

Pattern Analysis for Machine Olfaction: A Review

Ricardo Gutierrez-Osuna, *Member, IEEE*

Abstract—Pattern analysis constitutes a critical building block in the development of gas sensor array instruments capable of detecting, identifying, and measuring volatile compounds, a technology that has been proposed as an artificial substitute of the human olfactory system. The successful design of a pattern analysis system for machine olfaction requires a careful consideration of the various issues involved in processing multivariate data: signal-preprocessing, feature extraction, feature selection, classification, regression, clustering, and validation. A considerable number of methods from statistical pattern recognition, neural networks, chemometrics, machine learning, and biological cybernetics has been used to process electronic nose data. The objective of this review paper is to provide a summary and guidelines for using the most widely used pattern analysis techniques, as well as to identify research directions that are at the frontier of sensor-based machine olfaction.

Index Terms—Classification, clustering, dimensionality reduction, electronic nose, multicomponent analysis, pattern analysis, preprocessing, validation.

I. INTRODUCTION

AN electronic nose (e-nose) is an instrument that combines gas sensor arrays and pattern analysis techniques for the detection, identification, or quantification of volatile compounds. The multivariate response of an array of chemical gas sensors with broad and partially overlapping selectivities can be utilized as an “electronic fingerprint” to characterize a wide range of odors or volatile compound by pattern-recognition means. As illustrated in Fig. 1, this process can be split into four sequential stages: signal preprocessing, dimensionality reduction, prediction, and validation. The initial block in the figure represents the e-nose hardware, which typically consists of a gas sensor array, an odor delivery subsystem, an electronic instrumentation stage, and a computer for data acquisition.

The process of data analysis starts after the sensor signals have been acquired and stored into the computer. The first computational stage, called **signal preprocessing**, serves various purposes, including compensating for sensor drift, extracting descriptive parameters from the sensor array response and preparing the feature vector for further processing. A **dimensionality reduction** stage projects this initial feature vector onto a lower dimensional space in order to avoid problems associated with high-dimensional, sparse datasets. The resulting low-dimensional feature vector is then used to solve a given **prediction** problem, typically classification, regression, or clustering. *Classification* tasks address the problem of

identifying an unknown sample as one from a set of previously learned odorants. In *regression* tasks, the goal is to predict a set of properties (e.g., concentration, quality) for an analyte, typically a complex mixture. Finally, in *clustering* tasks the goal is to learn the structural relationships among different odorants. A final step, sometimes overlooked, is the selection of models and parameter settings and the estimation of the true error rates for a trained model by means of **validation** techniques.

II. PREPROCESSING TECHNIQUES FOR GAS SENSOR ARRAYS

The main purpose of a preprocessing stage is to carefully select a number of parameters that are descriptive of the sensor array response, as this choice can significantly affect the performance of the subsequent modules in the pattern analysis system [1]. Although preprocessing is somewhat tied to the underlying sensor technology, three general steps can be identified [2], [3]: baseline manipulation, compression, and normalization. **Baseline manipulation** procedures transform the sensor response relative to its baseline (e.g., response to a reference analyte) for the purposes of contrast enhancement and drift compensation, the latter requiring additional processing (Section II-A). Three baseline manipulation methods are commonly employed: difference, relative, and fractional. The *difference* method directly subtracts the baseline and can be used to eliminate additive drift from the sensor response. *Relative* manipulation, on the other hand, divides by the baseline, removing multiplicative drift, and generating a dimensionless response. *Fractional* manipulation, finally, subtracts and divides by the baseline, generating dimensionless and normalized responses. Various **compression** algorithms can be employed to generate descriptive parameters from the sensors’ transient response. The standard procedure is to select the *steady-state* response of the sensor, but a number of compression algorithms (see Table I) have been proposed to extract additional information from the *transient response*, resulting in improved selectivity, reduced acquisition time, and increased sensor lifetime [4]. Finally, **normalization** procedures prepare the feature vector for the subsequent pattern analysis modules on a local or a global fashion. Local methods operate across the sensor array for each individual “sniff” in order to compensate for sample-to-sample variations caused by analyte concentration and sensor drift, among others. The most widely used local method is *vector normalization*, in which the feature vector of each individual “sniff” is divided by its norm and, as a result, is forced to lie on a hyper-sphere of unit radius. Global methods, on the other hand, are typically used to ensure that sensor magnitudes are comparable, preventing subsequent pattern-recognition procedures from being overwhelmed by sensors with arbitrarily large values. Two global procedures are commonly employed in e-nose systems: (i) *sensor autoscaling*,

Manuscript received August 11, 2001; revised April 29, 2002. This work was supported by NSF/CAREER 9984426. The associate editor coordinating the review of this paper and approving it for publication was Dr. H. Troy Nagle.

The author is with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: rgutier@cs.tamu.edu).

Publisher Item Identifier 10.1109/JSEN.2002.800688.

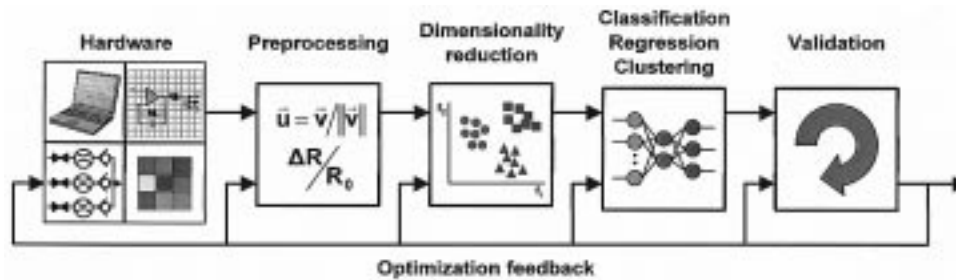


Fig. 1. Building blocks of the pattern analysis system for an electronic nose.

TABLE I
REVIEW OF PREPROCESSING TECHNIQUES
IN THE E-NOSE LITERATURE

	Technique	Sensor type*	References
Baseline manipulation	Difference	QMB, MISFET	6, 7
	Relative Fractional	MOS, CP	8, 9 10, 11
Transient compression	Sub-sampling	MOS	2
	Parameter extraction	MISFET	12
	Model fitting	MOS, CP, QMB	12, 13, 14
Normalization	Sensor, Vector, Autoscale	MOS, CP	1, 2, 15

*QMB: Quartz Crystal Microbalance, MISFET: Metal Insulator-Semiconductor Field-Effect Transistor, MOS: Metal-Oxide Semiconductor, CP: Conducting Polymer

in which the mean and standard deviation of each feature are set to zero and one, respectively, and (ii) *sensor normalization*, in which the range of values for each individual feature is set to $[0,1]$. It must be noted that these global techniques can amplify noise since all the sensors (including those that may not carry information) are weighted equally. Finally, a *logarithmic* transform can also be used to increase the dynamic range of the system [5]. Table I provides a few additional references on signal preprocessing from the e-nose literature.

A. Drift Compensation

The most serious limitation of current e-nose systems is the inherent drift of gas sensors, which results in a slow, random temporal variation of the sensor response when exposed to the same gases under identical conditions, as shown in Fig. 2(left). As a result of drift, which can affect both sensor baseline (additive) and sensitivity (multiplicative) [16], previously learned sensor patterns become obsolete over time and the system loses the ability to identify known odors. The most effective means of drift compensation is **periodic recalibration** with a reference gas that is chemically stable over time and highly correlated with the target analytes in terms of sensor behavior [17]. The array response to the calibration gas can then be directly subtracted from the response to the analytes [18], used to infer a temporal drift model for each individual sensor [17] or for the entire array [19]–[21]. As shown in Fig. 2, the multivariate direction of the drift can be obtained from the principal components (Section III-A) of the calibration gas. **Modulation** of the sensors' operating temperature can also be used to generate features that are more robust to drift than isothermal features [22], [23]. Finally, a number of computational methods based on **system identification** and self-organization principles have also been proposed for drift reduction [24]–[27].

III. DIMENSIONALITY REDUCTION

The feature vector that results from the preprocessing stage is oftentimes not suitable to be processed by a subsequent module due its high-dimensionality and redundancy. Problems with **high-dimensional** data, known as the “curse of dimensionality” in statistical pattern recognition, imply that the number of training examples must grow exponentially with the number of features in order to learn an accurate model. Since only a limited number of examples are typically available, there is an optimal number of feature dimensions beyond which the performance of the pattern analysis model starts to degrade. The problem of **redundancy**, also referred to as collinearity in chemometrics and statistics, is particularly significant in e-nose instruments due to the cross-selectivity of chemical gas sensors. When two or more feature dimensions are collinear, the covariance matrix of the entire dataset becomes singular and, therefore, noninvertible, which leads to numerical problems in various statistical techniques (e.g., quadratic classifiers and ordinary least squares). For these two reasons, a dimensionality reduction stage is required in most cases, either feature extraction or feature selection, as described in the next subsections.

