigenvectors matrix and the diagonal eigenvalues matrix of the within-class scatter $\boldsymbol{W}$. $\boldsymbol{S}_B$ is the eigenvectors matrix of the between class scatter $\boldsymbol{B}$. LDA was previously used for gas detection application [61, 85]. This work also discusses the possibilities of combining different sensors for E-Nose applications. Later in this section, LDA will be compared with other feature reduction techniques.
- Blind Source Separation (BSS) and Independent Component Analysis (ICA): The original feature vector x extracted from the sensor array can be seen as a linear mixture of sources

  $$\boldsymbol{x} = \boldsymbol{MMs}$$

  where MM is the mixing matrix. In electronic noses, the sources s can be different odors but can also be the environmental conditions such as temperature or humidity that affect the sensor responses. The objective of BSS algorithms is to find the inverse of the unknown mixing matrix in order to reconstruct the different sources s from the sensor observations x alone: $z = Tx = s$ when $T = MM^{-1}$. If one considers that the original sources are statistically independent, then ICA provides T that maximize the statistical independence of the projected data $z = Tx$. BSS-ICA algorithms have been successfully applied in electronic noses [86–88]. However, assuming a linear mixing model is not completely satisfactory because the responses of gas sensors are known to be non-linear. In [89], a non-linear transformation was used prior to the linear BSS-ICA algorithm in order to inverse the nonlinearity of the sensor responses. Setting this nonlinearity however assumes certain knowledge of the sensor responses. To overcome this drawback, a nonlinear BSS algorithm has been developed and successfully applied to gas sensor data [90].
- Neuroscale: Neuroscale is a non-linear topography (i.e., distance preserving) projection method. The concept of data topography is assumed to be captured by the interpoint distances, usually measured with a Euclidean metric:

  $$D^*_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$

  Each data point $\boldsymbol{x}_i$ is projected by radial basis function (RBF) to a point z_i, which minimizes the Sammon stress metric [91]:

  $$E_s = \sum_{i=1}^{N}\sum_{j>i}^{N}(D_{ij} - D^*_{ij})^2$$

  where the distance between z_i and z_j is denoted by D_{ij}.

The points z are generated by the RBF given the data point inputs. Thus, $z_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{W})$, where $\boldsymbol{f}$ is the non-linear transformation effected by the RBF with parameters (weights) $\boldsymbol{W}$. We use a non-linear optimization algorithm to find the optimal weights minimizing the Sammon stress metric. This method presents the advantage of preserving the data structure, as well as the possibility of incorporating subjective information. In fact, one useful change can be made to the distance measure in order to generate a feature space that separates classes. For example, if each data point $\boldsymbol{x}_i$ belongs to a known class C_i, a dissimilarity measure s_{ij} can be defined by: $s_{ij} = 1$ if $C_i \neq C_j$ and 0 elsewhere. This can then be incorporated into the distance measure:

$$\delta_{ij} = (1 - \alpha) D^*_{ij} + \alpha s_{ij}$$

where the parameter $\alpha \in [0, 1]$ controls the degree of supervisory information in the mapping. Intermediate values of α allow the points to be projected so as to retain the distance structure with extra separation from the classes.

Other methods, such as Kernel PCA [92], independent component analysis (ICA) [93] and projection pursuit [94] are more appropriate for non-Gaussian distributions. Recently, ICA has been successfully used for odor discrimination with excellent performance [49]. The described experiments show that ICA is capable of handling sensor drift combined with improved discrimination, dimensionality reduction, and more adequate data representation when compared to PCA.

Figure 4 illustrates the performance of PCA, LDA and Neuroscale using a dataset collected on a sensor array with 8 microelectronic gas sensors [85]. The dataset consisted of three gases: methane, carbon monoxide and their mixture. 168 patterns were extracted from the sensor array. Figure 4(a) presents the first two PCA scores for all the studied gas sensors steady state voltage. We can note that the decision boundaries are not well defined due to a strong overlapping. In [23] new technique based on discriminant factorial analysis (DFA) is proposed. DFA was also combined with PCA [33] and applied on both steady state and dynamic slope in order to reduce the drift effect of the sensors [35].

Figure 4(b) shows the Neuroscale plots for the studied gases. Compared to PCA results, it is clear from Fig. 4(b), that the Neuroscale method permits to considerably reduce the overlapping between the classes and hence shows better separability performance. Compared to PCA and Neuroscale results, LDA presents the most discriminatory projection as evidenced by Fig. 4(c).

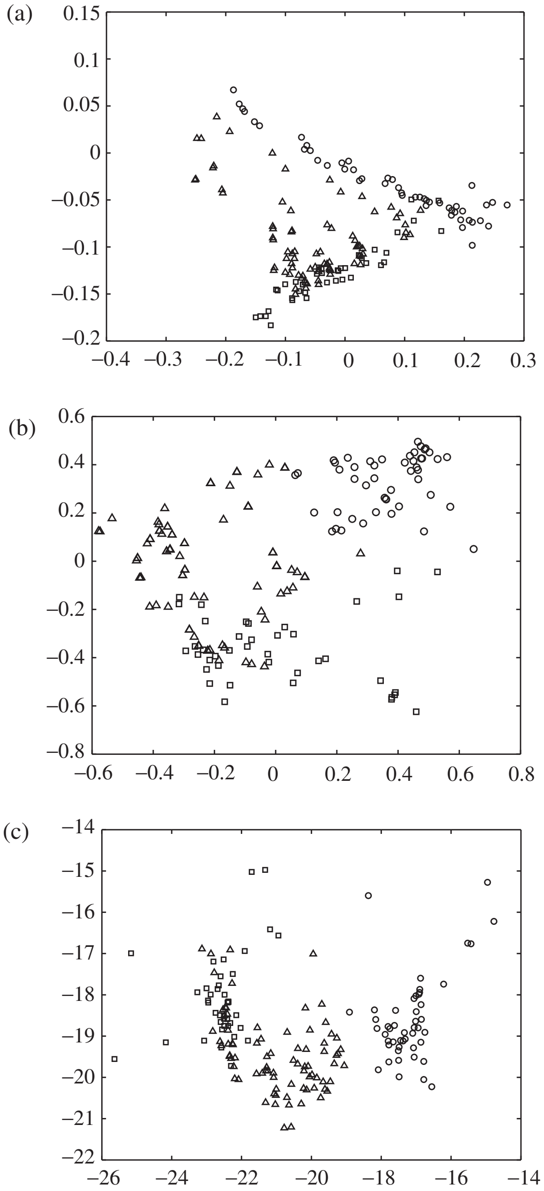

Figure 4. (a) PCA results for the gas sensor array steady state voltage. (b) Data projected using Neuroscale with class information. (c) LDA results for the gas sensor array steady state voltage. Measurement type, CO (circles), CH_4 (squares) and mixture (triangles).

3.2. Feature Selection

The problem of feature selection can be defined as that of selecting a subset of features that achieve the best classification performance. The objective is to find a subset $\hat{F} (\hat{F} \subseteq X)$ containing few features that minimizes a selection criterion $J(F)$:

$$\hat{F} = \arg \min_{F \subseteq X} J(F) \tag{2}$$

Classification error can be used as the selection criterion since the focus is on classifying different gases. Several search methods have been used to explore efficiently the feature space. A systematic procedure must be found for searching through candidate subsets of features. Exhaustive search of all possible subsets of features is in general the only approach, which is guaranteed to find the optimal subset. However, the number of possible subsets grows combinatorially, making this exhaustive search impractical for even moderate values of the cardinality of $\hat{F}$ and X [82]. The only optimal method which avoids the exhaustive search is based on the branch and bound algorithm [95]. The simplest suboptimal method is sequential forward selection (SFS) [96]. This procedure starts from the empty set and sequentially adds features that achieve the lowest value for the selection criterion J. We can also use sequential backward selection (SBS) [97] which starts from the full set and sequentially removes features. These methods provide suboptimal solution with a low computational cost. More sophisticated techniques have been proposed to improve the performance of SFS and SBS by using sequential floating search methods [98]. Others search methods are discussed in [82, 99].

4. STATISTICALLY BASED CLASSIFIERS

The objective of pattern recognition is to set a decision rule, which optimally partitions the data space into c regions, one for each class C_k. The boundaries between regions are the separating surfaces or decision boundaries. A pattern classifier generates a class label for an unknown feature vector $\boldsymbol{x} \in R^d$ from a discrete set of previously learned classes. The most general classification approach is to use the posterior probability of class membership $\wp(C_k|\boldsymbol{x})$. To minimize the probability of misclassification one should consider the maximum a posterior rule and assign $\boldsymbol{x}$ to class $C_{\hat{k}}$ [100]:

$$C_{\hat{k}} = \arg \max_{\{1,\ldots,c\}} [\wp(C_k|\boldsymbol{x})] = \arg \max_{\{1,\ldots,c\}} [\wp(\boldsymbol{x}|C_k)\wp(C_k)] \tag{3}$$

where $\wp(\boldsymbol{x}|C_k)$ is the class-conditional density and $\wp(C_k)$ is the prior probability. In the absence of prior knowledge, $\wp(C_k)$ can be approximated by the relative frequency of examples in the dataset. One way to build a classifier is to develop a model that estimates the posterior probabilities directly, where the boundaries are learnt from data. An alternative is to estimate the class-conditional densities by using representation models for how each pattern class populates the feature space.

