Computational Methods for the Analysis of Chemical Sensor Array Data from Volatile Analytes

P. C. Jurs,* G. A. Bakken, and H. E. McClelland

Chemistry Department, Pennsylvania State University, 152 Davey Laboratory, University Park, Pennsylvania 16802

Received August 5, 1999

Contents

I. Introduction 2649
II. Sensor Array Types 2651
A. Metal Oxide Semiconductors (MeOX) (MeO_x) 2651
B. Metal Oxide Semiconducting Field Effect Transistors (MOSFETs) 2651
C. Quartz Crystal Microbalance (QCM) Devices 2651
D. Surface Acoustic Wave (SAW) Devices 2652
E. Conducting Polymers−Chemiresistors 2652
F. Optical Sensors 2652
III. Commercial Electronic Nose Instruments 2653
IV. Application Areas 2653
V. Data Preprocessing and Normalization 2653
VI. Methods of Data Analysis 2654
A. Linear Discriminant Analysis 2655
B. Applications of LDA 2656
C. Principal Component Analysis 2657
D. Applications of PCA 2659
E. Principal Component Regression 2662
F. Applications of PCR 2662
G. Partial Least Squares 2663
H. Applications of PLS 2663
I. Cluster Analysis 2664
1. Mapping and Display 2664
2. Hierarchical Clustering 2665
3. K−Nearest Neighbor Classification 2666
4. Visually-Empirical Region of Influence 2667
J. Computational Neural Networks 2667
1. Feed-Forward Neural Networks 2667
2. Applications of Feed-Forward Neural Networks 2669
3. Self-Organizing Maps 2673
4. Learning Vector Quantization 2673
5. Adaptive Resonance Theory-Based Methods 2674
6. Other Neural Network Techniques 2675
VII. Summary and Conclusions 2675
VIII. References 2676

I. Introduction

Chemical sensor arrays have come to have an important role in the analysis of volatile analytes. Such arrays gather data which have broad, overlapping sensitivity profiles, which require substantial data analysis, often involving pattern recognition methods to solve the problems being addressed. The present review is concerned with computational methods that have been used for analysis of the chemical sensor array data that is generated when these devices are exposed to volatile analytes. We have done electronic literature searches using relevant keywords and followed the trail of references to relevant papers. In addition, a number of primary journals (e.g., *Analytical Chemistry*, *Analytica Chimica Acta*, *Sensors and Actuators*, *Talanta*) have been scanned thoroughly for chemical sensor array papers and for leading references. We have focused on application papers for the time period 1994−1999; some work outside this time period has been included where it was essential to provide background for the data analysis techniques. Altogether, we have considered hundreds of papers and in coming to the collection cited we have left out only those publications which report routine work, are repetitious, are of little general impact, or provide no explanation of the computational methods used. The area of chemical sensor array applications has a substantial number of papers in which commercially available sensors and/or commercially available instrumentation is used to analyze volatile materials or headspaces without advancing the science of the sensor array computational analysis.

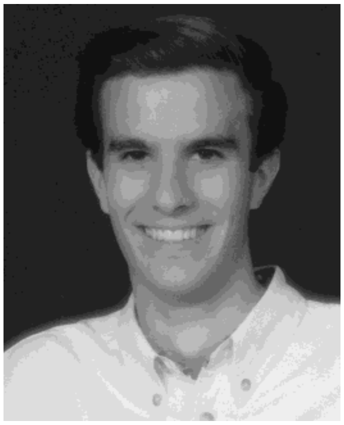
A number of excellent reviews covering differing aspects of chemical sensor arrays and related work with electronic noses have been published.[1−5] A recent review of chemical sensors[3] contained 921 references selected from the literature for the four-year period 1994−1998. A recent edited book[6] presents a detailed overview of biosensors and artificial noses by a number of authors. A very recent book[4] provides an excellent overview of the sensor technology, data analysis, and selected applications of electronic noses.

The importance of the chemical sensor array field is demonstrated by the existence of numerous organizations and symposia devoted to the area. Two Gordon Research Conferences—"Chemical Sensors and Interfacial Design" and "Bioanalytical Sensors"—meet regularly. The 6th World Congress on Biosensors meets in May 2000. The 8th International Conference on Chemical Sensors meets in July 2000. The Electrochemical Society has had a Sensors Division since 1988.

The review is organized as follows. First, we review sensor types with the aim of providing an overview

10.1021/cr9800964 CCC: $35.00
Published on Web 06/07/2000

Peter C. Jurs is Professor of Chemistry at The Pennsylvania State University. Born and raised in California, he received his B.S. degree in Chemistry from Stanford University and his Ph.D. degree in Chemistry from the University of Washington. He then joined the faculty of Penn State, where he has been Professor of Chemistry since 1978. He has been Assistant Head for Undergraduate Education and Acting Head of the Department. He was awarded the ACS Award for Computers in Chemistry in 1990. Jurs's research interests include the application of computer methods to chemical and biological problems. He has been actively involved in research in structure–property relationship studies for the prediction of physical and chemical properties of organic compounds and structure–activity relationship studies of biologically active compounds such as pharmaceuticals and genotoxic compounds. Recently he has been involved in a collaborative project aimed at developing an artificial nose using fiber-optic sensor arrays for detection of volatile organic analytes.



Gregory A. Bakken received his B.S. and M.S. degrees in Chemistry from Idaho State University in 1997. He is currently a Ph.D. candidate under the direction of Peter C. Jurs at The Pennsylvania State University. His current research interests include QSAR/QSPR and applications of pattern recognition methods to sensor array data.

in preparation for our discussions of computational methods and their applications. Among the computational methods, we focus first on preprocessing and normalization, which are the first methods ordinarily applied to raw sensor array data. We then systematically describe the major data analysis approaches used with chemical sensor arrays. For each method we present the basis of the approach, its suitability, applications that have appeared, and critiques of the applications. Linear discriminant analysis, principal components analysis, principal components regression, partial least squares, and several cluster analysis methods are covered. We then describe variations of computational neural networks that have been used with chemical sensor array data. We start with feed-forward neural networks, followed by their applications, and then describe self-organizing maps, learning vector quantization, adaptive resonance theory maps, and some additional approaches. The applications are accompanied by critiques comparing the application to the cautionary comments that preceded the discussion of each method. Finally, we conclude with some general comments meant to summarize the review.

Heidi Engelhardt McClelland received her B.A. degree in Chemistry from the University of Colorado in 1994. She is currently finishing her Ph.D. degree at The Pennsylvania State University under the direction of Peter C. Jurs. She is an avid reader, skier, and promoter of retired racing greyhound adoption.

As we introduce the methods used for analysis of chemical sensor array data, we provide some guidelines as to the applicability of each method and, in addition, cautionary comments on potential pitfalls. To analyze data from a chemical sensor array, some characteristics of the data are known in advance, such as the number of observations and the number of variables per observation. The analysis methods to be used and the ways in which they are used can be chosen with such knowledge being taken into account. However, some types of pitfalls or problems cannot be anticipated but are discovered later as data are being analyzed. Therefore, it is not possible to generate a standard procedure for the analysis of data from chemical sensor arrays. Rather it is necessary to have available a set of software tools for use where appropriate and to tailor the computational analysis to the problem at hand.

Sensor experiments, and the computational methods employed to support the experiments, can be for qualitative analysis or quantitative analysis. Qualitative analysis means seeking the presence or absence of a certain analyte or the presence of certain analytes exceeding a predetermined threshold. It can also refer to determining which or how many sensors to use. Quantitative analysis means measuring the concentrations, activities, or partial pressures of analytes or predicting parameters such as organoleptic scores or fermentation time. Applications in each of these areas are covered in the present review. In all such cases discussed here, the response pattern of a chemical sensor array is used in conjunction with computational methods to solve the analytical problem. Of the many steps that can form a complete analysis (sampling, sample transport, sample pretreatment, separation of the components, detection, data treatment, data interpretation), we focus here on the computational aspects of the analysis.