A. Feature Extraction

The goal of feature extraction is to find a low-dimensional mapping $f : x \in \mathbb{R}^N \rightarrow y \in \mathbb{R}^M (M < N)$ that preserves most of the information in the original feature vector x . Two basic criteria can be employed to measure the information content of the projection: signal classification and signal representation [28]. Signal classification methods associate information with discrimination capabilities (e.g., inter-class distance) and are the preferred choice for pattern classification problems, provided that sufficient data is available. With small or high-dimensional datasets, however, these techniques have a tendency to over-fit the training data, resulting in projections that may not generalize well for test examples. Signal representation methods, on the other hand, associate information with the structure of the data (e.g., variance) and should be favored when the goal is exploratory data analysis. Most feature extraction techniques for e-nose applications have been based on linear techniques, mainly principal components analysis (PCA) and Fisher's linear discriminant analysis (LDA). **PCA** is a signal-representation technique that generates projections along the directions of maximum variance, which are defined by the first eigenvectors of Σ_x , the covariance of x [28]. **LDA** is a signal-classification technique that directly maximizes class separability, generating projections where the examples of each class

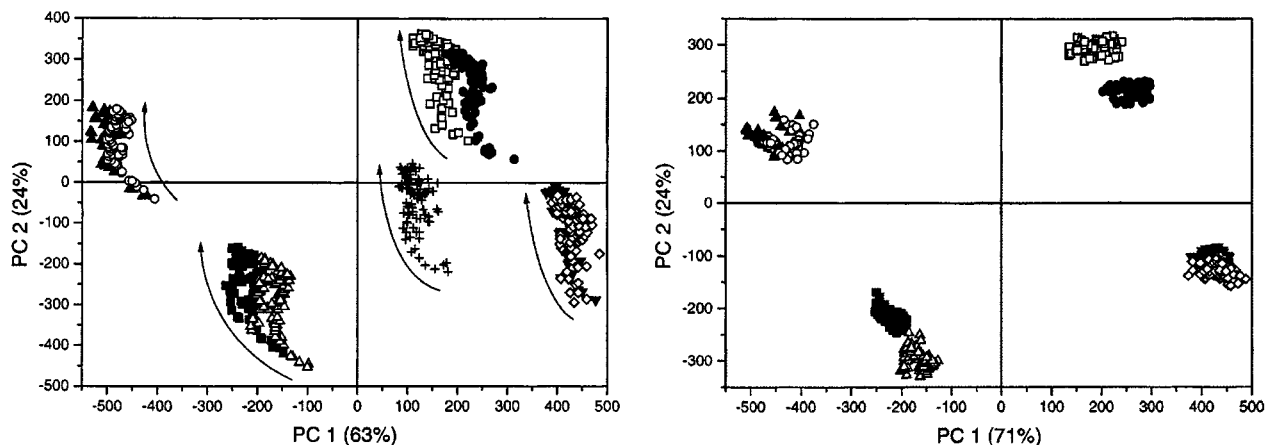


Fig. 2. PCA projections before (left) and after (right) drift compensation for an FET/MOS array under mixtures of H_2 , NH_3 , C_2H_5OH , and C_2H_4 . The center cluster (+) is the calibration gas (reproduced from [20] with permission of John Wiley & Sons).

form compact clusters and the different clusters are far from each other. These projections are, alternatively, defined by the first eigenvectors of the matrix $S_W^{-1}S_B$, where S_W and S_B are the within-class and between-class covariance matrices, respectively [29]. LDA and PCA are optimal techniques under unimodal Gaussian assumptions. For non-Gaussian distributions additional techniques may be used, including Sammon's maps [30], [31], multilayer perceptrons (MLPs) (Section IV-C), Kohonen self-organizing maps (Section VI-C), Kernel PCA, projection pursuit, and independent components analysis [32].

Fig. 3 illustrates the performance of PCA and LDA using a dataset collected by the author on an e-nose prototype [33] with 16 MOS sensors. The dataset consisted of five brand-name cookies (labels 1–5) sampled over a period of two weeks. A 64-dimensional feature vector was extracted from the sensor transients using windowed time slicing [34] and finally autoscaled. Fig. 3 shows the first two PCA and LDA projections. As expected, the LDA projections exhibit a high degree of class separability, whereas PCA preserves the original structure of the data, which includes odor- and drift-related variance.

B. Feature Subset Selection

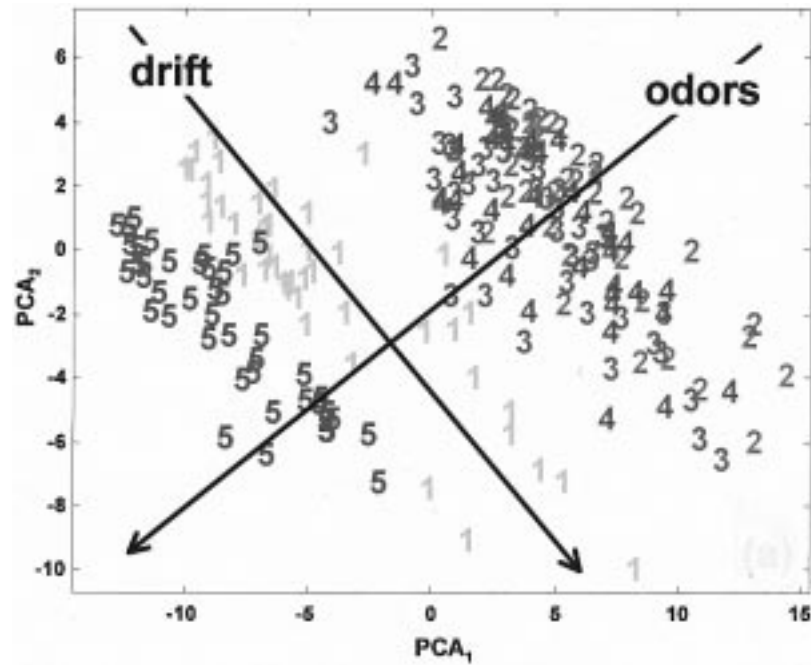
Feature subset selection (FSS) is a dimensionality reduction technique that can be used to configure small sensor arrays for specific odor-measurement applications. The goal of FSS is to find an “optimal” subset of M sensors (or features) that maximizes information content or predictive accuracy. The simplest FSS approach consists of evaluating each feature individually and selecting those M features with the highest scores. Unfortunately, this approach ignores feature redundancy and will rarely find an optimal subset. One may instead be tempted to evaluate all possible subsets of M features and select the global optimum, but the number of combinations $\binom{N}{M}$ becomes impractical even for moderate values of M and N .

To avoid the exponential explosion of an exhaustive search, several methods have been devised that explore the feature space in a more efficient fashion [35], [36]. These search strategies can be grouped into three categories: exponential, sequential, and randomized. **Exponential** techniques perform a

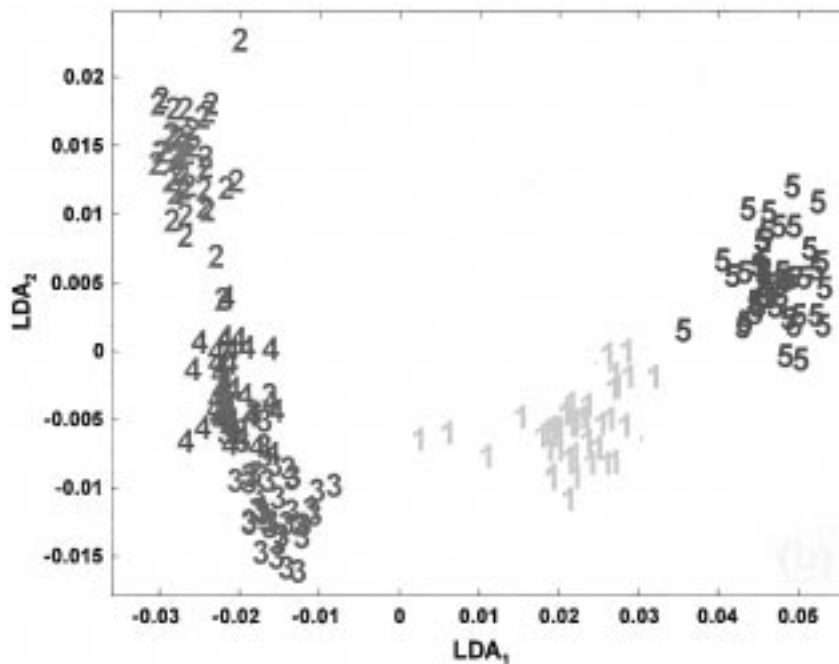
search whose complexity grows exponentially with the number of states. Among these, *branch and bound* (BB) [37] is very popular as it is guaranteed to find the optimal subset of a given size M if the evaluation function is monotonic¹. This assumption is, however, violated in practical problems since the addition of features can increase the risk of over-fitting. **Sequential** search algorithms are greedy strategies that reduce the number of states to be visited during the search by applying local search. The simplest methods are *sequential forward selection* (SFS) and *sequential backward selection* [35]. SFS starts from the empty set and sequentially adds features, whereas SBS starts from the full set and sequentially removes features. The performance of SFS/SBS may be improved by means of *sequential floating* methods with backtracking capabilities [38]. Finally, **randomized** search algorithms attempt to overcome the computational cost of exponential methods and the tendency of sequential methods to become trapped in local minima. Among these techniques, *simulated annealing* (SA) [39] and *genetic algorithms* (GA) [40] are most widely used. SA is based on the annealing process of thermal systems and performs a stochastic search on a single solution. GAs, conversely, are inspired by the process of natural selection and perform a global random search on a population of solutions.

Two strategies are available to evaluate the different feature subsets: filters and wrappers [41]. **Filters** compare feature subsets by their information content (e.g., inter-class distance), whereas **wrappers** evaluate feature subsets on the basis of their predictive accuracy on the pattern recognition algorithm, measured by statistical re-sampling or cross-validation. Each approach has a number of advantages and disadvantages [42]. The wrapper approach usually achieves better predictive accuracy since the feature subset can be tuned to the particular bias of the pattern recognition algorithm. In addition, the wrapper has a mechanism to avoid over-fitting since the feature subsets are evaluated by their performance on test data. Wrappers are, however, computationally intensive since they must continuously re-train the pattern recognition algorithm. Filters, on the other hand, tend to find a more general feature subset

¹Monotonicity implies that the addition of a new feature always improves the information content of the subset.



(a)



(b)

Fig. 3. (a) PCA versus (b) LDA for visualization and classification purposes.

that works well on a larger spectrum of pattern recognition techniques and are computationally more attractive.

1) *Review:* Eklöv *et al.* [43] used a wrapper approach combined with SFS to select features for a multilayer-perceptron (MLP) regression problem. Since an MLP wrapper was computationally impractical, the predictive accuracy of each feature subset was approximated with ordinary least squares regression. Experimental results showed that their feature selection procedure could find small (5–10) feature subsets with similar or better predictive accuracy than the complete set of 49–85 features. Gutierrez-Osuna [33] compared the

performance of eight search strategies on odor classification problems. The wrapper consisted of LDA combined with K nearest neighbors (Section IV-B) and predictive accuracy was estimated through five-fold cross validation (Section VII). This study indicated that all the search techniques perform similarly, yielding 25–30% increase in predictive accuracy while reducing the size of the feature set by 50%. Corcoran [44] have used a genetic algorithm and a filter based on Fisher's discriminant ratio [45] to select features from a temperature-modulated sensor array. Their procedure was able to reduce the feature set by 1/10 while maintaining classification rates on a rather

oversized MLP. In a second article [46], the authors report on the difficulties associated with finding a filter that correlates well with predictive accuracy. This observation is consistent with arguments in favor of wrappers.