In this approach, classifier systems are built by considering each of the classes in turn, and estimating the corresponding class-conditional densities $\wp(\boldsymbol{x}|C_k)$ from data. Methods for estimating probability density can be divided into parametric and nonparametric methods. In parametric methods a specific functional form for the density is assumed, such as unimodal gaussian density. This contains a number of parameters which are then estimated from training data. Such approaches might be incapable of providing an accurate representation of the true density.

4.1. *K* Nearest Neighbor

The most widely used method of nonparametric density estimation is the K Nearest Neighbors (KNN). KNN rule is a powerful technique that can be used to generate highly nonlinear classification with limited data. To classify a pattern $\boldsymbol{x}$, we find the closest K examples in the dataset and select the predominant class $C_{\hat{k}}$ among those K neighbors. Despite the simplicity of the algorithm, it often performs very well and is an important benchmark method. However, one drawback of KNN is that all the training data must be stored, and a large amount of processing is needed to evaluate the density for a new input pattern. KNN was used in electronic nose applications for classification [27] as well as quantification of the concentration of the gases [101]. In [27], KNN was combined with 4 different transient analysis techniques used to analyze the response of temperature modulated sensors.

4.2. Density Models

An alternative to KNN and quadratic classifiers, is to combine the advantages of both parametric and nonparametric methods, by allowing a very general class of functional forms in which the number of adaptive parameters can be increased to build more flexible models. This leads us to a powerful technique for density estimation, called mixture model [102]. More extensive discussion on density estimation can be found in [102–104]. We present briefly three density models namely: (i) Gaussian mixture models, (ii) generative topographic mapping and (iii) probabilistic PCA mixture.

4.2.1. Gaussian Mixture Models

Gaussian mixture model (GMM) is classed as a semiparametric density estimation method since it defines a very general class of functional forms for the density model. In a mixture model, a probability density function is expressed as a linear combination of basis functions. A model with M components is described as mixture distribution [102]:

$$\wp(\boldsymbol{x}) = \sum_{j=1}^{M} \wp(j)\wp(\boldsymbol{x}|j) \tag{4}$$

where $\wp(j)$ are the mixing coefficients and the parameters of the component density functions $\wp(\boldsymbol{x}|j)$ vary with j. Each mixture component is defined by a Gaussian parametric distribution in d dimensional space:

$$\wp(\boldsymbol{x}|j) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^{\top}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right\}$$

The parameters to be estimated are the mixing coefficients $\wp(j)$, the covariance matrix $\boldsymbol{\Sigma}_j$ and the mean vector $\boldsymbol{\mu}_j$. The method for training mixture model is based on maximizing the data likelihood. The log likelihood of the dataset $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, which is treated as an error function, is defined by:

$$l = \sum_{i=1}^{n} \log \wp(\boldsymbol{x}_i)$$

A specialized method is commonly used to produce optimum parameters, known as the expectation-maximization (EM) algorithm [105]. The EM algorithm iteratively modifies the model parameters starting from the initial iteration $k = 0$. For GMM, the EM optimization can be carried out analytically with a simple set of equations [105], where the mixing coefficients are estimated by:

$$\wp^{k+1}(j) = \frac{1}{n}\sum_{i=1}^{n} \wp^{k}(j|\boldsymbol{x}_i)$$

and the estimate for the means for each component is given by:

$$\boldsymbol{\mu}_j^{k+1} = \frac{\sum_{i=1}^{n} \wp^{k}(j|\boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{n} \wp^{k}(j|\boldsymbol{x}_i)}$$

and, finally, the update equation for the covariance matrix is:

$$\boldsymbol{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{n} \wp^k(j|\boldsymbol{x}_i)(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{k+1})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{k+1})^\top}{\sum_{i=1}^{n} \wp^k(j|\boldsymbol{x}_i)}$$

Although the class-conditional distributions in feature space are generally non-Gaussian, the resulting multi-modal approximation is remarkably accurate. GMMs have been successfully applied for a number of applications such as speech recognition [106] and image retrieval [107].

4.2.2. Generative Topographic Mapping

In many classification problems we have to deal with high dimensional data. Therefore we would like to model the distribution $\wp(\boldsymbol{x})$ parameterized by latent variables z in low dimensional space. After estimating $\wp(\boldsymbol{x}|z)$, the dependence on z has to be integrated out to obtain the density in data space $\wp(\boldsymbol{x})$, where:

$$\wp(\boldsymbol{x}) = \int \wp(\boldsymbol{x}|z)\wp(z)\,dz \tag{5}$$

The Generative topographic mapping (GTM) [108] is one of the more popular methods for dealing with this situation. It is a mixture model, which means Eq. (5) is approximated by a sum over M Gaussians:

$$\wp(\boldsymbol{x}) = \frac{1}{M}\sum_{j=1}^{M} \wp(\boldsymbol{x}|z_j)$$

$\wp(z)$ is assumed to be a uniform distribution and each mixture component is a spherical Gaussian with variance σ^2, and the jth center is given by a parameterized mapping $\boldsymbol{y}(z_j, \boldsymbol{W})$. Equation (6) can be rewritten as:

$$\wp(\boldsymbol{x}|\boldsymbol{W},\sigma) = \frac{1}{M(2\pi\sigma^2)^{d/2}}\sum_{j=1}^{M}\exp\left\{-\frac{\|\boldsymbol{y}(z_j,\boldsymbol{W}) - \boldsymbol{x}\|^2}{2\sigma^2}\right\}$$

It is a constrained mixture model because the centers are not independent but are related by the mapping $\boldsymbol{y}$. In GTM method, the mapping from z to $\boldsymbol{x}$ is modeled with a RBF model (Section 4.3.3).

The log likelihood for a dataset $\boldsymbol{x}_i, i = 1, \ldots, n$ is given by:

$$l(\boldsymbol{W},\sigma) = \sum_{i=1}^{n}\log\left(\frac{1}{M}\sum_{j=1}^{M}\wp(\boldsymbol{x}_i|z_j,\boldsymbol{W},\sigma)\right)$$

This log likelihood is then maximized in terms of $\boldsymbol{W}$ and σ using an expectation-maximization (EM) algorithm.

4.2.3. Probabilistic PCA Mixture

Classical PCA is made into a density model by using a latent variable approach, in which the data $\boldsymbol{x}$ is generated by a linear combination of number of variables z of low dimension ($q < d$). The mapping from z to $\boldsymbol{x}$ is given by:

$$\boldsymbol{y}(z,\boldsymbol{W}) = \boldsymbol{W}z + \boldsymbol{\mu} \tag{6}$$

$\boldsymbol{\mu}$ represents the data mean. The probability model of PCA can be written as a combination of the conditional distribution [109]:

$$\wp(\boldsymbol{x}|z,\boldsymbol{W},\sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left\{-\frac{\|\boldsymbol{x} - \boldsymbol{W}z - \boldsymbol{\mu}\|^2}{2\sigma^2}\right\}$$

and the latent variable distribution:

$$\wp(z) = \frac{1}{(2\pi)^{q/2}}\exp\left\{-\frac{zz^\top}{2}\right\}$$

By integrating out the latent variables z, we obtain the distribution of the observed data, which is also Gaussian:

$$\wp(\boldsymbol{x}|\boldsymbol{W},\sigma) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{C}|^{1/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top\boldsymbol{C}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right\}$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^\top + \sigma^2\boldsymbol{I}$. The covariance matrix is the sum of two terms: one is diagonal in a q-dimensional subspace spanned by the first q principal components and the other is spherical. A mixture of PPCA has the same form as (4), where each component density function is given by a probabilistic PCA. Hence, the training of such a model can be done in the maximum likelihood framework using an EM algorithm.

Minimum Description Length (MDL) criterion is able to select an optimal number of components in the density model and so partition the dataset. MDL derived by Risanen [110] from an information theoretic perspective. It is defined via

$$\mathrm{MDL}(M) = -\hat{l}_M + \frac{1}{2}N_p(M)\ln(n) \tag{7}$$

where $\hat{l}_M$ is the maximized log-likelihood of the dataset and $N_p(M)$ is the number of parameters in the M Gaussian model. The optimal number of components $\hat{M}$ is then defined as the minimizer of this cost function, which include the maximized log-likelihood function plus an additional term whose role is to penalize large value of M. Density models classifiers have been recently applied to odor discrimination. A comprehensive comparison between various density models including GMM, GTM, PPCA can be found in [85, 104], where it is shown that GMM outperforms all density models for gas identification applications. In [111], it is also shown that GMM is very competitive when compared to other pattern recognition techniques. In addition, the main advantage of GMM is related to its class based training. In other words, adding new gases (new class) to the database is done without retraining the whole system, which is not the case for other classifiers. For KNN, the complexity and memory requirement are proportional to the number of data patterns. Adding new gases will increase its complexity and memory requirement rapidly.

4.3. Neural Networks

The posterior densities can be estimated via discriminant analysis, where the boundaries are directly learnt from data.

This supervised classification is based on the learning of the input pattern-class label mapping. The simplest discriminant functions are linear in the features. The multi-layer perceptron and radial basis function are the two most popular types of neural network architectures used for electronic nose applications [112].