II. Sensor Array Types

A number of different physical or chemical phenomena can be exploited to generate cross-reactive sensor arrays. The basic idea is to create sensing elements of the array which exhibit distinct but overlapping broad selectivity profiles for the range of volatile analytes of interest. The sensor array used should generate a pattern of responses that are discernibly different for different samples. Thus, the individual sensors should respond to a broad range of compounds and should not be highly specific in their responses. The responses from the elements in an array to a specific organic vapor should be as independent as possible (in the statistical sense) to maximize the information being gathered for that compound. Reference 6 provides a table listing general guidelines regarding the performance of the different sensor techniques.

This section briefly describes each of the most common sensor devices that have been used in cross-reactive sensor arrays. The categories are metal oxide semiconductors, metal oxide semiconducting field effect transistors (MOSFETs), quartz crystal microbalance (QCM), surface acoustic wave (SAW), polymer-coated sensors, and optical sensors. A number of publications have provided overviews of the available sensor technologies, and they list these same categories with varying sequences and emphases.[1,3,4,6−9]

A. Metal Oxide Semiconductors (MeOX) (MeO_x)

Metal oxide semiconductor devices rely on changes in the conductance which are induced by adsorption of gases on the surface and by subsequent surface reactions. The device is made by depositing a thin, porous film of a metal oxide material onto an electrically heated ceramic pellet and then annealing at high temperature. These devices have been made from SnO_2, ZnO, In_2O_3, WO_3, Fe_2O_3, Ga_2O_3, and TiO_2, typically doped with a small amount of a catalytic metal such as platinum or palladium. Oxygen adsorbs onto the sensor surface, which removes electrons from the conduction band of the semiconductor material, which increases its resistance. Then the interaction of a reducing gas with the surface-adsorbed oxygen decreases this electron trapping which leads to characteristic increases in the electrical conductance of the sensor material. Elevated temperatures (100−600 °C) are used to ensure that the surface reactions are rapid and to decrease the chance that chemisorbed water will interfere. As the operating temperature range is elevated, heaters must be incorporated into the configuration. A very widely used device consists of a sintered tin oxide pellet on a heated ceramic base which also has electrical connections. Arrays of such simple devices have been used as cross-reactive sensor arrays. Metal oxide devices are susceptible to degradation if compounds bind irreversibly to the surface. An advantage of metal oxide sensors in many applications is their high sensitivity.

A commercially available device called the Taguchi gas sensor (TGS) is based on tin oxide sensors. It is comprised of a thin, porous SnO_2 film doped with precious metals. The doped tin oxide acts as an n-type semiconductor, and oxygen is chemisorbed onto the surface. Gas analytes then interact with the surface-absorbed oxygen, and the conductivity of the tin oxide film is changed. Typical operating temperatures are 300−400 °C. The temperature and the doping agent can be varied to alter the response characteristics. These sensors have been used for many analyses, and they are available in a variety of configurations for specific applications.

B. Metal Oxide Semiconducting Field Effect Transistors (MOSFETs)

A related type of sensor is the metal oxide semiconducting field effect transistor (MOSFET) device which operates at elevated temperatures. The MOSFET sensor consists of a doped semiconductor and an insulator (an oxide) covered with a catalytic metal. The output signal of the device derives from the change of potential in the sensor due to electric polarization when molecules react at the catalytic surface. These sensors operate at temperatures between 100 and 200 °C.

C. Quartz Crystal Microbalance (QCM) Devices

Quartz crystal microbalance (QCM) devices measure the mass of adsorbing molecules. These devices are known as bulk acoustic wave (BAW) devices and thickness-shear-mode sensors. The active element is a piezoelectric crystal with a fundamental resonant frequency. The basic principle is that the resonant frequency of a spring decreases when a mass is added to it

$$\Delta f = -2.3 \times 10^6 f_0^{\,2} (\Delta m/A)$$

where Δf is the change in frequency due to the absorbed mass (Hz), f_0 is the fundamental resonant frequency of the piezoelectric crystal (MHz), Δm is the mass of the substrate deposited on the surface (g), and A is the area of the sensor (cm^2). Thus, there is a predicted linear shift in measured frequency and mass loading of the device.

A quartz crystal oscillator is coated with a thin film of chemically selective absorbent material that forms a sensing membrane. Materials used include thin polymer films, monolayer films, and surface-attached molecules. When molecules are absorbed by the membrane, the resonant frequency decreases due to the increased mass. The frequency shift can be used as the output from the device. The response of the device can be altered by employing different membrane species. A discussion focusing on this aspect of sensor design can be found in refs 9 and 10.

The time constants for QCM devices have been reported to be on the order of tens of seconds or longer but are dependent on film thickness and can be less. Other characteristics include high sensitivity, linearity of response as a function of analyte concentration, high reproducibility, high stability over time, small in size, room-temperature operation, and low power consumption. Several commercial devices use QCM sensors.

D. Surface Acoustic Wave (SAW) Devices

Surface acoustic wave (SAW) devices are fabricated by forming interdigitated electrodes on a piezoelectric substrate such as quartz. A thin film coating of a selective absorbing material is deposited on the surface. A radio frequency voltage is applied, which produces a Rayleigh surface acoustic wave. The absorption of molecules onto the surface increases its mass which perturbs the wave which changes the frequency of the wave. The governing equation is the same as that for the QCM for acoustically thin solid films. The sensor is usually set up with a paired reference device, and the difference in frequency shift is measured. Temperature control is important to reduce drift. Just as with quartz crystal microbalances, the response of the device can be altered by employing different membrane species. A discussion focusing on this aspect of sensor design can be found in ref 10. Coatings used include polymers, lipids, Langmuir−Blodgett films, and self-assembled monolayers. SAW devices are operated at frequencies of several hundred megahertz, and they have very small areas, both of which lead to higher sensitivity and faster response times for SAW devices compared to QCM devices.

SAW devices have several additional advantages as well. They can be mass produced at low cost but with reproducible characteristics, and they can be miniaturized by using photolithography methodology for construction. Limitations include their sensitivity to humidity, which is shared by many other sensors.

One specific example of the use of a SAW device can be found in ref 11. This study used 10 SAW sensors which were coated with 10 different polymers. Relative frequencies were used with the reference being the frequency in the absence of a vapor. Patterns were generated from this array of 10 sensors to identify vapors using principal components analysis and clustering methods.

E. Conducting Polymers−Chemiresistors

Gas sensors based on measurements of the resistance changes in a thin film polymer have been extensively used. Polypyrrole is the material most widely used in these devices, and polyaniline has also been studied. These sensors are made by electropolymerizing a thin polymer film across a narrow electrode gap. The polymers have a conjugated π-electron system extending along the polymer backbone. Sorption of vapors changes interchain hopping, main-chain conduction, and counterion electrostatics, any or all of which can change conductivity. Polymers based on several classes of monomeric compounds have been used in these devices: pyrroles, anilines, thiophenes, indoles, furans. The usual property being measured is the change in conductivity of the polymer when volatile molecules interact with the polymer. Altering the structure or the functional groups incorporated into the polymer, and using differing doping ions, leads to changes in selectivity and sensitivity. Pyrrole monomers couple in the α position (adjacent to the N) to form the conjugated backbone. Changing the substituent on the β position leads to changes in the polymer characteristics. Changes in polymer characteristics can also be achieved by varying the electrochemical polymerization conditions.

Conducting polymer devices can operate at ambient temperature, simplifying their construction and lowering their power consumption. The response times of conducting polymer devices are on the order of 10 s. The responses are usually reversible at room temperature. Advantages include small size, operation at room temperature, and sensitivity to polar analytes. Limitations include reproducibility of fabrication of sensors, sensitivity to humidity, drift of baseline conductivity over time, lack of reproducibility of the response, susceptibility to poisoning, and sensitivity to light. Devices based on conducting polymers films have been used by AromaScan to generate a commercial analytical instrument.