IV. CLASSIFICATION

The goal of a pattern classifier is to generate a class label prediction ω_{PRED} for an unknown feature vector $y \in \mathfrak{X}^M$ from a discrete set of C previously learned labels $\{\omega_1, \omega_2, \dots, \omega_C\}$. It can be shown that to minimize classification errors² one should assign example y to the class ω_i with the largest posterior probability $P(\omega_i|y)$ [29]. This is known as the Maximum *a posteriori* (MAP) rule and is the best any classifier can do. The computation of $P(\omega_i|y)$ can be performed by applying Bayes Theorem, yielding

$$\begin{aligned} \omega_{MAP} &= \arg \max_{i \in \{1, C\}} [P(\omega_i|y)] \\ &= \arg \max_{i \in \{1, C\}} \left[\frac{P(y|\omega_i) P(\omega_i)}{P(y)} \right] \\ &= \arg \max_{i \in \{1, C\}} [P(y|\omega_i) P(\omega_i)] \end{aligned} \quad (1)$$

where $P(y|\omega_i)$ is the likelihood or class-conditional density, $P(\omega_i)$ is the prior probability, and $P(y)$ serves as a normalization constant that can be ignored for classification purposes³. In the absence of prior knowledge, $P(\omega_i)$ can be approximated by the relative frequency of examples in the dataset. Direct estimation of $P(y|\omega_i)$ from high-dimensional training data is, however, extremely difficult [47] unless strong simplifying assumptions about the underlying distributions are made (e.g., Gaussian). From this perspective, most pattern classifiers can be interpreted as attempts to estimate $P(y|\omega_i)$ from data.

A. Quadratic Classifiers

The simplest possible way to estimate $P(y|\omega_i)$ is to assume that the likelihood function of each class is a unimodal (this is, containing a single mean) Gaussian density

$$P(y|\omega_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp \left[-\frac{1}{2} (y - \mu_i)^T \Sigma_i^{-1} (y - \mu_i) \right] \quad (2)$$

where μ_i and Σ_i are the sample mean and sample covariance, respectively, estimated from training data. Merging (2) into the MAP rule (1) and taking natural logarithms yields the following decision rule:

$$\omega_{QUAD} = \arg \max_{i \in \{1, C\}} \left[-\frac{1}{2} (y - \mu_i)^T \Sigma_i^{-1} (y - \mu_i) - \frac{1}{2} \log |\Sigma_i| + \log P(\omega_i) \right]. \quad (3)$$

²Misclassification costs (Bayes Risk) could also be incorporated into the classifier. Please refer to the article by Pardo and Sberveglieri in this special issue for additional details.

³The function $\arg \max()$ returns the class label that maximizes the posterior (the argument), not the probability value itself. Therefore the equality holds after removing $P(y)$ since this does not depend on the class index i . However, $P(y)$ would be necessary if not only the best class assignment but also a confidence level in that assignment was required.

This is known as a *quadratic* classifier since the decision boundaries between classes are quadratic hyper-surfaces. The term $(y - \mu_i)^T \Sigma_i^{-1} (y - \mu_i)$ is known as the *Mahalanobis distance*, which can be thought of as a stretching factor of feature space along the directions of maximum variance. When the covariance Σ_i is the identity matrix, the Mahalanobis distance becomes the familiar Euclidean distance, and the quadratic classifier reduces to the simple nearest-mean classifier (assuming equiprobable priors). Quadratic classifiers are, therefore, Bayes optimal if the classes are unimodal Gaussian. They are computationally attractive, with only one caveat: estimation of the covariance matrix can be difficult as a result of collinearity. When the number of independent examples per class is lower than the dimensionality, the sample covariance becomes singular and, therefore, noninvertible. In this case, it is common practice to assume that all the classes have the same structure and a single covariance matrix is estimated from the entire dataset, regardless of class. This is known as a *linear* classifier because the decision boundaries in feature space become hyper-planes. Intermediate solutions between the quadratic and the linear classifier by shrinkage methods have also been proposed in the statistics literature [48].

B. K Nearest Neighbor Classifiers

The K nearest neighbors (kNN) rule is a powerful technique that can be used to generate highly nonlinear classifications with limited data. To classify example y , kNN finds the closest k examples in the dataset and selects the predominant class among those k neighbors. Although this formulation appears rather heuristic, kNN is formally a nonparametric⁴ approximation of the MAP criterion (1). kNN can generate highly local decision regions by choosing an appropriate value for k and presents very attractive asymptotic properties: as the number of examples approaches infinite the probability of error for the ($k = 1$) NN classifier will not be worse than twice the Bayes error, the best any classifier can achieve [29]. The main limitations of kNN are (i) storage requirements, since the entire dataset needs to be available during recall and (ii) computational cost, since for each unlabeled example, the distance to all training examples needs to be computed (and sorted). These limitations, however, can be overcome by editing the training set and generating a subset of prototypes [49]. Special attention must be paid to the scaling of each feature dimension, to which kNN is extremely sensitive.

C. Multilayer Perceptron Classifiers

MLPs, the most popular type of artificial neural networks, are feed-forward networks of simple processing elements or neurons whose connectivity resembles that of biological neuronal circuitry. Each neuron in an MLP performs a weighted sum of its inputs and transforms it through a nonlinear activation function, typically a squashing sigmoidal. An MLP is able to learn arbitrarily complex nonlinear regressions by adjusting the weights in the network by a gradient descent technique known as back-propagation of errors [50]. At each stage in the training process, the MLP processes all of its inputs in a feed-forward

⁴Parametric methods assume that the underlying distribution can be represented with a parameterized model (e.g., a Gaussian density, with mean and standard deviation as parameters). Non-parametric models do not make such assumptions (e.g., a histogram).

fashion, compares the resulting outputs with the desired ones and back-propagates these errors to adjust each weight in the network according to its contribution to the overall error. A number of heuristics are available to improve the operation of back-prop [29], [51], [52].

To avoid over-fitting, the complexity of the MLP can be controlled by **limiting the network size**. With the number of inputs and outputs being determined by the application, network size is strictly a function of the number of hidden layers and hidden neurons per layer. It can be shown [53] that one-hidden-layer MLPs can approximate any classification boundaries with arbitrary accuracy, provided that the number of hidden units is large enough. Nonetheless, the addition of extra hidden layers may allow the MLP to perform a more efficient approximation with fewer weights [54]. Despite a number of “rules of thumb” published in the literature, *a priori* determination of an appropriate number of hidden units is still an unsolved problem. The “optimal” number of hidden neurons depends on multiple factors, including complexity of the classification problem, number of inputs and outputs, activation functions, training algorithm, number of examples, and level of noise in the data. In practice, several MLPs are trained and evaluated in order to determine an appropriate number of hidden units. Constructive approaches, which start with a small MLP and incrementally add hidden neurons [55] and pruning approaches, which start with a relatively large MLP and sequentially remove weights [56], [57], can also be employed.

Network complexity can also be controlled by **constraining the weight values**. Regularization approaches such as *weight decay* are very effective at generating smooth mappings [29]. *Early stopping* may also be applied to prevent the MLP from over-fitting the training set [58]. The stopping point may be determined by monitoring the sum-squared-error of the MLP on a validation set during training. Finally, *training with noise*, also known as jitter, can be used to prevent the MLP from approximating the training set too closely and has been shown to improve generalization [59]. For additional details on complexity control, please refer to the tutorial article by Pardo and Sberveglieri in this special issue.

Finally, it is important to mention some remarkable similarities between MLPs and the statistical concepts covered in previous sections. It has been shown [60] that, using linear output neurons with one-from-C binary output encoding, minimization of the mean-square error at the output of an MLP is equivalent to maximizing a nonlinear version of the objective function employed in Fisher’s LDA, allowing the hidden neurons to function as discriminatory feature extractors. In addition, when the MLP is trained to minimize the cross-entropy cost function [61] with one-from-C encoding and the output layer has softmax activation functions, the outputs of the MLP will approximate the posterior $P(\omega_i|y)$.

D. Radial Basis Function Classifiers

Radial basis functions (RBFs) are feed-forward connectionist architectures consisting of a hidden layer of radial kernels and an output layer or linear neurons [54]. Although the structure of RBF networks resembles that of MLPs, their input-output mappings and training algorithms are fundamentally different.

Each hidden neuron in an RBF is tuned to respond to a rather local region of feature space by means of a radially symmetric function such as the Gaussian $N(\mu_j, \sigma_j)$. The output units, on the other hand, form linear combinations of the hidden units to predict the output variable in a similar fashion to MLPs.

RBFs are typically trained using a hybrid algorithm that employs unsupervised learning for the hidden layer followed by supervised learning of the output layer. The first step consists of selecting the radial basis centers μ_i using C-means clustering (Section VI-B). Then, the spreads σ_j are determined from the average distance between neighboring cluster centers [62] or the sample covariance of each cluster [63]. Alternatively, one may use the expectation maximization algorithm [54] to estimate both μ_i and σ_j simultaneously. Finally, training the output layer is a straightforward supervised problem, in which the radial basis activations are used as regressors to predict the target outputs. This can be efficiently solved using ordinary least squares (Section V-A) since the output neurons have linear activation functions.

Although very efficient, this hybrid training procedure is very sensitive to noisy dimensions with high variance, which may prevent the unsupervised stage from extracting clusters that are predictive of the output variables. To overcome this limitation, Chen *et al.* [64] introduced *orthogonal least squares*, a supervised technique that performs forward stepwise selection of radial basis functions from the training examples. At each step, the algorithm chooses the basis function that provides the greatest increment in explained variance in the output variables. This algorithm is able to generate not only the radial basis centers but also the hidden-to-output weight matrix and an appropriate number of radial basis. For a thorough survey of RBFs, the reader is referred to [52], [54].

E. Comparison Between Quadratic, kNN, MLP, and RBF Classifiers

MLPs and radial basis functions are the two most popular types of neural network architectures employed in the e-nose community. Although both models can function as “universal approximators,” it is important to highlight some of their differences [45], [52], [54]. (i) MLPs perform a global and distributed approximation of the target function, whereas RBFs perform a local approximation. (ii) The distributed representation of MLPs causes the error surface to have multiple local minima and nearly flat regions with very slow convergence. As a result, training times for MLPs are usually larger than those for RBFs. (iii) MLP partition feature space with hyper-planes; RBF decision boundaries are hyper-ellipsoids. (iv) MLPs exhibit better generalization properties than RBFs in regions of feature space outside of the local neighborhoods defined by the training set. On the other hand, extrapolation far from training data is dangerous. (v) MLPs typically require fewer parameters than RBFs to approximate a nonlinear function with the same accuracy. (vi) MLPs may have multiple layers with complex connectivity, whereas RBFs typically have only one hidden layer and full connectivity.