4.3.1. Generalized Linear Models

Generalized linear models (GLM) are single layer networks. These models consist of a linear combination of the input patterns, in which the weights are the parameters of the model, followed by an appropriate activation function. The network has a single output variable y_k for each class

$$y_k = \boldsymbol{w}_k^T \boldsymbol{x} + \boldsymbol{w}_{k0} \tag{8}$$

The posterior probability $\wp(C_k|\boldsymbol{x})$ is estimated by $\sigma(y_k(\boldsymbol{x}))$, where $\sigma(y_k)$ is the activation function. Usually we use the logistic function for the two-class problem and the softmax activation function for the multiclass classification. For classification problems it is often advantageous to optimize the network outputs to represent the posterior probabilities of each class [105]. Hence, by computing the likelihood of the data we obtain the entropy error function:

$$E = -\sum_{i=1}^{N}\sum_{k=1}^{c} t_k^i \log(y_k^i)$$

where $\boldsymbol{t}_k (k = 1, \ldots, c)$ is the target vectors with 1-of-c coding. Once the error function and its gradients have been computed, GLM can be trained with nonlinear optimization algorithms. However, it is also possible to take advantage of the linear structure of the network and use a particularly efficient special purpose training algorithm known as iterated re-weighted least squares (IRLS) [113].

4.3.2. Multi-Layer Perceptron

The multi-layer perceptron (MLP) is probably the most widely used architecture for practical applications of neural networks. In most cases the MLP network consists of two layers of adaptive weights with full connectivity between inputs and hidden units, and between hidden units and outputs. An MLP is able to learn arbitrarily complex non linear regressions by adjusting the weights in the network using an appropriate optimization algorithm [114]. One way to generalize the linear discriminant function, so as to permit a much larger range of possible decision boundaries, is to transform the input vector $\boldsymbol{x}$ using a set of predefined nonlinear basis functions $\Phi_j(\boldsymbol{x})$ and represent the output as a linear combination of these functions:

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{M} w_{kj}\Phi_j(\boldsymbol{x}) + w_{k0} \tag{9}$$

The basis functions can be given by tanh activation functions:

$$\Phi_j(\boldsymbol{x}) = \tanh(\boldsymbol{\theta}_j^T \boldsymbol{x} + \boldsymbol{\theta}_{j0}) \tag{10}$$

MLP is trained to minimize the entropy cost function [115] with 1-of-c coding. The parameters are learnt from data using generally a gradient descent technique known as a back-propagation of errors [114].

Network complexity must be controlled by determining the optimal number (in terms of small generalization error) of hidden units M. The goal is to find a network which gives the best prediction on a new data set. The simplest approach is to train several MLPs using the same training data set. The performance of the networks is then compared by evaluating the error function using an independent validation set. Network complexity can also be controlled by using regularization approaches such as weight decay [100]. The model can be regularized to prevent any weights becoming too large by adding to the error function a weight decay penalty term [116]:

$$E_r = E + \alpha \sum_i w_i^2$$

where the hyperparameter α is non-negative. Additional details on complexity control can be found in [117].

4.3.3. Radial Basis Function

The radial basis function (RBF) network is the main practical alternative to the multi-layer perceptron for non linear modeling. Although the structure of RBF networks resembles that of MLPs, their input-output mappings and training algorithms are fundamentally different. In an RBF network, the activation of the hidden units Φ is given by a non-linear function of the distance between the input vector and a center vector.

For a large class of basis functions such as Gaussian one

$$\Phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|^2}{\sigma_j^2}\right) \tag{11}$$

RBF networks are universal approximators [118, 119] . The centers $\boldsymbol{c}_j$ and the width σ_j^2 defined the parameters for each hidden unit. Thus, the mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ is given by:

$$\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{\Phi}(\boldsymbol{x}) \tag{12}$$

where $\boldsymbol{\Phi}(\boldsymbol{x})$ are M fixed basis functions $\Phi_j(\boldsymbol{x})$ and $\boldsymbol{W}$ is a $c \times M$ matrix of the adjustable network weights.

One attraction of RBF networks is that there is a two-stage training procedure which is considerably faster than the methods used to train MLPs. In the first stage, the parameters of the basis functions, are set by modeling the unconditional data density. In fact, the sum of the basis functions $\sum \Phi_j$ represents the unconditional density of the input data. Therefore, we can use an unsupervised learning procedure to estimate the basis functions parameters. The second stage of training determines the weights in the output layer using supervised method [100].

Pattern recognition techniques based on artificial neural networks (ANN) approaches were very widely used for gas sensors applications [12, 13, 36, 41, 43–45, 53, 54, 58, 60, 69, 70]. In most of the these application the ANN is combined with the previously reviewed pre-processing feature reduction techniques such as PCA [55], DFA [66]. In [40] the authors discuss the influence of flow rate of the gases for binary mixture quantification using ANN. The performance of ANN as compared to other techniques greatly depend on the dataset and it is rather difficult to draw a conclusion with that respect. Details about the comparison of ANN and other techniques can be found in [111] (comparison with GMM and KNN) and [56] (comparison with fuzzy version of neural network). It should be however noticed that the MLP classifier has most probably the lowest complexity and the smallest memory requirements when compared to KNN (very large storage requirement) as well as density models based classifiers. The complexity of MLP classifier when adding new gases may not increase dramatically but the topology of the network has to be modified and retraining in order to obtain the new set of parameters. For GMM, the structure and parameters of the original system will not change when adding new gases.

RBF classifiers combined with the previously reviewed pre-processing techniques were also used for odor discrimination [25, 30].

4.4. Support Vector Machines

A challenging problem in machine learning is how to use a restricted amount of training data for constructing classifiers that generalize well in high-dimensional spaces. Such a classifier in the linearly separable case is the so-called *optimal hyperplane* that provides the largest distance or *margin* from the separating hyperplane to the closest training vector (see Fig. 5). By maximizing the margin of a linear discriminant we are minimizing bounds on the generalization error and thus can expect better generalization with high probability [120]. The basic idea behind the Support Vector Machines (SVMs) is first to transform the original data onto a high-dimensional space by using some fixed a priori mapping so that the training set becomes linearly separable, and then find the optimal hyperplane in this feature space [121, 122]. The mapping onto the feature space is obtained using a given kernel representation of the dot product [123]. In this case, we just have to evaluate kernels in the input space instead of dot products in the feature space. By changing kernels we can obtain different nonlinear classifiers in the input space. Compared to neural networks where the results are dependent on the network architecture and the initialization of the learning algorithm, the optimization in SVMs is formulated as a quadratic programming problem with no local minima. Moreover, the architecture of the model (number of support vectors) is automatically selected during the optimization process. SVMs have been originally developed for two class discrimination but extensions of the method now exist for multi-class problems [124]. SVMs are now used in E-nose applications [125–128] because they are well-grounded in statistical learning theory and they overcome many of the drawbacks seen in previously described pattern recognition techniques. In particular, they can generalize well from a restricted amount of training data. This is particularly interesting in the gas sensor area where it is very costly and time-consuming to obtain a large reliable and representative set of examples.

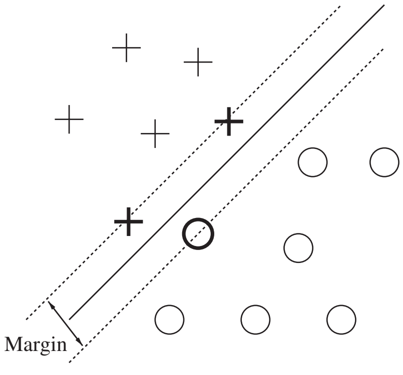

Figure 5. Optimal hyperplane that maximizes the margin in the case of linearly separable classes. The optimal hyperplane is defined by the support vectors indicated in bold.

4.5. Ensemble Methods

Combining multiple classifiers (such as neural networks or decision trees) to build an ensemble is an advanced pattern recognition technique, which has gained increasing attention within the machine learning community. Ensembles are well established as a method for obtaining highly accurate classifiers by combining different algorithms. Many methods for constructing ensembles have been developed. Here we will review bagging, boosting and a committee machine.

4.5.1. Bagging and Boosting

Bagging and Boosting are two popular methods proposed in order to create accurate ensembles (see compilation of papers at http://www.boosting.org). The two methods rely on re-sampling techniques to obtain different training sets for each of the individual classifiers. The resulting combined classifier is generally more robust and accurate than a single classifier trained on the original dataset.

Bagging [129] is a popular and effective technique for improving classification performance by creating ensembles. Bagging uses random sampling with replacement from the original data set in order to obtain different training sets. Because the size of the sampled data set has the same size as the original one, many of the original examples may be repeated while others may be left out. On average, 63% of the original data appears in the sampled training set [129]. Each individual classifier is built on each training set by applying the same learning algorithm. The resulting classifiers are then combined by a simple majority vote. It is well known that bagging significantly improves classifiers that are unstable in the sense that small perturbations in the training data may result in large changes in the generated classifier [129]. Empirical evaluations have shown that bagging improves decision trees or neural networks [130, 131] but does not improve the K-nearest neighbor algorithm [129]. For the nearest neighbor algorithm, a test case may change classification only if its nearest neighbor in the original dataset is not picked in at least half of the sampled training sets. The probability that this occurs gets very small as the number of classifiers within the ensemble gets larger [129]. A similar reasoning can be applied to support vector machines [132, 120] that are stable classifiers. Whether bagging decision trees are more accurate than bagging neural networks depends on the particular data set but on average they have similar performance and there is no clear evidence to prefer one or the other [130, 131]. However, because decision trees are fast to build with standard procedures (like CART [133], C4.5 [134] or OC1 [135]) and can be interpreted as a series of rules, they are often used in bagging ensembles.