An alternative polymer-coated chemiresistor array device has been developed by Lewis and co-workers based on a carbon black−organic polymer composite.[12] A conducting material, carbon black, is incorporated into various polymers, and then each polymer composite is painted across the foils of an ordinary ceramic capacitor. When exposed to an organic vapor, each polymer undergoes swelling. This moves the conducting particles apart, which increases the resistance across the capacitor. When the sensors are exposed to an analyte vapor, it takes on the order of seconds for the sensor to respond. The signal is taken as the relative change in the resistance compared to the resistance with an air stream blank sample. Polymer blends were used in a later study.[13] Another paper reported the use of chiral polymer composites which allowed the determination of the enantiomers of chiral gaseous samples.[14] Cyrano Sciences has commercialized this approach.

F. Optical Sensors

Optical gas sensors have also been used in artificial noses. Here, a light source excites either the gas being sensed or a reporter molecule and the signal measured is the resulting absorbance, reflectance, fluorescence, or chemiluminescence.

One example of an optical sensor[15] involves the use of beads of polymer on the end of optical fibers. In each polymer bead is an immobilized solvatochromic fluorescent dye. The dye is chosen to be sensitive to the polarity of its local environment. The dye is excited with a laser excitation source, and its fluorescent emission is monitored. When the polymer is exposed to volatile organic compounds, it swells and changes its polarity, and these changes result in a wavelength shift in the fluorescent emission. This wavelength shift is monitored as a function of time. Arrays of fiber optics with different polymers on each provide an array of sensors that can sense an analyte simultaneously. A shortcoming of such optical sensors is the susceptibility of the imbedded dye to photobleaching.

Table 1. Description of Some Preprocessing and Normalization Techniques[a]

Relative scaling	$\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij}}{\max(\mathbf{x}_i)}$ $\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij}}{\max(\mathbf{x}_r)}$, where $\mathbf{x}_r$ is a $1 \times p$ reference response $\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij}}{\|\mathbf{x}_i\|}$, where $\|\mathbf{x}_i\|$ denotes the Euclidean norm of $\mathbf{x}_i$
Background subtraction	$\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_b$, where $\mathbf{x}_b$ is the $1 \times p$ blank response
Signal average	$\mathbf{X}_{ij} = \frac{1}{\mathrm{N}}\sum_{k=1}^{\mathrm{N}} \mathbf{X}_{ij}^{k}$, where k replicates are available
Linearization	$\mathbf{X}_{ij} = \sqrt{\mathbf{X}_{ij}}$ $\mathbf{X}_{ij} = \log(\mathbf{X}_{ij})$
Mean-center	$\mathbf{X}_{ij} = \mathbf{X}_{ij} - \bar{\mathbf{x}}_j$
Autoscale	$\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij} - \bar{\mathbf{x}}_j}{\sigma_{\mathbf{x}_j}}$
Range scale	$\mathbf{X}_{ij} = \frac{\mathbf{X}_{ij} - \min(\mathbf{x}_j)}{\max(\mathbf{x}_j) - \min(\mathbf{x}_j)}$
Baseline subtraction	$\mathbf{X}_{ij} = \mathbf{X}_{ij} - \mathbf{X}_{i1}$

[a] X is an $n \times p$ calibration matrix (n samples, p sensors). X_{ij} denotes the response for the ith sample at the jth sensor, x_i is the $1 \times p$ response vector for sample i, and x_j contains response for all i samples at sensor j.

III. Commercial Electronic Nose Instruments

Tables showing the variety of commercially available electronic nose instruments have been published.[2,4,7,16] Several publications assert[16,17] that the leading commercial instruments are Aromascanner (AromaScan), e-NOSE (Neotronics Scientific), and Fox Intelligent Nose (Alpha MOS).

IV. Application Areas

The detection and measurement of chemical compounds in the gas phase is clearly important in most areas of modern life including industrial process control, health and safety monitoring, and environmental monitoring. Analytical applications of sensor arrays were developed before olfactory applications were investigated; we focus on artificial nose methodology here as a particularly apt application for cross-reactive sensor arrays. Industries using artificial nose technology include fragrance, cosmetics, flavor, food, drink, and brewing where monitoring of odor or freshness or the detection of taints is important. Other areas of application range widely and include the following: security screening; environmental protection, detection of excess volatiles in the workplace or housing; monitoring of effluents by plants; diagnosis of medical conditions through monitoring of wound infection or diagnosis of disease from exhalations; bioprocess monitoring; checking of raw material such as quality, taints, off-flavors; process monitoring such as odor quality during processing.

V. Data Preprocessing and Normalization

Data preprocessing can be used to systematically alter the raw signals from a sensor array in hopes that the altered signal will provide more useful input to the mathematical tool selected for data analysis (e.g., principal components analysis or neural networks). Unfortunately, no general guidelines exist to determine the appropriate data preprocessing technique given a particular type of sensor array. Often the appropriate preprocessing technique is not known. In such cases, it may be beneficial to explore several preprocessing strategies to determine which is best suited for a particular sensor array/data analysis method. A brief summary of some of the available data preprocessing strategies is presented below, along with some representative applications employing a variation of the technique.

Data preprocessing methods can be applied to both static and dynamic sensor responses.[18] Table 1 provides an overview of some of the available preprocessing techniques. Further discussion on preprocessing techniques can be found in refs 4 and 18−21. The first entry in Table 1 shows some of the available relative scaling procedures. The scaling can be done relative to a reference response or some aspect of the sample response. Such relative scalings can be used to try and eliminate the concentration dependence of the response intensity for each sensor. Therefore, this approach would be more desirable for qualitative applications. This preprocessing method is often encountered with electronic noses recording static responses.[12,22−25]

The subtraction technique is simply a background correction method. To reduce matrix effects, the response of a blank sample can be recorded and subtracted from each sample response. Another straightforward preprocessing method is signal averaging. This technique requires replicate measurements with each sensor. This can be accomplished by employing multiple sensors of each sensor type in an array, or by taking replicate measurements of each sample. The signal-to-noise ratio of the sample response can be improved by $N^{1/2}$, where N is the number of replicate measurements. Improved signals can be obtained in this manner.[26,27]

Linearization techniques seek to take a nonlinear response and transform it into a linear representation. This is desirable when linear data analysis methods are employed. However, it is often difficult to identify the nature of the nonlinearity of the sensor response. A general preprocessing method has been developed to allow data from nonlinear sensor responses to be analyzed with linear techniques.[28]

The method of mean-centering simply transforms the data such that the center of the variables coincides with the origin. A similar preprocessing method, autoscaling, involves mean-centering the data and dividing by the standard deviation of all sample responses at a particular sensor. Autoscaling is often used when measured responses are on different unit scales. The autoscaled data will have a mean of zero and unit variance for each sensor. Range scaling transforms all response values to lie between 0.0 and 1.0. That is, in the transformed domain, the minimum response at each sensor is at the origin and the maximum response is at 1.0. For an example of range scaling, see ref 21.

Some preprocessing methods are designed to handle dynamic data. For example, a baseline subtraction method can be used to eliminate signal recorded when no sample is present.[29] This is accomplished for a response by subtracting the first time point at a sensor from all the time points recorded at that sensor. This requires that the first time point of the response be recorded prior to exposure to sample. Instead of relying on a single time point, an average over several time points can be used to determine the amount to subtract provided all time points used in the average are recorded prior to exposure to sample.

A number of applications involve the measurement of data from sensors over time. This results in a large number of measurements per sensor. Typically, the number of data points must be reduced in some way to make the data matrix a reasonable size for pattern recognition methods. In the simplest case, the steady-state response is simply calculated, yielding one value per sensor. Several more complex methods for dealing with dynamic data responses have been used in various applications.