The various decision regions/boundaries that can be generated by the algorithms described in this section are illustrated in Fig. 4. The quadratic classifier models each likelihood function

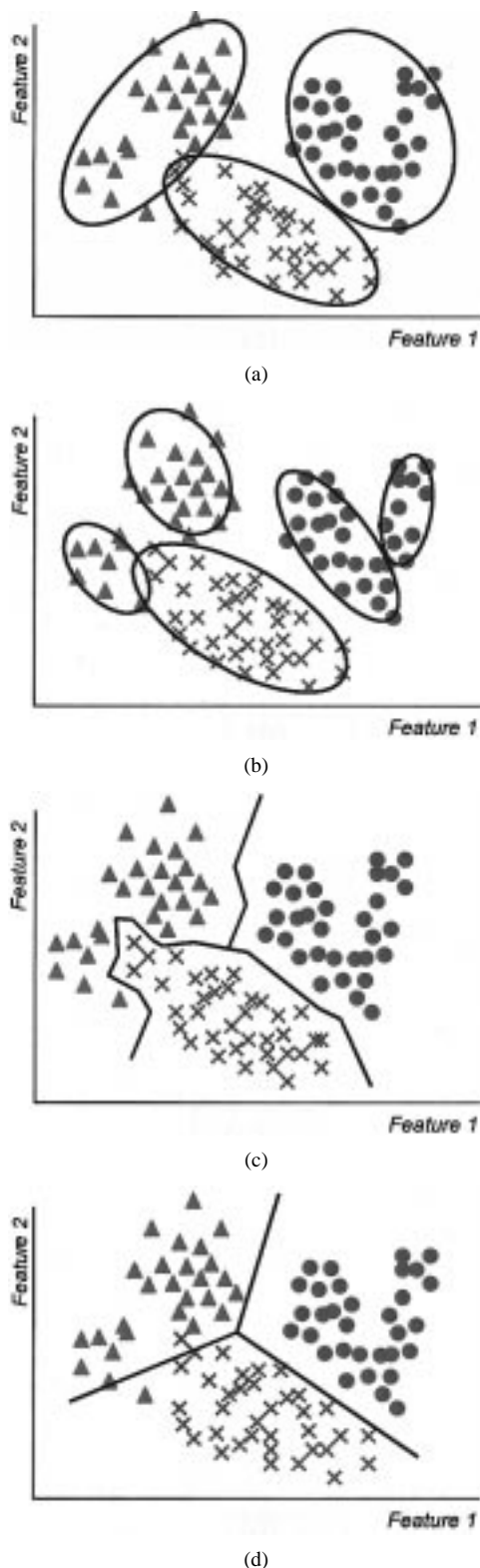


Fig. 4. Class separation for (a) quadratic, (b) RBF, (c) kNN, and (d) MLP (adapted from [54]).

$P(y|\omega_i)$ with a unimodal Gaussian density. The resulting decision boundaries, which are defined by the intersection of the equiprobable likelihood contours, are quadratic surfaces. The RBF classifier models the distribution of examples as a Gaussian mixture, which is more accurate than the Quadratic approach for multimodal data. The kNN classifier creates a

TABLE II
REPRESENTATIVE PUBLICATIONS IN CLASSIFICATION AND REGRESSION

Classification		Regression	
Technique	Reference	Technique	Reference
QUAD	74, 75	OLS, PCR	80
kNN	2, 74, 76	RR	69
MLP	74, 75, 76, 77, 78	PLS	6, 7, 80, 81, 82
RBF	78, 79	MLP	6, 7, 83

fine-grained partition of feature space, with polygonal decision boundaries around each training example, known as a Voronoi tessellation [65]. The MLP decision boundaries, at last, consist of hyper-planes in feature space. Fig. 4(d) illustrates the case of a reasonably sized MLP but, given enough hidden units, MLPs can generate arbitrarily complex decision boundaries, such as the one obtained for the kNN classifier. Finally, Table II provides a few representative publications of these pattern classification methods in e-nose applications. Additional references can be found in [66], [67].

V. REGRESSION

Regression problems constitute a more challenging domain for e-nose instruments. The goal of regression is to establish a predictive model from a set of independent variables (e.g., gas sensor responses) to another set of continuous dependent variables. Pattern classification could, therefore, be treated as a regression problem where the dependent variable is categorical. For this reason, most regression techniques can be (and have been) applied for classification purposes. Three basic regression problems have been addressed with e-nose instruments: multicomponent analysis, process monitoring, and sensory analysis. In **multicomponent analysis**, the dependent variable is the concentration of an isolated analyte or the relative concentration of P known components in a mixture. Multicomponent analysis is limited, in practice, to a few components (2–4) due to sensor cross-selectivity and the exponential growth in the number of calibration points with increasing values of P . In **process monitoring**, the dependent variable is a process variable (e.g., quality level) associated with an analyte that may be embedded in a matrix of unknown compounds. The broad selectivity of chemical gas sensors is an important shortcoming of e-noses for this type of regression problems. Finally, in **sensory analysis**, the dependent variable is the score of a human sensory panel (e.g., intensity, hedonic tone, organoleptic descriptors). Needless to say, mimicking the perception of odors by humans is the ultimate challenge for machine olfaction and an extremely complex regression problem. The following subsections review a number of linear regression techniques rooted in statistics and chemometrics that can be used to obtain a first-order regression model. For clarity, the following derivations will use a multicomponent analysis formulation in which c is a vector of concentrations for the P components in a mixture.

A. Ordinary Least Squares

A simple approach to regression is to assume that the dependent variables $c_{1 \times P}$ can be predicted from a linear combination of the sensor responses $y_{1 \times M} : c = yW$, where we have

adopted the chemometrics row-vector notation. Without loss of generality, we also assume that the dependent and independent variables are mean-centered. Alternatively, a constant dimension could be added to the feature vector to absorb the intercept. Given a matrix $C_{Q \times P}$ of Q calibration mixtures and their corresponding sensor responses $Y_{Q \times M}$ (one example per row), the prediction matrix $W_{M \times P}$ may be computed as $W = Y^{-1}C$. Unfortunately, this exact solution is only possible when Y is nonsingular. In practice, Y is rectangular with more rows (examples) than columns (sensors) or vice versa and, therefore, the inverse Y^{-1} does not exist. For this reason, it is common to seek the solution that minimizes the sum-squared prediction error over the entire calibration set. This is known as the ordinary least-squares (OLS) solution and is given by [29], [54]

$$W = (Y^T Y)^{-1} Y^T C = Y^\dagger C.$$

The matrix $Y^\dagger = (Y^T Y)^{-1} Y^T$ is known as the pseudo-inverse of Y , since $Y^\dagger Y = I$. Application of the OLS solution to e-nose calibration may result in numerical problems since the covariance matrix $Y^T Y$ can become singular or near-singular as a result of collinearity. These problems can be reduced by using shrinkage methods [68], which are briefly reviewed in the next subsections.

B. Ridge Regression

Ridge regression (RR) is a regularization method that stabilizes the OLS solution by adding a multiple of the identity matrix⁵ to the estimation of the covariance matrix $Y^T Y$ [48], [69]

$$W = \left((1 - \gamma) Y^T Y + \gamma \frac{\text{tr}(Y^T Y)}{M} I \right)^{-1} Y^T C$$

where $\gamma (0 \leq \gamma \leq 1)$ is a regularization parameter that controls the amount of shrinkage toward the identity matrix. For $\gamma = 0$, RR is equivalent to OLS, whereas for $\gamma = 1$, the solution becomes a constant model that always predicts the mean concentration of the training data. Selection of an appropriate value for γ is typically performed through cross-validation (Section VII).

C. Principal Components Regression

An alternative solution to the OLS collinearity problem is to perform PCA and retain only a few of the principal components as regressors or “latent variables,” hence the name principal components regression (PCR). As a result, the regressors are effectively decorrelated and, more importantly, the smaller eigenvalues of $Y^T Y$, which become infinite when computing the pseudo-inverse, are eliminated. The number of principal components to keep for the regression can also be determined through cross-validation. However, PCR presents similar problems to PCA since the directions of maximum variance in Y may not necessarily be correlated with the dependent variables.

⁵This expression assumes that the sensors have the same variance. Otherwise, the regularization term $\text{tr}(Y^T Y)I/M$ must be replaced by a diagonal matrix of the sensor variances [70].

D. Partial Least Squares

Partial least squares (PLS) is the “gold standard” in chemometrics due to its ability to handle collinear data and reduce the number of required calibration samples [71], [72]. As opposed to PCR, which extracts the “latent variables” from the directions of maximum variance in the sensor matrix Y (the eigenvectors of $Y^T Y$), PLS finds the directions of maximum correlation between Y and C in a sequential fashion. The first PLS latent variable ($t = Yw$) is obtained by projecting Y along the eigenvector w corresponding to the largest eigenvalue of $Y^T C C^T Y$ [73]. To find the second and subsequent latent variables, Y is deflated by its OLS prediction from the current PLS latent variable and the eigen-analysis is repeated. A stopping point for the sequential expansion is determined through cross-validation.

Table II provides references for published work on multicomponent analysis using gas sensor arrays. Additional references may be found in [66]. Non-linear extensions of these methods and connectionist approaches such as MLPs or RBFs may be employed if the relationships are found to be highly nonlinear. This may be the case, for instance, in multicomponent analysis since the concentration-response dependence of most gas sensors is nonlinear.

VI. CLUSTERING

Clustering is an unsupervised learning process that seeks to find spatial relationships or similarities among data samples, which may be hard to discern in high-dimensional feature space. The process of clustering involves three basic steps: (i) defining a dissimilarity measure between examples, typically the Euclidean distance, (ii) defining a clustering criterion to be optimized, typically based on within- and between-cluster structure (e.g., elongated, compact or topologically-ordered clusters), and (iii) defining a search algorithm to find a “good” assignment of examples to clusters, since exhaustive enumeration of all possible clusterings is clearly unfeasible. In most cases, a final validation by domain experts is required since, unlike supervised procedures whose results can be objectively measured (e.g., mean-squared-error), clustering results can be rather subjective. In the following subsections we review the basic clustering techniques that have been applied to process e-nose data.