Boosting method has shown success in a great variety of data domains [131, 136]. Boosting generates new classifier ensembles by readjusting the weight attached to each instance in a way that new ensemble classifiers will focus on difficult cases. The training set for each ensemble depends on the performance of previous classifier(s). After creating classifier replicates, boosting aggregates the votes from multiple classifiers. In bagging, the vote of each ensemble classifier carries the same weight, while the vote in boosting is weighted by the relative accuracy performance of each classifier. In [137], A new algorithm AdaBoost, has been introduced from the adaptive boosting. The basic idea of this algorithm is to construct a stronger classifier as a weighted sum of successively trained single classifiers. There are two ways in which the weights can be used to train a new classifier, boosting by sampling and boosting by weighting.

4.5.2. Committee Machine

Ensemble machine is a novel approach for assembling the outputs of various classification algorithms to obtain a unified decision with enhanced accuracy. A number of researchers have applied ensemble methods to improve the performance of neural networks [138]. The basic idea of a committee machine is to combine a mixture of experts and to effectively make use of the results produced by each expert within the ensemble. By combining the result of each classifier, we can arrive at a final result with improved performance. Each classifier gives its result R and the confidence C_f for the result to the combiner. We can introduce the use of confidence as a weighted vote for the combiner to avoid affecting the final decision by the result of individual expert featuring low confidence. In order to find confidences of various algorithms, we can adopt different approaches.

- K nearest neighbor: To classify a pattern $\boldsymbol{x}$, we find the closest K examples in the dataset and select the predominant class $C_{\hat{k}}$ among those K neighbors. Thus, the confidence of the result is defined as the number of data point of the result class divided by K, i.e.,

$$C_f = \frac{K_{\hat{k}}}{K}. \tag{13}$$

- Density models: We compute $\wp(C_k|\boldsymbol{x})$ by modeling the class conditional density $\wp(\boldsymbol{x}|C_k)$ (i.e., we train a model for each class) and then applying Bayes' rule to compute the posterior distribution:

$$\wp(C_k|\boldsymbol{x}) = \frac{\wp(\boldsymbol{x}|C_k)\wp(C_k)}{\sum_{l=1}^{c}\wp(\boldsymbol{x}|C_l)\wp(C_l)}. \tag{14}$$

  The class $C_{\hat{k}}$ with the maximum posterior probability is chosen as the result and this highest value is considered as the confidence for our density models.

- Discriminant functions: We choose a binary vector of size c for the classifier output. The target class is set to one and the others are set to zero. The class $C_{\hat{k}}$ with output value closest to 1 is chosen as the result and the output value is chosen as the confidence.

In order to reduce the risk of any algorithm that performs poorly on average from affecting the ensemble decision, the average accuracy is used as the weight W in the voting combiner. After collecting the results R, confidences C_f and the average accuracies W from the five algorithms, the combiner assembles the results by calculating the score S of each class as follows:

$$S_k = \sum_i W(i)C_{f_k}(i)R_k(i) \tag{15}$$

The class with the highest score would be selected as the recognized class of the committee machine.

Despite the fact that ensembles are very advanced pattern recognition techniques which have gained increasing attention within the machine learning community, their application to odor discrimination is seldom. In [139] bagging decision trees were used for E-Nose applications and their VLSI implementation was reported using 3D chip technology. In [140, 141] heterogenous classifiers including density models, KNN, ANN and SVM were reported for odor discrimination. In [142], the authors have showed how ensemble learning methods could be efficiently used in an array of chemical sensors.

5. Biologically Inspired Processing

Similar to the non-selectivity of gas sensors, a biological olfactory receptor neuron (ORN) is not tuned to a specific odor [143]. Because the variety of receptor proteins underlying chemoreception is not as rich as the repertoire of existing odorant molecules, different molecules may react with a particular protein and thus non-selectivity is probably an unavoidable situation an olfactory system has to face. This problem is still a challenge for electronic noses, as no really satisfactory general solution for improving both the sensitivity and the selectivity has been proposed so far. As many animals prove it by their impressive olfactory abilities, the olfactory pathway is probably fundamental in its design for facing this problem. The organization of the olfactory system is similar for insects and vertebrates [144], as depicted in Fig. 6. Understanding how the biological olfactory system works could then be highly beneficial for designing efficient electronic noses. As more and more underlying computational principles are understood by biologists, different processing stages in the olfactory pathway are modeled and applied to real gas sensor data.

5.1. Chemotopic Convergence for Improved Sensitivity

R. Axel and L. Buck, winners of the 2004 nobel prize in Medecine, have discovered a large family of genes leading to an equivalent number of olfactory receptor types [145]. As depicted in Fig. 6, sensory neurons that express the same odorant receptor gene converge onto very few glomeruli (e.g., 8000 ORNs for each glomerulus in the rabbit). By integrating signals from multiple ORNs of the same type, the olfactory system extends its dynamic range to lower concentrations leading to an improved sensitivity. Such a massive chemotopic convergence was exploited in [16], where it has

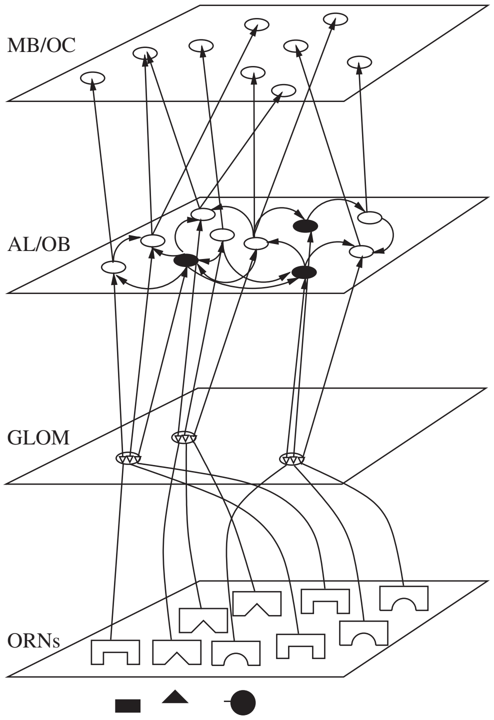

Figure 6. Schematic diagram of the olfactory system. In insects or mammals, odorant molecules are captured by olfactory receptor neurons (ORNs). A large number of ORNs that express the same odorant receptor gene converge onto glomeruli. The odorant identity is then represented in the glomerulus layer (GLOM) as a spatially distributed pattern and is further processed by higher brain structures: the olfactory bulb (OB) and olfactory cortex (OC) for vertebrates; the antennal lobe (AL) and mushroom body (MB) for insects. The AL/OB is a network of interconnected inhibitory local neurons (black circles) and excitatory projection neurons (white circles).

been shown that a large population of optical micro-bead chemical sensors can provide sensory hyperacuity by averaging out uncorrelated noise. However, building large sensor arrays still remains a challenge for most of the gas sensing technologies and existing electronic noses commonly use very few gas sensors, typically a dozen. As mentioned in Section 2.2, temperature modulation of a single sensor is an effective technique to create virtual sensors [81] and thus to bridge the gap between electronic noses and their biological counterparts. In [17, 14], a chemotopic map has been obtained by running the self-organizing map algorithm [146] on temperature modulated sensor data.

5.2. Phase Coding for Concentration Invariant Odor Recognition

Oscillations are very often encountered in experimentally recorded signals in the olfactory system. In mammals, the theta oscillation coupled to the respiration cycle seems to play an important role in the coding of sensory information by providing a "clock" or temporal frame of reference for the encoding neurons. In the mouse olfactory bulb, the phase of the firing of a mitral cell relative to the beginning of each oscillation cycle reflects input intensity [147]. The approximate logarithmic relationship makes the phase pattern invariant to the different concentrations of the same odor, as predicted by a theoretical study almost a decade ago [148]. When the odor concentration changes, the entire phase pattern is shifted but the relative phases are preserved. Reading the relative phase pattern can be accomplished in a biological plausible way with delay lines and synchronization detector neurons [148, 149]. These ideas have been implemented and successfully tested for invariant odor recognition in electronic noses [15, 18].

5.3. Temporal Decorrelation for Improved Selectivity

In the presence of an olfactory stimulus, subsets of neurons either in the olfactory bulb of vertebrates or the antennal lobe of insects fire in synchrony. The identities of these synchronized neurons evolve in time in an odor-specific manner. Each odor is thus represented by a specific sequence of transiently synchronized neurons defining a spatio-temporal code [150–152]. Although the spatial aspect of this code gives a synchronous population code which seems a priori sufficient to encode an odor, the role of the temporal aspect is to decorrelate the representations of similar odorants over time [153]. Recent computational modeling have shown that feedback inhibition and firing rate adaptation are responsible for achieving synchronization and setting its temporal evolution [154, 155]. Such a temporal decorrelation technique was exploited in a biologically plausible olfactory bulb model and successfully applied to classify data from a MOS sensor array [14, 17].

6. ODOR DISCRIMINATION SYSTEM—A CASE STUDY

We propose to study the performance of a committee machine for the identification of combustion gases. We combine five algorithms (MLP, KNN, GMM, SVM, PPCA) to form a Gas Identification Committee Machine (GICM) as shown in Fig. 7.

6.1. Experimental Setup

Measurements have been done using the experimental setup shown on Fig. 8 which consists of a special sensor chamber equipped with gas pumps and mass flow controllers as well as a data acquisition board.