Descriptors may be calculated from the raw data.[30−32] Sutter and co-workers calculated descriptors such as the average responses in six time regions and the steepest positive and negative slopes. Responses recorded over time also allow derivative signals to be calculated.[29] Derivative signals can be useful for determining the rate or direction of change in a response over time. Other preprocessing techniques which have been used for dynamic data include wavelet transforms,[33] Fourier transforms,[33,34] Gram-Schmidt orthogonalization,[33] and the autoregressive model.[35]

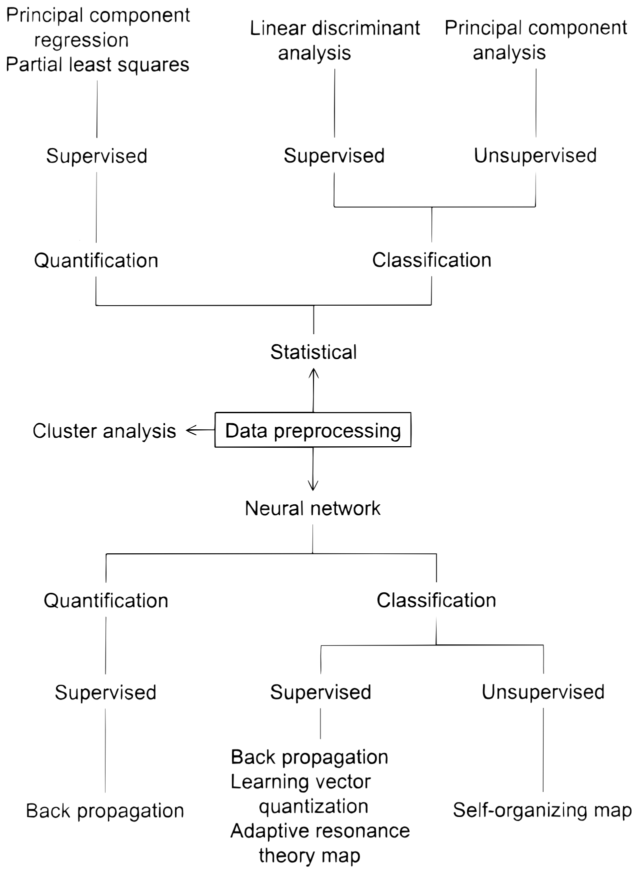


Figure 1. Properties of some of the available methods of analysis for data from sensor arrays.

VI. Methods of Data Analysis

There are many tools available for the analysis of data from an array of chemical sensors. As detailed in the previous section, the raw sensor responses are often preprocessed, and the preprocessed data are then used in a multivariate analysis technique. Figure 1 illustrates a means of categorizing some of the available methods as either statistical techniques or neural network-based approaches. Further delineations are based on whether the technique is used for quantification or classification. Additional groupings are defined by the data required for the technique. Those requiring only independent variable information (i.e., sensor responses) are termed unsupervised methods, while those that also use dependent variable information (e.g., analyte classes) are termed supervised methods.

The techniques listed in Figure 1 represent methods that will be discussed in this review. Often, for a given array application, the data will be analyzed by only one of the available methods. In some cases, one statistical and one neural network-based approach will be used to compare their effectiveness. Two recent papers by Shaffer and co-workers[36,37] provide comparisons of several of the available data analysis methods. One paper compares data analysis methods on multiple sets of data.[37] The second paper demonstrates how prediction accuracy can be improved using time-dependent information about the responses.[36]

Data preprocessing appears in the center of the diagram, because this is typically done regardless of the data analysis method chosen. Principal component analysis, shown on the top right of the diagram, is probably the most frequently used technique. Following the calculation of principal components, plots can be made to visually examine the data. The results of the principal components analysis can also be used as input to any of the other techniques.

The methods of data analysis depicted in Figure 1 represent a tool box, where the choice of a suitable method depends on the application. Sometimes there are several suitable tools to choose among, and then the choice depends on more specific details of the application and operator preference. Unsupervised methods are best for qualitative applications such as exploring relationships in the data. Supervised methods are used for quantitative applications, such as determining which class a particular observation belongs to.

The choice of an appropriate data analysis method is highly dependent on the nature of the data and the particular application. There is no universal method which will be right in all situations. The steps in a typical analysis, however, are usually similar regardless of the method chosen. First, a data set of sensor array responses for the analytes of interest is assembled. Second, the data are preprocessed appropriately for the application and the chosen method of data analysis. Third, the features which are to be used are chosen. Frequently, the steady-state response from each sensor is used as input for pattern recognition techniques. Other frequently used options include choosing a subset of sensors to use or doing a principal components analysis, then using the components as input for further analysis. Using an evolutionary algorithm[38,39] to select among descriptors[31] calculated from the sensor responses is another option. Fourth, analysis is performed using the chosen method.

A fifth step which is not always but should always be performed in classification and quantitation applications is validation of the models formed with a training set of data using data not used in creating the models. Another procedure for model validation consists of scrambling the dependent variable and testing the models using the scrambled dependent variable. If the results are good, this is an indication that there was a flaw in the data analysis. If the results are *not* good, this is further support that the analysis is valid. Leave-*n*-out validation is a third means frequently used to ensure models are not only applicable to the training set data used to create them.

The following sections contain descriptions of data analysis techniques used with sensor array data. Some cautions and potential pitfalls of the techniques are discussed. After each method of data analysis has been described, applications are presented describing the use of the technique with sensor array data.

A. Linear Discriminant Analysis

Linear discriminant analysis (LDA) can be used to separate classes of objects or assign new objects to appropriate classes.[40,41] The discriminants are linear combinations of the measured variables, e.g., sensor responses. Discriminant functions are calculated with the objective of maximizing the distance between classes relative to the variation within classes.

To begin, consider two classes, A and B, containing n_A and n_B samples each monitored at p sensors. The data can be collected in matrices $\mathbf{X}_A$ ($p \times n_A$) and $\mathbf{X}_B$ ($p \times n_B$). From these data matrices, sample mean vectors are computed as

$$\bar{\mathbf{x}}_A = \frac{1}{n_A}\sum_{i=1}^{n_A}\mathbf{x}_{Ai} \quad \bar{\mathbf{x}}_B = \frac{1}{n_B}\sum_{i=1}^{n_B}\mathbf{x}_{Bi} \tag{1}$$

where $\mathbf{x}_{Ai}$ and $\mathbf{x}_{Bi}$ denote the ith columns of $\mathbf{X}_A$ and $\mathbf{X}_B$, respectively. From this, sample covariance matrices can be calculated as

$$\mathbf{S}_A = \frac{1}{n_A - 1}\sum_{i=1}^{n_A}(\mathbf{x}_{Ai} - \bar{\mathbf{x}}_A)(\mathbf{x}_{Ai} - \bar{\mathbf{x}}_A)^T$$

$$\mathbf{S}_B = \frac{1}{n_B - 1}\sum_{i=1}^{n_B}(\mathbf{x}_{Bi} - \bar{\mathbf{x}}_B)(\mathbf{x}_{Bi} - \bar{\mathbf{x}}_B)^T \tag{2}$$

Assuming equal covariance matrices for the true populations represented by $\mathbf{X}_A$ and $\mathbf{X}_B$,[40,41] the covariance matrices can be combined as

$$\mathbf{S}_p = \frac{(n_A - 1)\mathbf{S}_A + (n_B - 1)\mathbf{S}_B}{(n_A + n_B - 2)} \tag{3}$$

to form a single estimate of the common covariance matrix for the sample populations.

With this information, discriminant functions can be calculated as

$$y = (\bar{\mathbf{x}}_A - \bar{\mathbf{x}}_B)^T\mathbf{S}_p^{-1}\mathbf{x} \tag{4}$$

where superscript −1 denotes the inverse of $\mathbf{S}_p$. For the inverse of $\mathbf{S}_p$ to exist, we must have $(n_A + n_B - 2) > p$. The discriminant functions can be used to visually examine clustering in the calibration data. Additionally, a discriminant classifier can be computed and used to assign an unknown sample into one of the two available classes. The discriminant classifier is computed as

$$m = \frac{1}{2}(\bar{\mathbf{x}}_A - \bar{\mathbf{x}}_B)^T\mathbf{S}_p^{-1}(\bar{\mathbf{x}}_A + \bar{\mathbf{x}}_B) \tag{5}$$

Using the response for an unknown compound, $\mathbf{x}_o$, the discriminant value for the unknown can be computed

$$y_o = (\bar{\mathbf{x}}_A - \bar{\mathbf{x}}_B)^T\mathbf{S}_p^{-1}\mathbf{x}_o \tag{6}$$

If $y_o \geq m$, $\mathbf{x}_o$ is assigned to class A, and if $y_o < m$, $\mathbf{x}_o$ is assigned to class B.