A. Hierarchical Clustering

These algorithms are capable of generating a multi-level clustering or taxonomy of examples using a tree structure known as a dendrogram [see Fig. 5(a) and (b)]. These dendrograms can be built in a bottom-up or top-down fashion, giving rise to two types of algorithms: agglomerative and divisive, respectively. **Agglomerative** algorithms construct the dendrogram starting at the leaves, where each example forms a unique cluster and proceed toward the root by sequentially merging the two “nearest” clusters. A measure of cluster similarity is used to determine which two clusters should be merged each time, typically minimum distance or maximum distance between examples from each cluster. Minimum-distance (single-linkage) favors elongated clusters, whereas maximum-distance (complete-linkage) generates compact clusters [29]. **Divisive** clustering algorithms

proceed in the opposite direction, constructing the dendrogram from the root, where all examples belong to one cluster and sequentially splitting the “worst” cluster until each cluster contains exactly one example. To determine the “worst” cluster at a given iteration, the algorithm must tentatively split all clusters and select the one whose two children have highest dissimilarity. This is a computationally intensive task and, for this reason, divisive clustering has received much less attention. Divisive clustering, however, is more likely to produce meaningful results than agglomerative methods for small number of clusters [70].

B. C-Means

C-means is a clustering algorithm that generates a single-level partition of the dataset, this is, an assignment of training examples into C disjoint clusters. Starting from an initial clustering (e.g., a random assignment of examples to clusters), C-means iteratively re-computes the sample mean of each cluster and reassigns each example to the cluster with the closest mean. This procedure is repeated until the assignment of examples to clusters no longer changes between iterations. Although the basic C-means algorithm requires a pre-specified number of clusters, heuristic procedures [84] can be employed to automatically determine an appropriate number of clusters.

C. Self-Organizing Maps

Self-organizing maps (SOMs) are connectionist techniques capable of generating topology-preserving clusterings [85]. An SOM is a network of clusters (or neurons) arranged in a lattice structure, typically two-dimensional. The behavior of SOMs results from the synergy of three processes: competition, cooperation, and adaptation [52]. First, all neurons in the lattice enter a *competition* for each incoming example. The closest neuron in feature space is selected as a winner and becomes activated. Neurons in the topological neighborhood of the winner also become activated in order to promote a topological ordering in feature space. Finally, all active neurons *adapt* their coordinates in feature space to become closer to the input pattern. SOMs have very interesting properties for data visualization but mapping onto the SOM manifold can be tricky if the structure of the data is inherently high dimensional.

D. Review

Clustering techniques have been widely used to explore and visualize e-nose data. Gardner [10] has used hierarchical clustering to analyze the response of 12 MOS sensors to different alcohols. As shown in Fig. 5(a) and (b), the dendrograms reveal good separation of the alcohols, particularly after sensor normalization. Zupan *et al.* [86] have employed Kohonen SOMs to analyze 572 Italian olive oils from nine different regions on the basis of the contents of eight fatty-acids, determined through analytical chemistry. Their SOM is able to separate northern and southern oils, with a clear gap of inactive neurons between both groups. Additional applications of clustering techniques for the processing of e-nose data may be found in [87], [88] for hierarchical clustering and in [89]–[91] for SOMs.

VII. VALIDATION

The previous sections have reviewed a number of pattern recognition techniques that can be utilized to process e-nose data. This final section addresses the issues of model selection and performance estimation. When facing a new application, the data analyst must determine not only an appropriate model among such a wide variety of processing algorithms but also the parameter settings of the model to achieve “optimal” performance. Any reasonable measure of performance must be associated with the ability of the model to predict new data or unveil the fundamental structure rather than the accidental correlations in the training data. The latter occurs when the model is allowed to over-fit the data, typically as a result of an unreasonably large number of model parameters or excessive training iterations (Section IV-C).

To avoid over-fitting it is customary to split the available data into training and validation sets. The training set is used to learn several models with different structures or learning meta-parameters. The trained model that performs best on the validation data is then selected as the final model. This simple validation technique is known as the **holdout** method and is illustrated in Fig. 6(a). Although the holdout works well in many situations, it has two drawbacks. First, in problems with limited data one may not be able to afford the luxury of setting aside a portion of the dataset for validation. Second, being a single train-and-validate experiment, the holdout estimate of performance can be misleading if we happen to get an unfortunate split [70]. The shortcomings of the holdout method can be overcome, at the expense of additional computation, by performing multiple partitions of the dataset and averaging the performance of the model across partitions. **K-fold cross-validation**, shown in Fig. 6(b), performs K data partitions in a way that each example is eventually used for both training and validation. At each of the K splits N/K examples are used for validation and the remaining $N(K-1)/K$ are used for training, where N is the total number of examples. When the number of folds is set equal to the number of examples ($K = N$), the method is known as **leave-one-out** (LOO) cross-validation.

When employing multiple partitions, the final model is then selected based on the average performance over the K data partitions. The behavior of this average estimate will clearly depend on K . With large K , the bias of the estimate will be small, but its variance across partitions will be large. With small K , on the other hand, the variance of the estimate will be small, but its bias will be large and conservative (pessimistic) since the effective number of training examples is reduced. The choice of the number of splits depends largely on the amount of data. For large datasets, a small value of K ($\cong 3$) will be sufficient. For very sparse datasets, on the other hand, one may have to use LOO in order to train on as many examples as possible. Constraints in computational resources may also be taken into consideration since execution times will definitely increase with K .

Better performance estimates, including their bias and variance, can be obtained with computer-intensive techniques such as the **bootstrap**[92], [93], a statistical technique that generates multiple training-test partitions $Y^{*(b)}$ by resampling the original dataset Y with replacement, as illustrated in Fig. 6(c). Ex-

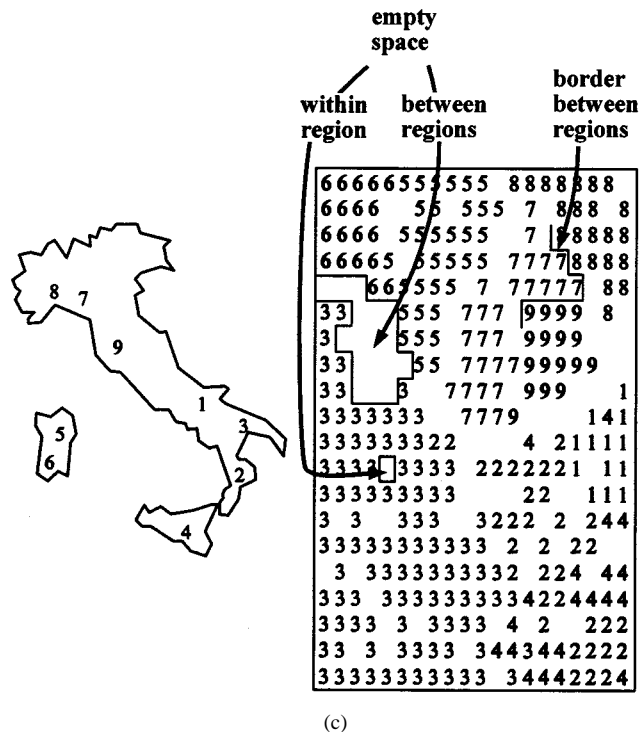
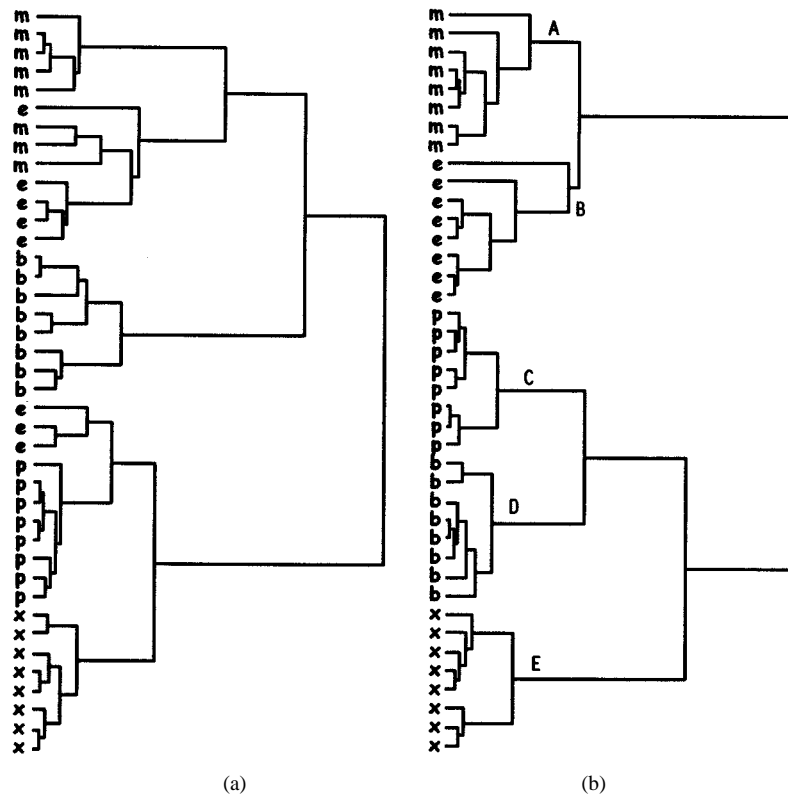


Fig. 5. Hierarchical clustering of five alcohols with (a) raw and (b) normalized sensor data (reproduced from [10] with permission of Elsevier Science). (c) Self Organizing Map on olive oils from nine different geographical locations in Italy (reproduced from [86] with permission of Elsevier Science).

amples that are not selected for training become the validation set. The underlying principle behind the Bootstrap is that we can learn about the effect that sampling the entire population had on our dataset Y by studying the effect that re-sampling Y has on the bootstrap partitions $Y^{*(b)}$. In addition, the bootstrap can be used to improve performance by training multiple instances of a learning algorithm on different data partitions and combining

their results with a voting or weighting scheme [94], [95]. These ensemble learning methods are covered in the tutorial article by Pardo and Sberveglieri.

A. Three-Way Data Partitions

Once model and parameter settings have been selected, it is still necessary to obtain an estimate of how well the final model

TABLE III
GUIDELINES FOR PATTERN ANALYSIS IN MACHINE OLFACTION

Baseline manipulation: The fractional change in conductance is a suitable measure for chemoresistors, which helps linearize the concentration-response curve (MOS and CP) and compensate for temperature fluctuations (MOS) [96, 97].