The sensor array composed of 8 micro-hotplate based SnO_2 thin film gas sensors, have been used [156]. Four sensors with Pt/SnO_2 sensing film, two with Au/SnO_2 sensing

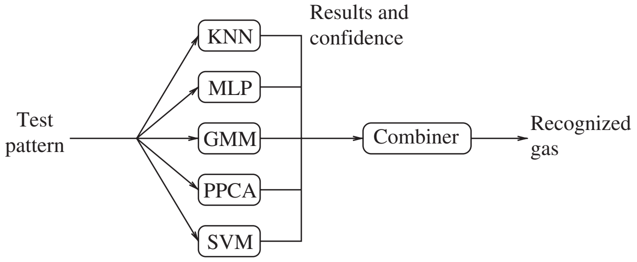

Figure 7. Architecture of the GICM system.

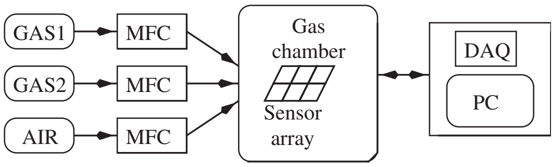

Figure 8. Experimental setup for gas discrimination. MFC stands for Mass Flow Controller.

film and the other two with Pt/Cu(0.16 wt%)-SnO_2 [157]. The sensors' operating temperature was chosen to be 300 °C for the purpose of good sensitivity to the studied gases. The sensors outputs are raw voltage measurements in the form of exponential-like curves.

Gases used in the experiment are methane, carbon monoxide, hydrogen, and two binary mixtures: one of methane and carbon monoxide and another of hydrogen and carbon monoxide. Vapors were injected into the gas chamber at a flow rate determined by the mass flow controllers (MFC). Concentration ranges are reported in Table 2.

The steady state values of the array sensor were recorded while periodically injecting different gases. A gas data set of 220 patterns was created to evaluate the performance of the proposed gas identification committee machine. We subtracted the baseline of each sensor in order to reduce the effects of the additive sensor drift. Since our goal is the qualitative classification of patterns, a normalization procedure is used in order to reduce the influence of concentrations and nonlinearities. Each input pattern is divided by its Euclidean norm.

6.2. Classification Results

Since the dataset we used was small, generalization performances were estimated by using the 10-fold cross validation approach. Density models are built by considering each of the class in turn, and estimating the corresponding class-conditional densities $\wp(\boldsymbol{x}|C_k)$ from the data. For discriminant functions a softmax output activation function is used to ensure that the outputs lay in the range $[0, 1]$ and summed to one. For SVMs, the multiclass implementation of [124] was used. the error penalty parameter C was set to 10 and a gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-(\|\boldsymbol{x} - \boldsymbol{x}_i\|^2)/\sigma^2)$ was used with $\sigma^2 = 0.001$ times the input dimensionality. A comparative study of different dimensionality reduction techniques has been conducted in [104]. The best results were found using the PCA projection. Therefore, the inputs to each classifier are the projections of the data using PCA. The selected classifiers are built on projected datasets with different numbers of principal components. The parameters of each mixture models were adapted to the training data in the maximum likelihood framework using EM algorithm. All parameters (for example, the number of hidden units) were chosen according to the validation stage. Figure 9 gives the classification performance of the trained classifiers for different numbers of principal components. This figure shows that the most accurate discriminant function is the SVM, while the most accurate density model is the GMM. For eight principal components, the best performance is obtained for SVMs. This is not surprising because SVMs work quite well when the dataset is small with respect to the input dimensionality. However, better performance is obtained for GMM when projecting the data to the five first principal components. This result points out the importance of feature reduction in improving the classifier performance by increasing the ratio of the number of training samples over the number of features.

Table 2. Gases and their concentration ranges.

Gas	Concentration range (ppm)
CO	25–200
CH_4	500–4000
CO & CH_4	25–200 & 500–4000
H_2	500–2000
CO & H_2	25–200 & 500–2000

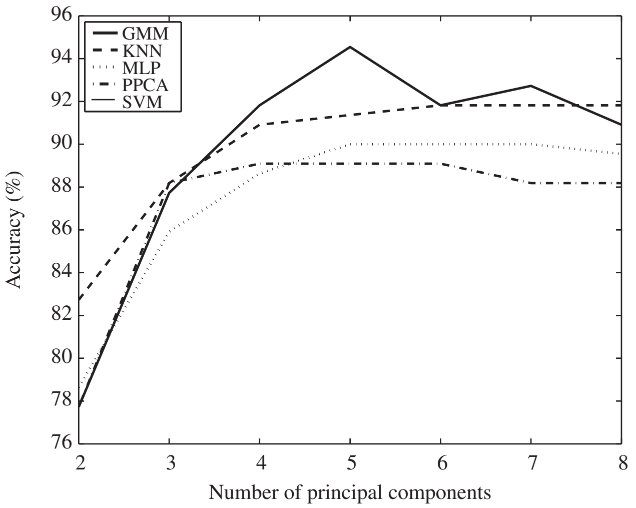

Figure 9. Accuracy as a function of the number of principal components.

To evaluate the performance of the proposed committee machine, we collected the results, confidences and the average accuracies from the five classifiers to determine the score for each class. Table 3 reports the accuracy of the trained classifiers for different sets. It is clear from this table that GICM gives superior performances. This is clearly

Table 3. Gas identification results (%). The most accurate discriminant function is SVM, while the most accurate density model is GMM. For eight principal components, the best performance is obtained for SVMs.

Set	MLP	KNN	GMM	SVM	PPCA	GICM
1	95.4	95.4	86.4	86.4	86.4	**100**
2	**100**	**100**	95.4	**100**	95.4	**100**
3	90.9	86.4	95.4	95.4	90.9	95.5
4	90.9	95.4	**100**	**100**	95.4	**100**
5	90.9	90.9	95.4	86.4	90.9	95.5
6	86.4	86.4	90.9	86.4	77.3	90.9
7	90.9	95.4	95.4	90.9	90.9	**100**
8	86.4	81.8	90.9	86.4	90.9	90.9
9	86.4	90.9	**100**	90.9	86.4	95.5
10	81.8	90.9	95.4	95.4	86.4	95.4
Average	90	91.4	**94.5**	91.8	89	**96.4**

shown in set 1 and 7, where none of the classifiers has 100% accuracy but our committee machine achieves it. The result also demonstrates that with the use of confidence and weight function, poor result from individual classifiers would not affect the ensemble result significantly. This robustness property can be seen for example in set 8 or 10, where the poor results of the KNN or MLP classifier does not affect the ensemble accuracy.

7. SUMMARY

In the past decade, the demand for odor discrimination and gas detection has emerged and the number of applications involving systems that can detect and discriminate odors (E-Nose) is sharply increasing. Gas sensors currently available suffer from a wide number of problems including non-selectivity, non-linearities, drift and noise, which makes further processing an inevitable task. Therefore, preprocessing the sensors signals and using advanced pattern recognition techniques are fundamental parts of an electronic nose. However, training a classifier for detecting and recognizing different odors still remains challenging partly because of the temporal variability of the gas sensors, the large intra-class variance as compared to the small inter-class separation and the small amount of training data available. Here we have reviewed a range of classical and advanced pre-processing and pattern recognition techniques for odor discrimination. Table 4 summarizes again the techniques discussed. A large number of pre-processing and classification techniques have been used in the past decade. The pre-processing technique can be categorized in two main families namely feature extraction and feature reduction, while the processing techniques can be divided into two main families namely pattern recognition classification and biologically inspired processing. The latter is gaining more and more attention as recent breakthroughs and answers are being provided by biologist with respect to the functioning of biological olfaction. Pre-processing techniques including feature extraction and reduction are very important means to improve the classification performance in gas detection. Most of the pre-processing techniques encountered in the literature are based on various sampling procedures (steady state, transient, dynamic slope, temporal windowing, temperature modulation, etc.), as well as basic manipulation of the acquired sensor information such as baseline manipulation. Even though some of the proposed techniques require extensive manipulation of the sensor' data, they do offer improved classification performance when combined with the right post-processing techniques. A large number of classification techniques have been applied for odor discrimination and there is no clear answer to which pattern recognition technique is preferable as the performance is highly dependant on the problem at hand. The use of dimensionality reduction depends on the relationship between the training dataset size and number of features. If the number of training examples is very large, then the classification error does not increase as the number of features increases. However, if the number of training examples is small relative to the number of features, a dimensionality reduction technique is often needed to guarantee an acceptable classification accuracy. This is mainly true for density models classifiers as they are based on statistical approaches. Since only a limited number of examples are typically available for e-nose applications, there is an optimal number of feature dimensions beyond which the classifier performance starts to degrade. It has been shown on a case study that the most accurate discriminant function is the SVM, while the most accurate density model is the GMM. For eight principal components, the best performance is obtained for SVMs. SVMs are shown to work quite well when the dataset is small with respect to the input dimensionality. However, better performance is obtained for GMM when projecting the data to the five first principal components. This points out to an important result which suggests that higher generalization performance can be obtained by using feature reduction and selection techniques as preprocessing for increasing the ratio of the number of training samples over the number of features. Moreover, a more accurate and robust classification can be obtained by combining different classifiers within an ensemble structure.

Table 4. Summary of some of the pre-processing and pattern recognition algorithms used in electronic nose and odor discrimination applications. SS, DR and DS stand for Steady state, Dynamic response and Dynamic Slope, respectively.