LDA can also be used in cases where more than two classes are present. For the case of k populations, compute $\bar{\mathbf{x}}_i$ and $\mathbf{S}_i$ for $i = 1$ to k according to eqs 1 and 2. With this, calculate

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{k} n_i \bar{\mathbf{x}}_i}{\sum_{i=1}^{k} n_i} \tag{7}$$

where $\bar{\mathbf{x}}$ is the average taken over all calibration samples. The combined covariance matrix can be computed as

$$\mathbf{S}_{\mathrm{p}} = \frac{\sum_{i=1}^{k}[(n_i - 1)\mathbf{S}_i]}{[\sum_{i=1}^{k} n_i] - k} \tag{8}$$

A measure of within groups variation for the calibration data can be obtained as

$$\mathbf{W} = [(\sum_{i=1}^{k} n_i) - k]\mathbf{S}_{\mathrm{p}} \tag{9}$$

Additionally, a measure of variation between groups for the calibration data can be calculated as

$$\mathbf{B} = \sum_{i=1}^{k}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^{\mathrm{T}} \tag{10}$$

The coefficients for the discriminant functions can be obtained by calculating the eigenvectors of $\mathbf{W}^{-1}\mathbf{B}$ and scaling them such that $\mathbf{e}_j^{\mathrm{T}}\mathbf{S}_{\mathrm{p}}\mathbf{e}_j = 1$, where $\mathbf{e}_j$ denotes the jth eigenvector.[40] That is, for the jth discriminant function, the discriminant values for a calibration sample, $\mathbf{x}_i$, can be computed as $\mathbf{e}_j^{\mathrm{T}}\mathbf{x}_i$.

Assuming $p > (k - 1)$, then $k - 1$ discriminant functions are available. Typically, only s of these discriminant functions are used, where $s \leq (k - 1)$. For $s \leq 3$, the discriminant functions can be used for graphical display of the calibration data. A classification rule can be derived to assign an unknown sample to one of the k available classes. Assign $\mathbf{x}_{\mathrm{o}}$ to population l if, for all $l \neq i$

$$\sum_{j=1}^{s}[\mathbf{e}_j^{\mathrm{T}}(\mathbf{x}_{\mathrm{o}} - \bar{\mathbf{x}}_l)]^2 \leq \sum_{j=1}^{s}[\mathbf{e}_j^{\mathrm{T}}(\mathbf{x}_{\mathrm{o}} - \bar{\mathbf{x}}_i)]^2 \tag{11}$$

That is, find the center of the class closest to the unknown sample in the space defined by s discriminant functions and assign the unknown sample to that class.

LDA is best used in situations when the relationship between array response and sample class is believed to be linear. Since the discriminant functions are generated based on the means and covariance matrices of training samples, it is extremely important that the training samples be representative of the population of samples that may be encountered. Therefore, it is beneficial to have a relatively large number of samples from each class in the training data. Like all supervised methods, LDA may provide overly optimistic results if only samples used in generating the discriminants are considered. To avoid this, it is best to use an external prediction set (samples not used in generating the discriminants) to determine model accuracy.

B. Applications of LDA

LDA has been used successfully to formulate boundaries between components of different classes based on responses from sensor arrays. An array of 16 sensors was used with LDA to classify three perfume samples at two injection concentrations (six classes).[42] Predictive ability was verified using a leave-one-out approach and by randomly selecting 33% of the data to be held out of the training phase. In all cases, prediction set samples were classified correctly. Additionally, LDA could be used to classify samples according to perfume type, disregarding injection volume.

Maricou and co-workers analyzed a large set of 12 test analytes with an array of 12 MOS sensors.[23] Two discriminant functions were used to cluster the samples into three groups of functionally similar compounds. The number of samples in each class is relatively small (5−7). When all 63 samples were used to generate discriminants, only one misclassification resulted. To verify prediction accuracy, 34 samples were used to generate discriminants and 29 were held out of training. A classification rate of 93% correct was obtained for the 29 external prediction set samples, demonstrating the ability of the model to generalize.

Doleman and co-workers used an array of 14 carbon black−polymer composites as sensors.[43] With the array, 19 volatile organic compounds were analyzed. LDA was used to maximize a resolution measure for pairs of analytes. Performance of the carbon black−polymer sensors was compared to that of arrays of tin oxide sensors and conducting polymer sensors. The carbon black−polymer composite array was shown to produce superior pairwise resolution values in all cases. Array performance decreased when signals were normalized to remove concentration effects. No data were collected after determination of the discriminants to verify that the pairwise resolution would not decrease.

An array of seven piezoelectric quartz crystals was used to monitor the smoke generated from four burning materials: paper, wood, plastic, and grain.[44] LDA was used to generate discriminants to classify samples based on sensor responses. For the 52 samples used to calculate the discriminants, only one was misclassified. This work provides a good example of using external prediction set samples to verify predictive ability of calculated discriminants. Of 12 samples in the external prediction set, only one was misclassified.

In a flavor analysis application, an array of 24 conducting polymer sensors was used to analyze three spiked flavor samples and a control sample.[45] LDA allowed over 80% of the samples to be classified correctly. However, all samples were used in generating the discriminant functions so true predictive ability may be over-estimated.

The stage at which tomatoes are harvested can influence their taste. Tomatoes that are not fully ripe are often exposed to ethylene gas after harvest. Maul and co-workers used an array of 12 polymer sensors to monitor aroma profiles from tomatoes exposed to varying amounts of ethylene.[46] Two discriminant functions were used to classify tomatoes according to the number of days they had been exposed to ethylene (1, 3, or 5 days). In the same study, similar analyses were carried out using tomato homogenates. Although additional samples were not used to verify the predictive ability of the discriminants, GC profiles confirmed that the volatile components produced by the tomatoes changed according to the number of days of ethylene exposure.

Sensor arrays have been used together with LDA for the measurement of boar taint.[47,48] An array of five MOS sensors was used to measure responses for back fat samples from male pigs. LDA was used to classify the samples into one of two classes. Seventeen of 19 samples in an external prediction set were correctly classified with the developed discriminants. Anner-Frempong and co-workers used LDA to group samples into three classes according to concentration of androstenone and skatole.[48] The array of conducting polymer sensors was only able to classify 53% of the prediction set samples correctly. However, as pointed out by the authors, the concentrations of androstenone and skatole are not necessarily a direct indicator of taint status. Therefore, the same samples were labeled as normal, abnormal, or doubtful according to taint status as determined by sensory measurements.[48] Using a set of calibration samples, discriminant functions were developed that correctly classified 84% of the prediction set samples. The main confusion was the classification of normal samples as abnormal. The two applications involving classification of boar taint demonstrate effective use of an external prediction set to validate a model.

Keshri and co-workers used a Bloodhound (Bloodhound Sensors Ltd., Leeds, U.K.) electronic nose consisting of an array of 14 polymer sensors to monitor five types of fungi and a control group.[49] Using two discriminant functions, the fungi could be separated from the control group and from each other. In a similar study, an array of 12 polypyrrole conducting polymer sensors was used to monitor bioprocesses.[50] Different lots of growth media, some containing 'good' casein hydrolysate and some 'bad', were examined. Two discriminant functions were able to separate the 'good' and 'bad' lots. Samples not used in model formation were correctly classified using the developed discriminants, providing a good example of model verification with LDA. The same array was used to examine cultures of *M. carbonacea*. With three discriminant functions, uncontaminated batches could be separated from contaminated batches and the contaminated batches could be grouped according to the contaminating agent. Additionally, the growth phase of uncontaminated cultures could be monitored with LDA.