Compression: Transient response analysis can improve the selectivity of e-nose systems, but it also increases the dimensionality of the feature vector, which requires an exponentially increasing number of training examples to accurately train the pattern-recognition model.

Normalization: Vector normalization should only be used when discrimination is to be performed on the basis of odor quality (the direction of the feature vector) rather than odor intensity (its magnitude). Therefore, it is not appropriate for most regression problems, where analyte concentration is a variable of interest.

Drift compensation: Commonly associated to sensor drift (a random process) are artifacts caused by improper instrument pre-heating, recovery times or sample preparation/conditioning. These problems are best treated at their source rather than through signal processing.

Dimensionality reduction: Despite its shortcomings, PCA is an invaluable tool to visualize the structure of the data. For classification problems with unimodal likelihoods, however, Fisher's LDA is the "gold standard." In terms of feature subset selection, sequential floating search combined with a fast wrapper (e.g., linear regression [43]) will generally provide a good solution.

Classification and regression: MLPs and RBFs are "universal approximators" capable of solving any classification or regression problem, but they are not always necessary. If non-linearities in the sensor array response can be compensated for through preprocessing (e.g., a logarithmic transform), a simpler model (e.g., linear, parametric) is generally preferred. For linear or near-linear problems, the performance of PLS is hard to match.

Clustering: Dendrograms can effectively unfold the hierarchical relationship among different analytes, but they are not best suited for visualizing large numbers of samples. In this case, PCA scatter plots can be utilized to identify data clusters visually or through c-means.

Validation: Pattern recognition performance should be measured following an experimental design that reproduces field operation and reduces systematic errors (e.g., temporal correlations in the samples). Since drift and cross-sensitivities cannot be fully accounted for, it is wise to use a conservative measure such as the bootstrap's E0 estimate (see [29], pp. 474-475; [93], pp. 373-377).

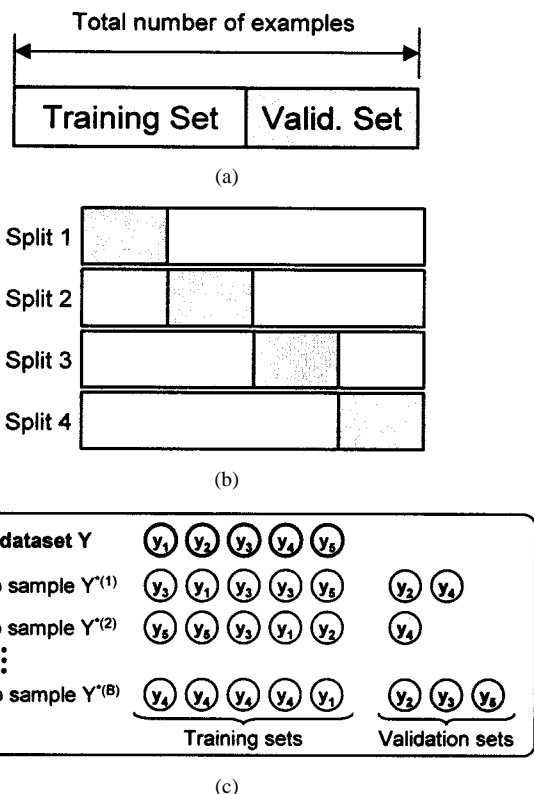


Fig. 6. Validation techniques: (a) the holdout, (b) K-fold cross-validation, and (c) the bootstrap.

will perform on new data. One may be tempted to use the performance estimate for the validation set(s) since it is readily avail-

able. This will be, however, an exceedingly optimistic estimate since the final model was chosen to minimize error rate on that particular validation data. To obtain an independent measure of performance it is necessary to use a third subset containing data that was not previously used at all, either for training or selecting the model. This argument gives rise to a data-partitioning scheme with three subsets: training, validation, and test sets [70]. The **training set** is used for learning the parameters of a model. In the case of an MLP, the training set would be used to find the "optimal" weights with the back-propagation rule. The **validation set** is used to tune the meta-parameters of the model. In an MLP, the validation set would be used to find the "optimal" number of hidden units or to determine a stopping point for the back-propagation algorithm. Finally, the **test set** is used only to assess the performance of a fully trained model. In an MLP, the test would be used to estimate the error rate after the final model (MLP structure and weights) has been determined.

VIII. CONCLUSIONS AND OUTLOOK

This paper has presented an overview of the most relevant statistical, chemometrics, connectionist and machine learning approaches for e-nose data analysis, including signal pre-processing, dimensionality reduction, classification, regression, clustering, and validation. Although the most appropriate approach clearly depends on the specific sensor type(s) and application domain, a few general guidelines can be drawn to help the reader select a reasonable starting point, as summarized in Table III.

A few additional approaches, not covered in this review due to space constraints, hold a promising future for the

processing of electronic nose data, particularly fuzzy, adaptive, and biological cybernetics paradigms. **Fuzzy logic** provides a useful framework for representing uncertainty in sensor data, model parameters, and outputs. The fuzzy paradigm may be particularly relevant for mimicking the organoleptic perception of odors by humans, arguably the ultimate goal of machine olfaction. The use of fuzzy sets [98] has been reported in the e-nose literature [99]–[102], including hybrid approaches such as fuzzy C-means/RBFs [103], [104], fuzzy MLPs [105], [106], and fuzzy learning vector quantization [107]. **Adaptive** techniques have also been explored for on-line learning in non-stationary environments [25], [26]. Adaptive resonance theory (ART) [108], in particular, provides a mechanism that solves the “stability-plasticity dilemma,” which refers to the inability of most learning systems to adapt to changing environments without compromising previously acquired knowledge. For this reason, the family of ART algorithms has been proposed as a processing mechanism for e-nose data [109]–[111]. ART is a plausible model of human information processing but, unfortunately, has a tendency to over-fit, which result in a proliferation of categories in the presence of noisy data [70]. Finally, the study of signal-processing mechanisms in the biological olfactory system constitutes a promising direction for future work [67], [112]. The wealth of **computational models of the olfactory pathway** developed in biological cybernetics and computational neuroscience [113]–[116] can serve as a starting point to mimic biological olfactory processes including (i) receptor-glomerular convergence for improved sensitivity and fault tolerance in large sensor arrays, (ii) bulbar excitatory-inhibitory dynamics for odor contrast enhancement and normalization, (iii) cortical associative memory functions for pattern completion, and (iv) centrifugal modulation of the olfactory bulb for chemosensory adaptation. A few recent publications have begun to explore the use of these mechanisms on chemical sensor arrays [117]–[120].

We conclude this review with a summary of the topics that, in our opinion, are at the frontiers of pattern analysis for electronic nose instruments. A few of these are of critical importance in order for the technology to receive widespread acceptance as a viable approach for the measurement of odors and volatile compounds. (i) **Drift compensation** algorithms, in combination with appropriate sampling and recalibration procedures, are the most immediate need for industrial applications. (ii) **Calibration transfer** algorithms also need to be developed in order to avoid a complete retraining of the pattern-analysis system for each individual instrument. (iii) For these reasons, the development of odor, sampling, and calibration **standards** is of utmost importance. (iv) The use of **transient, dynamic, and multimodal** sensor information is a promising area for improving the selectivity of existing sensor technologies. (v) **Processing of mixtures** and detection of volatiles in complex matrices are challenging problems that will require the combination of pattern-recognition and analytical sample pre-conditioning procedures. (vi) As reviewed in this article, a wide range of algorithms has been utilized for processing e-nose data. In order to establish the relative merits of these algorithms and, more importantly, the contribution of future approaches, it will be necessary to develop **dataset repositories** similar to those that have

been promoted in machine learning (UC Irvine) and speech processing (TIMIT), to mention a few. (vii) To conclude, we believe that the development of **biologically plausible computational models** of human olfactory perception and processing constitutes a grand challenge for the machine olfaction of the future.

ACKNOWLEDGMENT

Constructive comments from the anonymous reviewers were very helpful in improving the final version of this manuscript.