Processing description	References
Feature extraction	
• Baseline manipulation	[71, 72]
• Data normalization	[71]
• SS, DR and DS extraction	[35, 37, 48, 59, 68, 71–77, 79, 158]
• Temporal windowing	[72, 78]
• Temperature modulated	[72, 17, 80, 81]
Feature reduction	
• PCA	[33, 35, 37, 51, 55, 67, 83, 84, 103]
• LDA	[61, 85, 103]
• Neuroscale	[85, 103]
• BSS	[86–90]
Classification	
• KNN	[27, 101]
• DM	[85, 104, 111]
• ANN	[12, 13, 36, 41, 43–45, 53, 54, 56, 58, 60, 69, 70]
• RBF	[25, 30]
• SVM	[125–128]
• Ensembles	[139–142]
Biologically inspired	
• Chemotopic convergence	[14, 16, 17]
• Phase coding	[15, 18]
• Temporal decorrelation	[14, 17]

LIST OF SYMBOLS

$\boldsymbol{x}$ original feature vector

$\boldsymbol{x}_i$ data point

z reduced dimensionality feature vector

z_i projected data point

$\boldsymbol{s}$ original source vector

$\boldsymbol{MM}$ Mixing Matrix of the BSS problem.

$\boldsymbol{T}$ transformation function for dim. reduction

$\boldsymbol{T}_{\mathbf{PCA}}$ PCA transformation matrix
$\boldsymbol{T}_{\mathbf{LDA}}$ LDA transformation matrix
$\boldsymbol{T}_{\mathbf{LDA}}$ LDA transformation matrix
$\boldsymbol{U}$ eigenvectors matrix of data Cov. Matx
$\boldsymbol{\Lambda}$ diagonal eigenvalues matx of data Cov. Matx
$\boldsymbol{W}$ the within-class scatter
$\boldsymbol{S}_w$ eigenvectors matrix of $\boldsymbol{W}$
$\boldsymbol{\Lambda}_w$ diagonal eigenvectors matrix of $\boldsymbol{W}$
$\boldsymbol{B}$ the between class scatter
$\boldsymbol{S}_B$ eigenvectors matrix of $\boldsymbol{B}$
$\boldsymbol{D}_{ij}^{*}$ Euclidean distance
$\boldsymbol{E}_s$ Sammon stress metric
$\hat{\boldsymbol{F}}$ feature subset
$\boldsymbol{J}(\boldsymbol{F})$ selection criteria for feature selection
$\wp(\boldsymbol{C}_k|\boldsymbol{x})$ posterior probability of class membership
$\wp(\boldsymbol{x}|\boldsymbol{C}_k)$ class-conditional density
$\wp(\boldsymbol{C}_k)$ prior probability
$\wp(\boldsymbol{j})$ the mixing coefficients of GMM
$\boldsymbol{l}$ log likelihood of the dataset $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$
$\hat{\boldsymbol{l}}_M$ maximized log-likelihood of the dataset
$\boldsymbol{N}_p(\boldsymbol{M})$ number of parameters in the M Gaussian model
$\hat{\boldsymbol{M}}$ optimized number of GMM components
$\boldsymbol{E}$ Entropy error function
$\boldsymbol{t}_k$ target vectors
$\boldsymbol{\Phi}(\boldsymbol{x})$ activation function of the hidden units
$\boldsymbol{C}_f$ confidence of individual classifier
$\mathbf{H_2}$ hydrogen
$\mathbf{CO}$ carbon monoxide
$\mathbf{NO_2}$ nitrogen dioxide
$\mathbf{NH_3}$ ammonia
$\mathbf{CH_4}$ methane
$\mathbf{CHCl_3}$ chloroform
$\mathbf{H_2S}$ hydrogen sulfide
$\mathbf{C_2H_5OH}$ ethyl alcohol
$\mathbf{C_4H_{10}}$ propane
$\mathbf{CH_3SH}$ methyl mercaptan
$\mathbf{(CH_3)_3N}$ trimethylamine
$\mathbf{SO_2}$ sulfur dioxide
$\mathbf{C_2H_6O}$ ethyl cellulose
$\mathbf{C_3H_8}$ Propane
$\mathbf{C_6H_6}$ benzene
$\mathbf{CF_3CH_2F}$ Tetrafluoroethane
$\mathbf{CH_4O}$ methyl alcohol
$\mathbf{NO}_x$ nitrogen oxides

LIST OF ABBREVIATIONS

ANN Artificial Neural Network
AS Analysis line of the sensor response
BS Baseline of the sensor response
BSS Blind Source Separation
CART Classification And Regression Trees
CPLD Complex Programmable Logic Device
DAQ Data Acquisition Board
DFA Discriminant Factorial Analysis
EM Expectation Maximization
GMM Gaussian Mixture Models
GTM Generative Topographic Mapping
GLM Generalized Linear Models
ICA Independent Component Analysis
IRLS Iterated Re-weighted Least Squares
KNN *K* Nearest Neighbor
LDA Linear Discriminant Analysis
MDL Minimum Description Length
MFC Mass Flow Controller
MLP Multi-Layer Perceptron
ORN Olfactory Receptor Neuron
PC Personal Computer
PCA Principal Component Analysis
PPCA Probabilistic PCA
RBF Radial Basis Function
SFS Sequential Forward Selection
SVM Support Vector Machine

ACKNOWLEDGMENTS

The work described in this paper was supported by a Competitive Earmarked Research Grant (reference No: HKUST 6162/04E) from the Research Grant Council of Hong Kong. Part of the work was also supported by a grant from the *PROCORE*-France/Hong Kong Joint Research Scheme sponsored by the Research Grant Council of Hong Kong and the Consulate General of France in Hong Kong (Reference No: F-HK15/03T). Thanks are due to Professor Philip Chan for providing us with the microelectronic gas sensors used for the experimental part of the chapter.

REFERENCES

1. D.-S. Lee, D.-D. Lee, S.-W. Ban, M. Lee, and Y. T. Kim, *IEEE Sensors Journal* 2(3), 140 (2002).

2. J. W. Gardner, H. W. Shin, E. L. Hines, and C. S. Dow, *Sensors and Actuators* 69, 336 (2000).

3. H. W. Shin, E. Llobet, J. W. Gardner, E. L. Hines, and C. S. Dow, *IEE Proceedings-Science Measurement and Technology* 147(4), 158 (2000).

4. H. T. Nagle, R. Gutierrez-Osuna, B. G. Kermani, and S. S. Schiffman, in"Handbook of Machine Olfaction: Electronic Nose Technology" (T. C. Pearce, S. S. Schiffman, H. T. Nagle, and J. W. Gardner, Eds.). Wiley-VCH, 2002.

5. S. Capone, A. Forleo, R. Rella, P. Siciliano, J. Spadavecchia, D. S. Presicce, and A. M. Taurino, *Journal of Optoelectronics and Advanced Materials* 5(5), 1335 (2003).

6. E. Llobet, E. L. Hines, J. W. Gardner, and S. Franco, *Meas. Sci. Technol.* 10, 538 (1999).

7. E. L. Hines, E. Llobet, and J. W. Gardner, *Electronics Letters* 35, 821 (1999).

8. J. J. Classen, S. S. Schiffman, H. T. Nagle, and R. Gutierrez-Osuna, in "International Symposium on Ammonia and odour emissions from animal production facilities" (J. A. M. Voermans and G. J. Monteny, Eds.). Vinkeloord, The Netherlands, NVTL: Rosmalen, The Netherlands, 1997.

9. J. W. Gardner, M. Craven, C. Dow, and E. L. Hines, *Meas. Sci. Technol.* 9, 120 (1998).

10. J. W. Gardner, H. W. Shin, and E. L. Hines, *Sensors and Actuators B* 70, 19 (2000).

11. S. S. Schiffman, D. W. Wyrick, R. Gutierrez-Osuna, and H. T. Nagle, in "The 7th International Symposium On Olfaction and Electronic Nose," Brighton, England, 2000.

12. L. Marques, U. Nunes, and A. de Almeida, *Thin Solid Films* 418(1), 51 (2002).

13. K. C. Persaud, A. M. Pisanelli, S. Szyszko, M. Reichl, G. Horner, W. Rakow, H. J. Keding, and H. Wessels, *Sensors and Actuators B* 55(2–3), 118 (1999).

14. B. Raman, A. Gutierrez-Galvez, A. Perera-Lluna, and R. Gutierrez-Osuna, in "Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems," Vol. 1. p. 319. Sendai, Japan, 2004.

15. J. White, T. A. Dickinson, D. R. Walt, and J. S. Kauer, *Biol. Cybern.* 78, 245 (1998).

16. T. C. Pearce, P. F. M. J. Verschure, J. White, and J. S. Kauer, "Emergent Neural Computation Architectures Based on Neuroscience," p. 461. Springer-Verlag, 2001.

17. B. Raman and R. Gutierrez-Osuna, "Neural Information Processing Systems (NIPS)." Vancouver, BC, 2004.

18. O. Rochel, D. Martinez, E. Hugues, and F. Sarry, in "Eurosensor Conf.," Prague, 2000.

19. K. J. Albert, N. S. Lewis, C. L. Schauer, G. A. Sotzing, S. E. Stitzel, T. P. Vaid, and D. R. Walt, *Chem. Rev.* 100, 2595 (2000).

20. J. W. Gardner and P. N. Bartlett, *Sensors and Actuators B* 18–19, 211 (1994).

21. J. W. Gardner and P. N. Bartlett, "Electronic Noses, Principles and Applications." Oxford University Press, 1999.

22. J. W. Gong, Q. F. Chen, W. F. Fei, and S. Seal, *Sensors and Actuators B* 102(1), 117 (2004).

23. C. Delpha, M. Lumbreras, and M. Siadat, *Sensors and Actuators B* 98(1), 46 (2004).

24. R. E. Cavicchi, G. E. Poirier, N. H. Tea, M. Afridi, D. Berning, A. Hefner, J. Suehle, M. Gaitan, S. Semancik, and C. Montgomery, *Sensors and Actuators B* 97(1), 22 (2004).