Fenner and co-workers did several studies with an array of 12 conducting polypyrrole sensors to analyze wastewater from sewage treatment plants.[26,27,51,52] LDA was used on data from the array to differentiate samples from three treatment plants.[26,52] Although clusters were present, there was a significant amount of overlap. The same array was used to classify samples from a particular treatment plant as raw wastewater, settled wastewater, or final effluent. Again, overlapping clusters were generated. Further analysis revealed that seasonal changes in the samples may have introduced variation in the data resulting in the overlapping of clusters.[26,52] The sensor array responses were also compared to data from sensory panels.[27,51] In some instances, the sensor array was able to produce results similar to those obtained by sensory panels. This series of papers provides a nice example of how LDA can be used to gain insight into the relationships in samples and how this information can be used to guide further study.

C. Principal Component Analysis

Sensor arrays can be used to generate a great deal of data in a very short time. A significant challenge exists in finding ways to extract information useful in solving the problem at hand from the data. Graphical analysis of the raw data is often not possible since the number of samples and sensors is typically greater than three. Therefore, methods reducing the data to dimensions that can be accommodated graphically are often used. Visual examination of sensor array data in reduced dimensions can provide useful information about both samples and sensors.

Principal component analysis (PCA) provides one efficient approach for reducing the dimensionality of a data set. Often two or three principal components provide an adequate representation of the data, which is convenient for graphical output. Typically, the calibration data is collected in a matrix, **X**, with m samples as rows and n sensors as columns. Before PCA, the data matrix **X** is often preprocessed according to one of the methods discussed in the data preprocessing section. For the discussion that follows, it will be assumed that the data in **X** have been preprocessed as desired and that the number of samples in **X** is greater than the number of sensors, i.e., $m > n$.

The underlying premise in PCA is that the raw data in **X** can be decomposed into eigenvectors and associated eigenvalues.[53] Of the several methods available to decompose **X**, one of the most common is through singular value decomposition (SVD).[53-55] This method is often preferred because SVD provides the most stable alternative under the widest range of conditions.[53]

Performing an SVD on **X** produces

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$$

$$= (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n)\begin{bmatrix} \sigma_{11} & 0 & . & . & . & 0 \\ 0 & \sigma_{22} & . & . & . & 0 \\ . & . & & . & . & \\ . & . & & . & & \\ . & . & & . & . & \\ 0 & 0 & . & . & . & \sigma_{nn} \end{bmatrix} (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)^{\mathrm{T}} \quad (12)$$

where **U** is an $m \times n$ orthonormal matrix, Σ is an n

× n matrix with singular values (σ_{ii}) on the diagonal and zeros off diagonal, and **V** is an $n \times n$ orthonormal matrix. The values σ_{ii} are nonnegative numbers, which are listed such that $\sigma_{11} \geq \sigma_{22} \geq ... \geq \sigma_{nn}$. The squares of the singular values are equal to the eigenvalues of **X**, i.e., $\sigma_{ii}^2 = \lambda_i$, where λ_i denotes the ith eigenvalue of $\mathbf{XX}^T$ and $\mathbf{X}^T\mathbf{X}$. The columns $\mathbf{u}_1...\mathbf{u}_n$ of **U** denote eigenvectors of $\mathbf{XX}^T$ associated with eigenvalues $(\sigma_{11})^2...(\sigma_{nn})^2$. The columns $\mathbf{v}_1...\mathbf{v}_n$ of **V** represent eigenvectors of $\mathbf{X}^T\mathbf{X}$ associated with eigenvalues $(\sigma_{11})^2...(\sigma_{nn})^2$. The eigenvectors forming **U** and **V** are termed principal components. Frequently, if SVD is not employed, the decomposition of **X** will produce only two matrices, **U** and **H**. In this case, the columns $\mathbf{h}_1...\mathbf{h}_n$ of **H** are equal to the columns $\mathbf{v}_1...\mathbf{v}_n$ of **V** scaled by the singular values $\sigma_{11}...\sigma_{nn}$, i.e., $\mathbf{h}_i = \sigma_{ii}\mathbf{v}_i$. In practice, the columns of **U** and **H** are usually of interest.

The amount of variance described by each eigenvector is determined by the magnitude of the associated eigenvalues

$$\mathrm{var}_i = \frac{\sigma_{ii}^2}{\sum_{k=1}^{n} \sigma_{kk}^2} \qquad (13)$$

where var$_i$ denotes the percent variance contributed by the ith eigenvalue. Since eigenvalues are sorted in descending order, the greatest amount of data variance will be described by the first PC. The second PC will describe the greatest amount of data variance in a direction orthogonal to the first PC. If sensor responses are collinear, which is often true for the arrays discussed in this paper, the first two PCs often account for 80% or more of the data variance, indicating that the data variance can be well described in only two dimensions. However, this does not imply that PCA will provide class separation. If sensors behave in a nonlinear or nonadditive fashion, class separation may not be possible using PCA. Therefore, it is best to use PCA when the sensors in an array are thought to behave linearly.

Once the decomposition of **X** has been completed, the columns of **U**, termed scores, can be used to project the raw data into a lower dimensional space for comparison of samples. For example, plotting $\mathbf{u}_1$ versus $\mathbf{u}_2$ allows the samples comprising **X** to be compared in the two dimensions accounting for the greatest amount of variance in the data. Such plots are termed score plots. Examination of these plots reveals any clustering of samples that may be present in the principal components space examined. Larger numbers of PCs can be included in the dimensionality reduction, but useful graphical output is only available for up to three PCs.

Figure 2 shows an example of a plot using two PCs. The data used to generate the figure are from unpublished work[56] using three sensor responses. Each sensor response is the signal-averaged response of 20 individual sensors, each comprised of a surface-functionalized microsphere. Microspheres are exposed to a solvatochromic dye, which provides the fluorescent signal. Time-dependent fluorescent intensity changes were recorded over 45 time points. For each sensor, nine descriptors were calculated as the average response over successive five time point intervals, for a total of 27 descriptors for each sample. Responses were recorded eight times each for five analytes, for a total of 40 responses. Of the five analytes, three were types of coffee and two were types of roses. As Figure 2 illustrates, two PCs were adequate to separate the coffee and roses. This same plot could be used to attempt to classify coffee samples into one of the three types, but this classification would be less certain since some overlap does exist among the coffee samples. Clearly, the two rose types are too similar to be differentiated with this analysis.

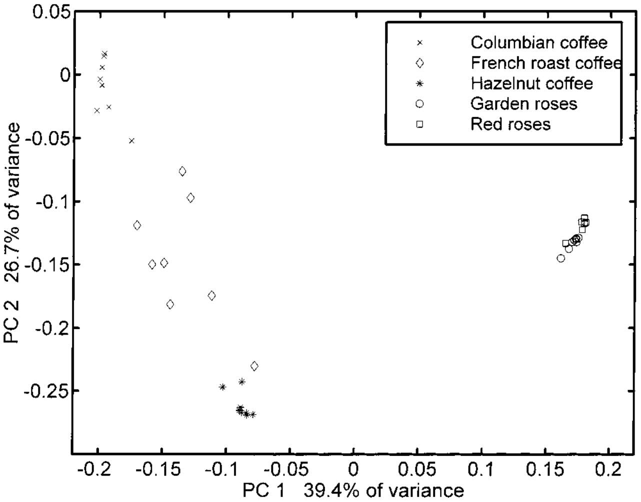


Figure 2. Plot of the first two principal components illustrating the separation between samples of coffee and roses, and the partial separation of the three types of coffee present. Note that the first two principal components account for 66.1% of the data variance.

Loading plots, obtained by plotting columns of **H** relative to one another, can be used to evaluate sensors. Each PC can be examined to determine which sensors contribute significantly to that PC. A high loading value indicates that the PC is aligned in a direction close to the original sensor response. Selection of important sensors (those contributing highly to the data variance) can be accomplished by examination of loading plots. Additionally, the relation of the sensors to one another can be explored. Loading plots can be used to determine which sensors are providing unique information and which sensors are providing similar information.

As shown in Figure 1, PCA is an unsupervised method of classification. PCA is generally applied to data with high linear dependence among the independent variables. Near-infrared spectroscopic data are a good example of this. Care should be taken when examining PC plots. The PCs describe data variation and may or may not provide class separation. It is best to apply PCA in situations where sensor response is thought to behave linearly relative to sample concentration. The popularity of PCA is related to the ease with which it can be implemented. Two- and three-dimensional plots can be examined to determine if clusters are present. However, the graphical nature of the technique results in some degree of subjectivity in interpreting results. Additionally, PCA is simply applied to the data in **X**, and PC plots are generated to see if any clustering is present. Samples that are not part of **X** could then be projected onto the PC plots to see if they fall in appropriate clusters, but this verification is typically not done.