REFERENCES

- [1] J. W. Gardner, M. Craven, C. Dow, and E. L. Hines, “The prediction of bacteria type and culture growth phase by an electronic nose with a multi-layer perceptron network,” *Meas. Sci. Technol.*, vol. 9, pp. 120–127, 1998.
- [2] R. Gutierrez-Osuna and H. T. Nagle, “A method for evaluating data-preprocessing techniques for odor classification with an array of gas sensors,” *IEEE Trans. Syst. Man Cybern. B*, vol. 29, pp. 626–632, May 1999.
- [3] R. Gutierrez-Osuna, H. T. Nagle, B. Kermani, and S. S. Schiffman, “Signal conditioning and pre-processing,” in *Handbook of Machine Olfaction: Electronic Nose Technology*, T. C. Pearce, S. S. Schiffman, H. T. Nagle, and J. W. Gardner, Eds. Weinheim, Germany: Wiley-VCH, 2002.
- [4] E. Llobet, J. Brezmes, X. Vilanova, X. Correig, and J. Sueiras, “Qualitative and quantitative analysis of volatile organic compounds using transient and steady-state responses of a thick-film tin oxide gas sensor array,” *Sens. Actuators B*, vol. 41, no. 1–3, pp. 13–21, 1997.
- [5] T. C. Pearce, J. W. Gardner, and W. Gopel, “Strategies for mimicking olfaction: The next generation of electronic noses?,” *Sens. Update*, vol. 3, no. 1, pp. 61–130, 1998.
- [6] A. Hierlemann, U. Weimar, G. Kraus, M. Schweizer-Berberich, and W. Göpel, “Polymer-based sensor arrays and multicomponent analysis for the detection of hazardous organic vapors in the environment,” *Sens. Actuators B*, vol. 26, no. 1–3, pp. 126–134, 1995.
- [7] H. Sundgren, F. Winquist, I. Lukkari, and I. Lundstrom, “Artificial neural networks and gas sensor arrays: Quantification of individual components in a gas mixture,” *Meas. Sci. Technol.*, vol. 2, no. 5, pp. 464–469, 1991.
- [8] K. Ikohura and J. Watson, *The Stannic Oxide Gas Sensor*. Boca Raton, FL: CRC, 1994.
- [9] G. Horner and C. Hierold, “Gas analysis by partial model building,” *Sens. Actuators B*, vol. 2, pp. 173–184, 1990.
- [10] J. W. Gardner, “Detection of vapors and odours from a multisensor array using pattern recognition. Part 1. Principal components and cluster analysis,” *Sens. Actuators B*, vol. 4, pp. 109–115, 1991.
- [11] K. C. Persaud, S. M. Khaffaf, J. S. Payne, A. M. Pisanelli, D.-H. Lee, and H.-G. Byun, “Sensor array techniques for mimicking the mammalian olfactory system,” *Sens. Actuators B*, vol. 36, no. 1–3, pp. 267–273, 1996.
- [12] T. Eklov, P. Martensson, and I. Lundstrom, “Enhanced selectivity of MOSFET gas sensors by systematical analysis of transient parameters,” *Anal. Chim. Acta*, vol. 353, pp. 291–300, 1997.
- [13] R. Gutierrez-Osuna, H. T. Nagle, and S. Schiffman, “Transient response analysis of an electronic nose using multi-exponential models,” *Sens. Actuators B*, vol. 61, no. 1–3, pp. 170–182, 1999.
- [14] T. Nakamoto, A. Iguchi, and T. Moriizumi, “Vapor supply method in odor sensing system and analysis of transient sensor responses,” *Sens. Actuators B*, vol. 71, no. 3, pp. 155–160, 2000.
- [15] T. C. Pearce and J. W. Gardner, “Predicting organoleptic scores of sub-ppm flavor notes. Part 2. Computational analysis and results,” *Analyst*, vol. 123, pp. 2057–2066, 1998.
- [16] M. Holmberg, “Drift compensation,” in *NOSE 2nd Int. Workshop*, Lloret de Mar, Spain, Oct. 6–7, 2000.
- [17] J. E. Haugen, O. Tomic, and K. Kvaal, “A calibration method for handling the temporal drift of solid-state gas sensors,” *Anal. Chim. Acta*, vol. 407, pp. 23–39, 2000.
- [18] M. Fryder, M. Holmberg, F. Winquist, and I. Lundstrom, “A calibration technique for an electronic nose,” in *Eurosens. IX*, 1995, pp. 683–686.
- [19] R. Goodacre and D. B. Kell, “Correction of mass spectral drift using artificial neural networks,” *Anal. Chem.*, vol. 68, no. 2, pp. 271–280, 1996.

- [20] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg, "Drift correction for gas sensors using multivariate methods," *J. Chemom.*, vol. 14, no. 5–6, pp. 711–723, 2000.
- [21] R. Gutierrez-Osuna, "Drift reduction for metal-oxide sensor arrays using canonical correlation regression and partial least squares," in *Electronic Noses and Olfaction 2000*, J. W. Gardner and K. C. Persaud, Eds. Bristol, U.K.: IOP, 2000.
- [22] M. Roth, R. Hartinger, R. Faul, and H.-E. Endres, "Drift reduction of organic coated gas-sensors by temperature modulation," *Sens. Actuators B*, vol. 36, no. 1–3, pp. 358–362, 1996.
- [23] R. Aigner, P. Auerbach, P. Huber, R. Müller, and G. Scheller, "Sinusoidal temperature modulation of the Si-Planar-Pellistor," *Sens. Actuators B*, vol. 18–19, pp. 143–147, 1994.
- [24] C. Di Natale, F. Davide, and A. D'Amico, "A self-organizing system for pattern classification: Time varying statistics and sensor drift effects," *Sens. Actuators B*, vol. 26–27, no. 1–3, pp. 237–241, 1995.
- [25] S. Marco, A. Pardo, A. Ortega, and J. Samitier, "Gas identification with tin oxide sensor arrays and self organizing maps: Adaptive correction of sensor drifts," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, 1997, pp. 904–907.
- [26] M. Holmberg, F. Winquist, I. Lundström, F. Davide, C. Di Natale, and A. D'Amico, "Drift counteraction for an electronic nose," *Sens. Actuators B*, vol. 36, no. 1–3, pp. 528–535, 1996.
- [27] M. Holmberg, F. Davide, C. Di Natale, A. D'Amico, F. Winquist, and I. Lundström, "Drift counteraction in odour recognition applications: Life-long calibration method," *Sens. Actuators B*, vol. 42, no. 3, pp. 185–194, 1997.
- [28] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic, 1991.
- [29] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.
- [30] J. W. Sammon Jr., "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput. C*, vol. 18, pp. 401–409, May 1969.
- [31] H. G. Byun, K. C. Persaud, S. M. Khaffaf, P. J. Hobbs, and T. H. Misselbrook, "Application of unsupervised clustering methods to the assessment of malodour in agriculture using an array of conducting polymer odour sensors," *Comput. Electron. Agric.*, vol. 17, no. 2, pp. 233–247, 1997.
- [32] A. K. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 4–37, Jan. 2000.
- [33] R. Gutierrez-Osuna, "Signal Processing and Pattern Recognition for an Electronic Nose," Ph.D., North Carolina State Univ., 1998.
- [34] B. G. Kermani, "On using neural networks and genetic algorithms to optimize the performance of an electronic nose," Ph.D. dissertation, North Carolina State Univ., Raleigh, 1996.
- [35] P. A. Devijver and J. Kittler, *Pattern Recognition, A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [36] J. Doak, "An Evaluation of Feature Selection Methods and Their Application to Computer Security," Univ. California, Davis, Tech. Rep. CSE-92-18, 1992.
- [37] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. 26, pp. 917–922, Sept. 1977.
- [38] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, pp. 1119–1125, 1994.
- [39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [40] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. New York: Springer-Verlag, 1996.
- [41] G. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Conf. Machine Learning*, 1994, pp. 121–129.
- [42] M. A. Hall and L. A. Smith, "Feature subset selection: A correlation based filter approach," in *Proc. Intl. Conf. Neural Inform. Processing Intell. Inform. Syst.*, 1997, pp. 855–858.
- [43] T. Eklöv, P. Mårtensson, and I. Lundström, "Selection of variables for interpreting multivariate gas sensor data," *Anal. Chim. Acta*, vol. 381, no. 2–3, pp. 221–232, 1999.
- [44] P. Corcoran, P. Lowery, and J. Anglesea, "Optimal configuration of a thermally cycled gas sensor array with neural network pattern recognition," *Sens. Actuators B*, vol. 48, no. 1–3, pp. 448–455, 1998.
- [45] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. San Diego, CA: Academic, 1999.
- [46] P. Corcoran, J. Anglesea, and M. Elshaw, "The application of genetic algorithms to sensor parameter selection for multisensor array configuration," *Sens. Actuators A*, vol. 76, no. 1–3, pp. 57–66, 1999.
- [47] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.
- [48] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Statistics Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.
- [49] B. V. Dasarthy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Comp. Soc. Press, 1991.
- [50] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error backpropagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition Vol1: Foundations*. Cambridge, MA: MIT Press, 1986, pp. 318–362.
- [51] Y. Le Cun, L. Bottou, G. B. Orr, and K. K. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 1998, pp. 9–53.
- [52] S. Haykin, *Neural Networks, A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [53] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Math. Contr., Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.
- [54] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [55] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Inform. Processing Systems*. San Mateo, CA: Morgan Kaufman, 1990, vol. 2, pp. 524–532.
- [56] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Inform. Processing Systems*. San Mateo, CA: Morgan-Kaufman, 1990, vol. 2, pp. 598–605.
- [57] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Inform. Processing Systems*. San Mateo, CA: Morgan Kauffman, 1993, vol. 5, pp. 164–171.
- [58] W. S. Sarle, "Stopped training and other remedies for overfitting," in *Proc. 27th Symp. Interface Comput. Sci. Statistics*, 1995, pp. 352–360.
- [59] J. Sietsma and R. J. F. Dow, "Creating artificial neural networks that generalize," *Neural Networks*, vol. 4, no. 1, pp. 67–79, 1991.
- [60] A. R. Webb and D. Lowe, "The optimized internal representation of multilayer classifier network performs nonlinear discriminant analysis," *Neural Networks*, vol. 3, pp. 367–375, 1990.
- [61] J. B. Hampshire and B. A. Perlmutter, "Equivalence proofs for multilayer perceptron classifiers and the Bayesian discriminant function," in *Connectionist Models: Proc. 1990 Summer School*, D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, Eds., San Mateo, CA, 1990, pp. 159–172.
- [62] T. J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Comput.*, vol. 1, pp. 151–160, 1989.
- [63] D. R. Hush and B. G. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Mag.*, vol. 10, no. 1, pp. 8–39, 1993.
- [64] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
- [65] J. Schürmann, *Pattern Classification: A Unified View of Statistical and Neural Approaches*. New York: Wiley, 1996.
- [66] J. W. Gardner and P. N. Bartlett, *Electronic Noses. Principles and Applications*. New York: Oxford Univ. Press, 1999.
- [67] T. C. Pearce, "Computational parallels between the biological olfactory pathway and its analogue 'The electronic nose': Part II. Sensor-based machine olfaction," *Biosystems*, vol. 41, no. 2, pp. 69–90, 1997.
- [68] I. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–148, 1993.
- [69] M. Burl, B. Doleman, A. Schaffer, and N. S. Lewis, "Assessing the ability to predict human percepts of odor quality from the detector responses of a conducting polymer composite-based electronic nose," *Sens. Actuators B*, vol. 72, no. 2, pp. 149–159, 2001.
- [70] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1996.
- [71] S. Wold, "Soft modeling by latent variables: The nonlinear iterative partial least squares approach," in *Perspectives in Probability and Statistics, Papers in Honor of M. S. Bartlett*, J. Gani, Ed. London, U.K.: Academic, 1975, pp. 520–540.
- [72] P. Geladi and B. R. Kowalski, "Partial least-squares: A tutorial," *Anal. Chim. Acta*, vol. 185, pp. 1–17, 1986.
- [73] F. Lindgren, P. Geladi, and S. Wold, "The Kernel algorithm for PLS," *J. Chemom.*, vol. 7, no. 1, pp. 45–59, 1986.
- [74] R. E. Shaffer, S. L. Rose-Pehrsson, and A. McGill, "A comparison study of chemical sensor array pattern recognition algorithms," *Anal. Chim. Acta*, vol. 384, pp. 305–317, 1999.
- [75] G. Niebling, "Identification of gases with classical pattern-recognition methods and artificial neural networks," *Sens. Actuators B*, vol. 18–19, pp. 259–263, 1994.