25. R. Ionescu, E. Llobet, J. Brezmes, X. Vilanova, and X. Correig, *Sensors and Actuators B* 95(1–3), 177 (2003).

26. M. A. Aronova, K. S. Chang, I. Takeuchi, H. Jabs, D. Westerheim, A. Gonzalez-Martin, J. Kim, and B. Lewis, *Applied Physics Letters* 83(6), 1255 (2003).

27. R. Gutierrez-Osuna, A. Gutierrez-Galvez, and N. Powar, *Sensors and Actuators B* 93(1–3), 57 (2003).

28. K. D. Mitzner, J. Sternhagen, and D. W. Galipeau, *Sensors and Actuators B* 93(1–3), 92 (2003).

29. J. Wollenstein, J. A. Plaza, C. Cane, Y. Min, H. Bottner, and H. L. Tuller, *Sensors and Actuators B* 93(1–3), 350 (2003).

30. R. Ionescu, E. Llobet, X. Vilanova, J. Brezmes, J. E. Sueiras, J. Calderer, and X. Correig, *Analyst* 127(9), 1237 (2002).

31. Y. Sakurai, H. S. Jung, T. Shimanouchi, T. Inoguchi, S. Morita, R. Kuboi, and K. Natsukawa, *Sensors and Actuators B* 83(1–3), 270 (2002).

32. A. Jerger, H. Kohler, F. Becker, H. B. Keller, and R. Seifert, *Sensors and Actuators B* 81(2–3), 301 (2002).

33. C. Delpha, M. Siadat, and M. Lumbreras, *IEEE Trans. on Instrumentation and Measurement* 50(5), 1370 (2001).

34. A. Ortega, S. Marco, A. Perera, T. Sundic, A. Pardo, and J. Samitier, *Sensors and Actuators B* 78(1–3), 32 (2001).

35. C. Delpha, M. Siadat, and M. Lumbreras, *Sensors and Actuators B* 78(1–3), 49 (2001).

36. B. S. Joo, N. J. Choi, Y. S. Lee, J. W. Lim, B. H. Kang, and D. D. Lee, *Sensors and Actuators B* 77(1–2), 209 (2001).

37. F. Sarry and M. Lumbreras, *IEEE Trans. on Instrumentation and Measurement* 49(4), 809 (2000).

38. T. Roppel and D. M. Wilson, *Pattern Recognition Letters* 21(3), 213 (2000).

39. C. Delpha, M. Siadat, and M. Lumbreras, *Sensors and Actuators B* 59(2–3), 255 (1999).

40. T. Eklov and I. Lundstrom, *Sensors and Actuators B* 57(1–3), 274 (1999).

41. M. A. Martin, J. P. Santos, H. Vasquez, and J. A. Agapito, *Sensors and Actuators B* 58(1–3), 469 (1999).

42. K. K. Shukla, R. R. Das, and R. Dwivedi, *Sensors and Actuators B* 50(3), 194 (1998).

43. S. Marco, A. Ortega, A. Pardo, and J. Samitier, *IEEE Trans. on Instrumentation and Measurement* 47(1), 316 (1998).

44. E. Llobet, J. Brezmes, X. Vilanova, J. E. Sueiras, and X. Correig, *Sensors and Actuators B* 41(1–3), 13 (1997).

45. H. K. Hong, H. W. Shin, D. H. Yun, S. R. Kim, C. H. Kwon, K. Lee, and T. Moriizumi, *Sensors and Actuators B* 36(1–3), 338 (1996).

46. V. K. Singh, R. Dwivedi, and S. K. Srivastava, *Microelectronics Journal* 27(6), 531 (1996).

47. H. Sundgren, F. Winquist, I. Lukkari, and I. Lundstrom, *Measurement Science and Technology* 2(5), 464 (1991).

48. E. Llobet, J. Brezmes, X. Vilanova, L. Fondevila, and X. Correig, in "International Conference on Solid State Sensors and Actuators," Vol. 2, p. 971. Transducers '97, Chicago, 1997.

49. M. Kermit and O. Tomic, *IEEE Sensors Journal* 3(2), 128 (2003).

50. R. Menzel and J. Goschnick, *Sensors and Actuators B* 68(1–3), 115 (2000).

51. A. C. Romain, J. Nicolas, V. Wiertz, J. Maternova, and Ph. Andre, *Sensors and Actuators B* 62(1), 73 (2000).

52. P. Althainz, J. Goschnick, S. Ehrmann, and H. J. Ache, *Sensors and Actuators B* 33(1–3), 72 (1996).

53. J. Getino, L. Ares, J. I. Robla, M. C. Horrillo, I. Sayago, M. J. Fernandez, J. Rodrigo, and J. Gutierrez, *Sensors and Actuators B* 59(2–3), 249 (1999).

54. E. L. Kalman, F. Winquist, and I. Lundstrom, *Atmospheric Environment* 31(11), 1715 (1997).

55. X. B. Zou, J. W. Zhao, S. Y. Wu, and X. Huang, *Sensors* 3(4), 101 (2003).

56. R. R. Das, K. K. Shukla, R. Dwivedi, and A. R. Srivastava, *Microelectronics Journal* 30(8), 793 (1999).

57. C. Nicolas, A. S. Barros, D. N. Rutledge, J. Hossenlopp, G. Trystram, and C. Emonet, *Analysis* 26(3), 135 (1998).

58. A. Jonsson, F. Winquist, J. Schnurer, H. Sundgren, and I. Lundstrom, *International Journal of Food Microbiology* 35(2), 187 (1997).

59. H. Nanto, S. Tsubakino, M. Ikeda, and F. Endo, *Sensors and Actuators B* 25(1–3), 794 (1995).

60. F. Winquist, E. G. Hornsten, H. Sundgren, and I. Lundstrom, *Measurement Science and Technology* 4(12), 1493 (1993).

61. T. Aishima, *Journal of Agricultural and Food Chemistry* 39(4), 752 (1991).

62. M. L. Rodriguez-Mendez, A. A. Arrieta, V. Parra, A. Bernal, A. Vegas, S. Villanueva, R. Gutierrez-Osuna, and J. A. de Saja, *Sensors Journal, IEEE* 4(3), 348 (2004).

63. P. Wide, F. Winquist, P. Bergsten, and E. M. Petriu, *Instrumentation and Measurement, IEEE Transactions* 47(5), 1072 (1998).