D. Applications of PCA

Examination of score plots is frequently used to reveal clustering of samples. The first two PCs are often employed, since they capture the most data variance, and therefore theoretically best describe the data. Applications of PCA to data from chemical sensor arrays is used for a variety of things, from a complete analysis of the data to a guide for other methods of data analysis. Oftentimes, work with sensor arrays, especially electronic noses, requires only differentiation of patterns, e.g., coffee versus roses. It is not necessary to identify components in a sample, which may be important in other areas such as environmental monitoring. Some of the many applications involving PCA applied to sensor array data are presented below.

To evaluate sensor array performance, volatile organic compounds are often selected as test analytes. Freund and Lewis used an array of 14 conducting polymer sensors to analyze eight such test analytes.[57] Responses from single sensors were first examined, but no single sensor could distinguish all eight analytes. However, application of PCA to data from all 14 sensors resulted in separation of all analytes using information from the first four PCs in various three-dimensional plots. A similar experiment was conducted with an array of 17 carbon black−polymer resistors as sensors.[12] In this study, nine organic test analytes were used. Although some overlap was present using the first three PCs, complete separation could be achieved using PCs 3, 4, and 5. The same array was also used to examine the responses of binary mixtures of methanol and ethanol. For a given sample flow rate, a linear behavior was seen in the plot of the first two PCs as samples varied in composition from pure methanol to pure ethanol. The linear behavior of the sensors demonstrates that PCA was an appropriate choice of analysis method. This finding suggests that not only can the components of mixtures be identified using PCA, but also that the mole fraction of each component in the binary mixture can be determined.

Kraus and Gauglitz used an array of six sensors whose responses were measured using reflectometric interference spectroscopy.[58] Responses for three test analytes, as well as binary mixtures of the analytes, were recorded. A detailed analysis of the PCs is presented to demonstrate which analytes and sensors influence which PCs. This provides a good example of how analysis of PCs can be used to identify particular sensors that are responsible for the separation of particular analytes. Additionally, clusters were mathematically defined based on PCs to separate the test samples into appropriate groups. Once the clusters had been defined, responses of validation samples were recorded with the array. The location of these samples relative to the defined clusters allowed classification of the validation data.

Sensor arrays are often employed to monitor beverages. For example, headspace samples of drink mixes allowed differentiation of genuine and faulty batches.[8] Nanto and co-workers were successful in separating beverages based on responses obtained with an array of four quartz resonators.[59] Targeted problems included samples with varying amounts of ethanol and samples with different types of alcohols. In both cases, complete separation was achieved with two PCs. When an array of nonspecific sensors was utilized in beer flavor analysis, the within classes variation was large relative to the between class variation and PCA was unsuccessful in separating the flavors.[45] Di Natale and co-workers used an array of five MOS sensors to record responses for samples of wine.[60] Five vintages of wine were examined at three concentrations each. PCA was not able to separate the samples by vintage. When responses from one sensor were removed and PCA was performed on the data from the remaining four sensors, better separation of the wines was achieved. This demonstrates that PCA can be adversely affected by information that contributes to data variance but does not contribute information useful in differentiating samples.

In food analysis, 18 MOS sensors were used to record responses from the headspace of cheese samples.[61] Of six varieties of cheese, only two were found to overlap in a PC plot. Cheese age was also analyzed in this study to determine during what time the aroma of the cheese most significantly changed during the aging process. A discrimination index is also given for each PC analysis, but a definition of discrimination index is not presented making interpretation of the results difficult. In a similar way, the storage time of fish samples was monitored.[62] An array of eight amperometric sensors was used to obtain responses from fish samples over a period of days. A plot of the first two PCs described storage time coherently with a minor amount of disorder. Similarly, storage time for codfish was monitored with an array of eight quartz microbalances.[63] Samples at low storage time (1−3 days) formed distinct clusters using two PCs; however, samples that were older (4−6 days) were more overlapped.

Göpel and co-workers developed a concept they term modular sensor system (MOSES).[64,65] MOSES involves using sensors that respond by different transduction principles. All sensors used collectively form an array from which data is gathered for pattern recognition with PCA. In one application, an array composed of eight quartz microbalance sensors and eight metal oxide sensors was used with PCA to separate different types of packing material used in the food industry.[65] The same sensor array was also used to differentiate four types of olive oils. Several other samples have been categorized using a MOSES system and PCA, including coffee brands, textile material, and whiskey brands.[64] These studies provide good examples of how combinations of different sensor types can be used with PCA to obtain better separations than those with a single sensor type. It is important in such analyses to consider the linear (or nonlinear) behavior of all sensor types in the array.

In the analysis of odor from sausage fermentation, PCA was used to analyze results from a trained sensory panel and from an array of four TGS and 10 MOSFET sensors.[66] Both showed one sausage sample to be an outlier. Data from the sensor array was collected at 4, 29, and 52 h in addition to 3 days. The outlier batch could be identified clearly on the PC plot after only 4 h of fermentation, indicating its potential use in process monitoring. In a study of tomato paste aroma, data from a trained sensory panel was compared with electronic nose data.[67,68] The trained panel evaluated eight classes of tomato paste and rated them in nine categories. The electronic nose used an array of eight quartz microbalances to record data for the eight classes of tomato paste. PCA was performed on both the panel results and the nose results. In neither case were the eight classes separable, but there was somewhat less overlap for the electronic nose data. It should be noted that there was more array data than sensory panel data for each of the eight classes of tomato, which may help to explain the improved clustering in the PCA plot of the array data.

In a related application, an array of 10 MOSFET, four TGS, and one IR sensor was used in on-line monitoring of an ethanol batch cultivation with the yeast *Saccharomyces cerevisiae*.[69] Responses of all 15 sensors were used to generate plots of scores and loadings. On the basis of the analysis of these plots, a reduced set of five sensors was selected for further evaluation. The score plot for the reduced sensor set showed a correlation between variation in the PCs and the stage of the cultivation. Gibson and co-workers found that PC plots were not adequate to separate three strains of yeast, control media, and distilled water.[70] Responses were collected with an array of 16 conducting polymer sensors, but only the seven sensors showing the highest statistical difference in responses were considered.

Mandenius and co-workers used an array of 15 sensors (10 MOSFET, four TGS, one IR) to monitor the manufacturing of baker's yeast.[71] PCA plots allowed the time course of the cultivation to be followed. Features such as the ethanol maximum and carbon dioxide evolution optimum could be identified. The sensor data was supplemented with other on-line signals including aeration rate, reactor volume, and ethanol content. PCA of the augmented data allowed further analysis of the time course of the cultivation. This work provides a good example of how other information about a system can be combined with sensor array data before PCA to improve results.

Odor analysis plays a large role in the perfume industry. Human sensory panels are typically relied upon to differentiate scents. Carrasco and co-workers employed an array of 18 MOS gas sensors to differentiate perfumes.[42] Of the three perfume samples considered, two were of the same type, one of which had previously been labeled as 'bad' by a sensory panel, and the third sample was a separate perfume type. Each sample was injected at two volumes, producing a total of six classes. Two unresponsive sensors were removed, and PCA was carried out on the remaining 16 responses. Using three PCs, six clusters were formed with some overlap between clusters. A better separation, with virtually no overlap among the six clusters, was achieved using LDA. Additional experiments with PCA involved determining if samples for each perfume type could be separated based on injection volume. In all three cases, two PCs were able to separate the samples with slight overlap existing in only one case. Additionally, PCA showed that the 'bad' perfume batch could not be fully separated from the normal batch using three PCs.