- [76] J. R. Stetter, M. W. Findlay, K. M. Schroeder, C. Yue, and W. R. Penrose, "Quality classification of grain using a sensor array and pattern recognition," *Anal. Chim. Acta*, vol. 284, pp. 1–11, 1993.
- [77] J. W. Gardner, E. L. Hines, and H. C. Tang, "Detection of vapors and odours from a multisensor array using pattern-recognition techniques. Part 2. Artificial neural network," *Sens. Actuators B*, vol. 9, pp. 9–15, 1992.
- [78] D. Wilson, K. Dunman, T. Roppel, and R. Kalim, "Rank extraction in tin-oxide sensor arrays," *Sens. Actuators B*, vol. 62, no. 3, pp. 199–210, 2000.
- [79] P. Evans, K. C. Persaud, A. S. McNeish, R. W. Sneath, N. Hobson, and N. Magan, "Evaluation of a radial basis function neural network for the determination of wheat quality from electronic nose data," *Sens. Actuators B*, vol. 69, no. 3, pp. 348–358, 2000.
- [80] X. Wang, W. P. Carey, and S. S. Yee, "Monolithic thin-film metal-oxide gas-sensor arrays with application to monitoring of organic vapors," *Sens. Actuators B*, vol. 28, no. 1, pp. 63–70, 1995.
- [81] K. Domanský, D. L. Baldwin, J. W. Grate, T. B. Hall, J. Li, M. Josowicz, and J. Janata, "Development and calibration of field-effect transistor-based sensor array for measurement of hydrogen and ammonia gas mixtures in humid air," *Anal. Chem.*, vol. 70, no. 3, pp. 473–481, 1998.
- [82] W. P. Carey and S. S. Yee, "Calibration of nonlinear solid-state sensor arrays using multivariate regression techniques," *Sens. Actuators B*, vol. 9, pp. 113–122, 1992.
- [83] M. E. H. Amrani, R. M. Dowdeswell, P. A. Payne, and K. C. Persaud, "An intelligent gas sensing system," *Sens. Actuators B*, vol. 44, no. 1–3, pp. 512–516, 1997.
- [84] C. W. Therrien, *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. New York: Wiley, 1989.
- [85] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, pp. 59–69, 1982.
- [86] J. Zupan, M. Novic, X. Li, and J. Gasteiger, "Classification of multi-component analytical data of olive oils using different neural networks," *Anal. Chim. Acta*, vol. 292, pp. 219–234, 1994.
- [87] M. Holmberg, F. Winquist, I. Lundström, J. W. Gardner, and E. L. Hines, "Identification of paper quality using a hybrid electronic nose," *Sens. Actuators B*, vol. 27, no. 1–3, pp. 246–249, 1995.
- [88] T. Aishima, "Discrimination of liquor aromas by pattern recognition analysis of responses from a gas sensor array," *Anal. Chim. Acta*, vol. 243, pp. 293–300, 1991.
- [89] T. Albrecht, G. Matz, T. Hunte, and J. Hildemann, "An intelligent gas sensor system for the identification of hazardous airborne compounds using an array of semiconductor gas sensors and Kohonen feature map neural networks," in *Proc. 2nd Int. Conf. Intell. Syst. Eng.*, London, U.K., 1994, Conf. Pub. no. 395, pp. 130–137.
- [90] C. Di Natale, A. Macagnano, A. D'Amico, and F. Davide, "Electronic-nose modeling and data analysis using a self-organizing map," *Meas. Sci. Technol.*, vol. 8, no. 11, pp. 1236–1243, 1997.
- [91] E. L. Hines, J. W. Gardner, and C. E. R. Potter, "Olfactory feature maps from an electronic nose," *Meas. Contr.*, vol. 30, no. 9, pp. 262–268, 1997.
- [92] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman-Hall, 1993.
- [93] T. Masters, *Advanced Algorithms for Neural Networks*. New York: Wiley, 1995.
- [94] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, pp. 123–140, 1996.
- [95] Y. Freund and R. S. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1995.
- [96] T. C. Pearce and J. W. Gardner, "Predicting organoleptic scores of sub-ppm flavor notes. Part 1. Theoretical and experimental results," *Analyst*, vol. 123, pp. 2047–2055, 1998.
- [97] E. L. Hines, E. Llobet, and J. W. Gardner, "Electronic noses: A review of signal processing techniques," *Proc. Inst. Elect. Eng., Circuits Develop. Syst.*, vol. 146, no. 6, pp. 297–310, 1999.
- [98] L. A. Zadeh, "Fuzzy sets," *Inform. Contr.* 8, pp. 338–353, 1965.
- [99] P. Wide, F. Winquist, and D. Driankov, "An air-quality sensor system with fuzzy classification," *Meas. Sci. Technol.*, vol. 8, pp. 138–146, 1997.
- [100] G. Bargagna, B. Lazzarini, and A. C. Partridge, "Fuzzy logic classification of olive oils," in *Electronic Noses and Olfaction 2000*, J. W. Gardner and K. C. Persaud, Eds. Bristol, U.K.: IOP, 2000.
- [101] T. Sundic, A. Perera, S. Marco, A. Pardo, A. Ortega, and J. Samitier, "Fuzzy logic processing in combined carbon monoxide and methane domestic gas alarms," in *Electronic Noses and Olfaction 2000*, J. W. Gardner and K. C. Persaud, Eds. Bristol, U.K.: IOP, 2000.
- [102] B. Yea, R. Konishi, T. Osaki, and K. Sugahara, "The discrimination of many kinds of odour species using fuzzy reasoning and neural networks," *Sens. Actuators A*, vol. 45, pp. 159–165, 1994.
- [103] F. Marcelloni, "Recognition of olfactory signals based on supervised fuzzy C-means and k-NN algorithms," *Pattern Recognit. Lett.*, vol. 22, no. 9, pp. 1007–1019, 2001.
- [104] W. Ping and X. Jun, "A novel recognition method for electronic nose using artificial neural network and fuzzy recognition," *Sens. Actuators B*, vol. 37, no. 3, pp. 169–174, 1996.
- [105] D. Vlachos and J. Avaritsiotis, "Fuzzy neural networks for gas sensing," *Sens. Actuators B*, vol. 33, pp. 77–82, 1996.
- [106] B. Yea, T. Osaki, K. Sugarhara, and K. Ryosuke, "The discrimination and concentration-estimation of inflammable gases with tree structured neurofuzzy algorithm," in *Proc. Int. Joint Conf. Neural Networks*, 1998, pp. 663–668.
- [107] T. Nakamoto, S. Hanaki, and T. Moriizumi, "Artificial odor-recognition system using neural network for estimating sensory quantities of blended fragrance," *Sens. Actuators A*, vol. 57, no. 1, pp. 65–71, 1996.
- [108] S. Grossberg, "Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction and illusions," *Biol. Cybern.*, vol. 23, pp. 187–202, 1976.
- [109] J. W. Gardner, E. L. Hines, and C. Pang, "Detection of vapors and odours from a multisensor array using pattern recognition: Self-organizing adaptive resonance theory," *Meas. Contr.*, vol. 29, pp. 172–178, 1996.
- [110] E. Llobet, E. L. Hines, J. W. Gardner, P. N. Bartlett, and T. T. Mottram, "Fuzzy ARTMAP based electronic nose data analysis," *Sens. Actuators B*, vol. 6, no. 1–3, pp. 183–190, 1999.
- [111] D. S. Vlachos, D. K. Fragoulis, and J. Avaritsiotis, "An adaptive neural network topology for degradation compensation of thin film tin oxide gas sensors," *Sens. Actuators B*, vol. 45, no. 3, pp. 223–228, 1997.
- [112] T. C. Pearce, "Computational parallels between the biological olfactory pathway and its analogue 'The electronic nose': Part I. Biological olfaction," *Biosystems*, vol. 41, no. 2, pp. 43–67, 1997.
- [113] J. Ambros-Ingerson, R. Granger, and G. Lynch, "Simulation of paleocortex performs hierarchical clustering," *Science*, vol. 247, pp. 1344–1348.
- [114] W. Freeman, "Simulation of chaotic EEG patterns with a dynamic model of the olfactory system," *Biol. Cybern.*, vol. 56, pp. 139–150, 1990.
- [115] Z. Li and J. Hertz, "Odor recognition and segmentation by a model olfactory bulb and cortex," *Network: Comput. Neural Syst.*, vol. 11, pp. 83–102, 2000.
- [116] M. Wilson and J. Bower, "The simulation of large-scale neural networks," in *Methods in Neuronal Modeling: From Synapses to Networks*. Cambridge, MA: MIT Press, 1989, pp. 291–334.
- [117] J. White, T. A. Dickinson, D. R. Walt, and J. S. Kauer, "An olfactory neuronal network for vapor recognition in an artificial nose," *Biol. Cybern.*, vol. 78, pp. 245–251, 1998.
- [118] L. Rattou, T. Kunt, T. McAvoy, T. Fuja, R. Cavicchi, and S. Semancik, "A comparative study of signal processing techniques for clustering microsensor data (a first step toward an artificial nose)," *Sens. Actuators B*, vol. 41, no. 1–3, pp. 105–120, 1997.
- [119] T. C. Pearce, P. Verschure, J. White, and J. S. Kauer, "Stimulus encoding during the early stages of olfactory processing: A modeling study using an artificial olfactory system," *Neurocomputing*, vol. 38–40, pp. 299–306, 2001.
- [120] R. Gutierrez-Osuna, N. Powar, and P. Sun, "Chemosensory adaptation in an electronic nose," in *Proc. 2nd IEEE Bioinformatics Bioeng. Symp.*, 2001, pp. 223–229.



Ricardo Gutierrez-Osuna (M'00) received the B.S. degree in industrial/electronics engineering from the Polytechnic University of Madrid, Spain, in 1992, and the M.S. and Ph.D. degrees in computer engineering from North Carolina State University, Raleigh, in 1995 and 1998, respectively.

From 1998 to 2002, he served on the faculty at Wright State University, Dayton, OH. He is currently an Assistant Professor with the Department of Computer Science at Texas A&M University. His research interests include pattern recognition, machine learning, biological cybernetics, machine olfaction, speech-driven facial animation, computer vision, and mobile robotics.