64. R. Singh, "Electronic Design, Test and Applications" in "The First IEEE International Workshop," p. 489, 2002.

65. T. Moriizumi and T. Nakamoto, *Power Electronics and Motion Control* 3, 1645 (1992).

66. P. Corcoran and P. Lowery, "Artificial Neural Networks" in "Fourth International Conference," p. 415, 1995.

67. M. Penza, G. Cassano, F. Tortorella, and G. Zaccaria, *Sensors and Actuators B* 73(1), 76 (2001).

68. P. Corcoran, P. Lowery, and J. Anglesea, *Sensors and Actuators B* 48(1–3), 448 (1998).

69. H. Liden, C. F. Mandenius, and L. Gorton, *Analytica Chimica Acta* 361(3), 223 (1998).
70. J. Brezmes, B. Ferreras, E. Llobet, X. Vilanova, and X. Correig, *Analytica Chimica Acta* 348(1–3), 503 (1997).
71. R. Gutierrez-Osuna and H. T. Nagle, *IEEE Trans. Syst. Man Cybern. B* 29, 626 (1999).
72. R. Gutierrez-Osuna, H. T. Nagle, B. Kermani, and S. S. Schiffman, in "Handbook of Machine Olfaction: Electronic Nose Technology" (T. C. Pearce, S. S. Schiffman, H. T. Nagle, and J. W. Gardner, Eds.). Wiley-VCH, Weinheim, Germany, 2002.
73. J. W. Gardner, M. Craven, C. Dow, and E. L. Hines, *Meas. Sci. Technol.* 9, 120 (1998).
74. T. Eklov, P. Martensson, and I. Lundstrom, *Anal. Chim. Acta* 353, 291 (1997).
75. R. Gutierrez-Osuna, H. T. Nagle, and S. Schiffman, *Sens. Actuators B* 61(1–3), 170 (1999).
76. T. Eklov, P. Martensson, and I. Lundstrom, *Anal. Chim. Acta* 381, 221 (1999).
77. C. Distante and P. Siciliano, *Circuits and Systems* 3, 1243 (2001).
78. C. Bordieu, D. Rebiere, J. Pistre, and R. Planade, *Sensors and Actuators B* 35(1–3), 52 (1996).
79. S. Brahim-Belhouari, A. Bermak, G. Wei, and P. C. H. Chan, in "IEEE TENCON," Vol. A, p. 693, Chiang Mai, Thailand, 2004.
80. R. Gutierrez-Osuna, S. Korah, A. Perera, in "The 8th International Symposium on Olfaction and the Electronic Nose," Washington, DC, 2001.
81. A. P. Lee and B. J. Reedy, *Sensors and Actuators B* 60, 35 (1999).
82. A. K. Jain, R. Duin, and J. Mao, *IEEE Trans. Pattern Anal. Machine Intell.* 22, 4 (2000).
83. C. M. McEntegart, W. R. Penrose, S. Strathmann, and J. R. Stetter, *Sensors and Actuators B* 70(1–3), 170 (2000).
84. M. Kermit and O. Tomic, *IEEE Sensors Journal* 3(2), 218 (2003).
85. S. Brahim-Belhouari, A. Bermak, G. Wei, and P. C. H. Chan, in "The IEEE International Symposium on Signal Processing and Information Technology," p. 138, ISSPIT'03, Germany, 2003.
86. G. Bedoya, S. Bermejo, and J. Cabestany, in "European Symposium on Artificial Neural Networks," p. 131. Bruges, Belgium, 2003.
87. G. Wei, Z. Tang, P. C. H. Chan, and J. Yu, in "Lectures Notes in Computer Science," p. 3173, 2004.
88. C. Di Natale, E. Martinelli, and A. D'Amico, *Sensors and Actuators B* 82(2–3), 158 (2002).
89. S. Bermejo and J. Solé-Casals, in "Third IEEE Sensor Array and Multichannel Signal Processing Workshop," Barcelona, Spain, 2004.
90. D. Martinez and A. Bray, *IEEE Trans. Neural Networks* 14(1), 228 (2003).
91. D. Lowe and M. E. Tipping, *Neural Computing and Applications* 4, 83 (1996).
92. B. Scholkopf, A. J. Smola, and K.-R. Muller, *Neural Computation* 10, 1299 (1998).
93. A. Hyvrinen and E. Oja, *Neural Comput. Surveys* 2, 94 (1999).
94. J. H. Friedman and J. Am, *Statistical Assos.* 82, 249 (1987).
95. P. M. Narendra and K. Fukunaga, *IEEE Trans. Comput.* 26, 917 (1977).
96. T. Eklov, P. Martensson, and I. Lundstrom, *Anal. Chim. Acta* 381, 221 (1999).
97. P. A. Devijver and J. Kittler, "Pattern Recognition, A Statistical Approach." Prentice-Hall, Englewood Cliffs, NJ, 1982.
98. P. Pudil, J. Novovicov, and J. Kittler, *Pattern Recognit. Lett.* 15, 1119 (1994).
99. A. Webb, "Statistical Pattern Recognition." London, Arnold, UK, 1999.
100. R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," 2nd Edition. Wiley, New York, 2000.
101. L. Chambon, J. P. Germain, A. Pauly, V. Demarne, and A. Grisel, *Sensors and Actuators B* 60(2–3), 138 (1999).
102. D. M. Titterington, A. F. M. Smith, and U. E. Makov, "Statistical Analysis of Finite Mixture Distributions." John Wiley, New York, 1985.
103. S. Brahim-Belhouari, A. Bermak, G. Wei, and P. C. H. Chan, in "IEEE Conf. on IMTC," p. 675, Italy, 2004.
104. S. Brahim-Belhouari and A. Bermak, "Pattern Recognition Letters." Elsevier Science Publisher, 2005.
105. C. M. Bishop, "Neural Networks for Pattern Recognition." Clarendon Press, Oxford, 1995.
106. Y. Zhang, M. Alder, and R. Togneri, in "ICASSP Conf.," Vol. 1, p. 613, 1994.
107. N. Vasconcelos and A. Lippman, in "IEEE ICME Conf.," Vol. 2, p. 899, 2000.
108. C. M. Bishop, M. Svensen, and C. K. I. William, *Neural Computation* 10(1), 215 (1996).
109. M. E. Tipping and C. M. Bishop, *J. Roy. Statist. Soc.* B(61), 611 (1999).
110. J. Rissanen, *Automatica* 14, 465 (1978).
111. S. Brahim-Belhouari, A. Bermak, P. C. H. Chan, in "IEEE International Conference on Acoustics Speech and Signal Processing," Vol. V, p. 833. Montreal, Canada, 2004.
112. R. Gutierrez-Osuna, *IEEE Sensors Journal* 2(3), 189 (2002).
113. I. T. Nabney and Netlab, "Algorithms for Pattern Recognition." Springer Press, London, 2003.
114. D. Rumelhart, G. Hinton, and R. Williams, in "Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations," Vol. 1, p. 318. MIT Press, Cambridge, MA, 1986.
115. J. B. Hampshire and B. A. Perlmutter, in "Connectionist Models: Proc. 1990 Summer School" (D. S. Touretzky, J. L. Elman, T. J. Sejnowski, and G. E. Hinton, Eds.), p. 159. San Mateo, CA, 1990.
116. G. E. Hinton, in "RARLE Proc.," p. 1. Springer-Verlag, Berlin, 1987.
117. M. Pardo and G. Sberveglieri, *IEEE Sensors Journal* 2(3), 189 (2002).
118. E. J. Hartman, D. Keeler, and Jacek M. Kowalski, *Neural Computation* 2(2), 210 (1990).
119. J. Park and I. W. Sandberg, *Neural Computation* 5(2), 305 (1993).
120. V. Vapnik, "Statistical Learning Theory." John Wiley & Sons, Inc, 1998.
121. B. Boser, I. Guyon, and V. Vapnik, in "Fifth Annual Workshop on Computational Learning Theory," p. 144. ACM, Pittsburgh, 1992.
122. C. Cortes and V. Vapnik, *Machine Learning* 20, 1 (1995).
123. M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer, *Automation and Remote Control* 25, 821 (1964).
124. Y. Guermeur, *Pattern Analysis and Applications* 5(2), 168 (2002).
125. S. Al-Khalifa, S. Maldonado-Bascon, and J. W. Gardner, *IEE Pro.-Sci. Technol. Journal* 150(1), (2003).
126. D. DeCoste and M. Burl, "Fourth Workshop on Mining Scientific Datasets (KDD-2001)," August 2001.
127. C. Distante, N. Ancona, and P. Siciliano, *Sensors and Actuators B* 88(1), 30 (2003).
128. J. Trihaas and H. H. Bothe, "ISOEN Symposium," Roma, Italy, 2002.
129. L. Breiman, *Machine Learning* 24(2), 123 (1996).
130. D. Opitz and R. Maclin, *Journal of Artificial Intelligence Research* 11, 169 (1999).
131. R. Maclin and D. Opitz, in "Proc. of AAAI," 1997.
132. C. J. C. Burges, *Data Mining and Knowledge Discovery* 2(2), (1998).
133. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Wadsworth Int. Group, Belmont, CA, 1984.
134. J. R. Quinlan, *Machine Learning* 1, 81 (1986).
135. S.K. Murthy, S. Kasif, and S. Salzberg, *Journal of Artificial Intelligence Research* 2, 1 (1994).
136. E. Bauer and R. Kohavi, *Machine Learning* 5, 1 (1998).
137. Y. Freund and R. E. Schapire, in "Proc. 13th Int. Conf.: Mach. Learn.," p. 148. Bari, Italy, 1996.

138. M. Su and M. Basu, in “Proc. IEEE Conf. on IJCNN” 2001, Vol. 3, p. 2159.

139. A. Bermak and D. Martinez, *IEEE Trans. Neural Networks* 14(5), 1097 (2003).

140. M. Shi, S. Brahim-Belhouari, and A. Bermak, in “IEEE International Symposium on Circuits and Systems,” Kobe, Japan, 2005.

141. M. Shi, S. Brahim-Belhouari, A. Bermak, and D. Martinez, in “International Symposium on Olfaction and Electronic Nose,” ISOEN05, Barcelone, Spain, 2005.

142. S. Bermejo and J. Cabestany, *Neural Processing Letters* 19, 25 (2004).

143. P. Duchamp-Viret, M. A. Chaput, and A. Duchamp, *Science* 284, 2171 (1999).

144. N. J. Strausfeld and J. G. Hildebrand, *Current Opinion in Neurobiology* 9, 634 (1999).

145. L. B. Buck and R. A. Axel, *Cell* 65, 175 (1991).

146. T. Kohonen, *Biological Cybernetics* 43, 59 (1982).

147. T. W. Margrie and A. T. Schaefer, *J. Physiol.* 546(2), 363 (2003).

148. J. J. Hopfield, *Nature* 376, 33 (1995).

149. M. Lysetskiy, A. Lozowski, and J. M. Zurada, *Biol. Cybern.* 87, 58 (2002).

150. G. Laurent and H. Davidowitz, *Science* 265, 1872 (1994).

151. G. Laurent, M. Wehr, and D. Davidowitz, *The Journal of Neuroscience* 16, 3837 (1996).

152. G. Laurent, *Science* 286, 723 (1999).

153. R. Friedrich and G. Laurent, *Science* 291, 889 (2001).

154. M. Bazhenov, M. Stopfer, M. Rabinovich, R. Huerta, H. D. I. Abarbanel, T. J. Sejnowski, and G. Laurent, *Neuron* 30, 553 (2001).

155. D. Martinez and E. Hugues, in “Electronic Noses and Sensors for the Detection of Explosives,” Vol. 159, p. 209. Nato Science Series, Kluwer Academic Publishers, 2004.

156. G. Yan, L. Sheng, Z. Tang, J. Wu, P. C. H. Chan, and J. K. O. Sin, *Sensors and Actuators B* 49, 81 (1998).

157. P. C. H. Chan, G. Yan, L. Sheng, R. K. Sharma, Z. Tang, J. K. O. Sin, I.-M. Hsing, and Y. Wang, *Sensors and Actuators B* 82, 277 (2002).

158. R. Gutierrez-Osuna, A. Gutierrez-Galvez, N. U. Powar, *Sensors and Actuators B* 93(1–3), 57 (2003).