A hypothesis testing method was devised by Marth and co-workers to determine if a sample contained response information from a species not present during array calibration.[72] PCA was performed on an augmented matrix, consisting of the calibration responses and the sample response. Using the number of PCs previously determined to account for relevant variation in the calibration data, the sources of known variation were projected out of the augmented matrix. PCA was then performed on the remainder of the augmented matrix, to determine the largest eigenvalue. Hypothesis testing consisted of comparing this value to the largest eigenvalue from a pure random matrix of the same size as the augmented matrix. If the values were determined to be different at some confidence level, the sample response was labeled as containing species not present in the calibration samples. This approach represents a unique use of PCA. Prediction samples can be classified as represented by or different from the calibration samples, but no further groupings of samples can be accomplished.

The perineal odor of cows was monitored with an array of 12 conducting polymer sensors.[73] PC plots were used to demonstrate that samples could be grouped as taken on the day of estrus or in the midluteal phase. Discrimination of eight brands of glue was accomplished using two PCs from data taken with an array of four plasticized conducting polymer sensors.[25] A mixed array of 15 sensors, composed of 10 MOSFET, four TGS, and one CO_2 sensor, was used to separate classes of paper.[74] This study found that the separations based on two PCs were more successful using data from a reduced sensor array, composed of seven or only four sensors, compared to data from all 15 sensors. However, the rationale for selecting subsets of sensors to use is not clear. Bodenhöfer and co-workers designed a sensor array to discriminate chiral analytes.[75] To accomplish this, the sensor array was designed with chiral polymers, enabling differentiation of mixtures of different enantiomeric ratios of analyte using PC plots.

Although PCA is a classification tool, it can sometimes be used to obtain some degree of quantitative information. Hong and co-workers used an array of four gas sensors to obtain responses to four gases each at three concentrations.[76] PCA revealed that the analytes clustered according to analyte identity and concentration. There was some overlap of samples at lower concentrations, indicating poor sensitivity in these regions. Additionally, no mention is made regarding the usefulness of the sensor array for separating the four types of gases if the concentrations covered a continuous range rather than three discrete values. The same array was used to record responses for six flavor samples (carrot, green onion, woman's perfume, man's perfume, 25% liquor, and 40% liquor). The six flavor samples contained only slight overlap based on two PCs. Both examples showed some ability to classify the same analyte based on a concentration difference. However, normalization of sample responses may remove any concentration effects from the response, allowing samples to be grouped according to analyte type, regardless of concentration.

Loading plots can be examined to evaluate sensor performance. In one study, the stability of films for QCM sensors was examined.[22] A starting set of coatings was first used, and PC plots were generated so the location of the sensors in PC space could be examined. Films for sensors that were not close together in the PC plot were considered in generating mixed films, in an effort to incorporate the useful properties of both the films into one sensor. In a similar study, arrays of eight QCM sensors, with different film coatings, were evaluated.[77] For each array, responses to test analytes were measured. PCA was used to reduce the data into two dimensions, and the ability of the array to cluster the samples was examined. An optimal array of coatings was selected and used to examine a series of chemically and physically similar alcohols. Although PCA could be used to separate the alcohols, LDA was shown to produce much tighter clusters. However, it is important to remember that LDA is a supervised method and will often produce tighter clusters in training data. To better assess the performance of LDA, samples not used in generating the discriminants should be examined.

Four organically modified silicate films were used by Inoue and co-workers to demonstrate that acidic and basic analytes could be differentiated by conductance changes in the films due to interaction with analyte vapor.[78] Tetraethoxysilane was used as a modifier in the silicate films to increase response diversity. Plots of two PCs showed that the silicate films were capable of responding to a wide range of acidic and basic organic molecules.

Barkó and Hlavay used PCA to determine if sensors were providing unique information.[79] An acceptable array of four piezoelectric sensors was then used to measure the responses of several test analytes. The analytes overlapped slightly when plotted using the first three PCs. The authors discuss one analyte at a time, providing possibilities for the degree of separation. They suggest that as the number of sensory elements in an array increases, the array can be used to distinguish more analytes. This could be true, provided that the additional sensing elements provide new information useful for separating the analytes of interest.

An array of SAW sensors was used to compare the responses with and without hydrogen bond acidic polymers as sensors.[80] Plots of the first two PCs revealed that low-polarity test vapors were well separated from more polar test vapors with an array of 11 sensors, four of which were hydrogen bond acidic polymers. Without these four sensors, the test analytes could no longer be separated based on polarity. This provides another good example of how PCA can be used to determine the array best suited for a given task.

Roussel and co-workers calculated features based on sensor array responses and used the features for further data analysis instead of the actual sensor responses.[29] They recorded time-dependent responses using an array of five tin oxide sensors. Instead of using the adsorption maxima for each sensor, 29 features were calculated from the time-dependent response and the first and second derivatives of the response. This approach is unique in that both feature extraction and PCA are used to compress the data into a lower dimension. Measures of repeatability and discriminant distance were developed. These measures were used on the extracted features, as well as on the projections of the extracted features into PC space. Calculation of repeatability in PC space showed that a subset of five features would be best. However, the discriminant distance was best using all 29 features and PCA. No comparison is made to calculations of PCA on raw data to more clearly demonstrate the effectiveness of feature extraction.

PCA can also be used to analyze data collected with respect to time. For example, Auge and co-workers used an array of four quartz resonators to collect data for four test analytes.[81] The analytes were presented to the array in sequence, with data collection at successive time increments. The data matrix generated was analyzed with PCA to determine if the analytes would cluster. Four clusters were observed, along with scattering points representing carrier gas and the rising and falling of sensor signals. Additionally, Llobet and co-workers compared clustering obtained based on steady-state and transient measures.[82] An array of four Taguchi gas sensors was used to analyze three organic volatiles at varying concentrations. Two response values, one at steady-state and one capturing temporal information, were recorded for each sensor during each sample presentation. Separate PC analyses were performed on the steady-state and transient data. Superior clustering was achieved using the temporal information. This is not surprising since the authors point out that the temporal measurements were more reproducible than the steady-state measurements.

Nakamurra and co-workers also developed a method of extracting temporal information from an array of six QCM sensor responses.[24,35] They described sensor responses with the autoregressive model, from which they extracted time constants for the response. The time constants were used with PCA and provided better separation of test analytes than the static measure of saturation mass.[35] The method was refined to accommodate situations in which the gas concentration was changing.[24] Smoke from different sources was examined in an effort to determine the source of the fire based on the smoke. Samples of burning PVC cables and circuit boards were significantly overlapped when PCA was performed on static responses. Incorporation of time constants allowed the PVC cables and circuit boards to be differentiated.

In an effort to work toward a method to standardize electronic noses, Gardner and Bartlett proposed methods to define instrument range and resolving power.[83] Mathematical expressions were derived to model response characteristic of several sensor types. Simulated array responses were used to examine the affects of random error in the system. PCA was used to demonstrate how the addition of random error affects sample clustering. At 5% random concentration errors, three analytes clustered well using the first two PCs. When the random concentration error was increased to 20%, analytes overlapped in plots of the first two PCs. However, the first and third PCs produced clusters that were separable, indicating that the increased random error significantly degraded the useful information in the second PC.

E. Principal Component Regression

Principal component regression (PCR) provides a link between the matrix **X** of response information and the matrix **C** of concentration information.[55] The matrix **C** is $m \times s$, where s denotes the number of analytes. The matrices are related by $\mathbf{C} = \mathbf{XP}$, where **P** is the $n \times s$ matrix of regression coefficients.[53-55]

The calibration information in **C** and **X** can be used to find an approximation to the regression coefficients. The true regression coefficients cannot be calculated exactly due to the unavoidable presence of experimental error. The first step in the computation of the regression coefficients is the decomposition of **X** according to eq 12. The approximation to **P**, denoted $\widehat{\mathbf{P}}$, can be computed as

$$\widehat{\mathbf{P}} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{\mathrm{T}}\mathbf{C}$$
$$= (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n)\begin{bmatrix} \sigma_{11} & 0 & . & . & . & 0 \\ 0 & \sigma_{22} & . & . & . & 0 \\ . & . & . & & & . \\ . & . & & . & & \\ . & . & & & . & . \\ 0 & 0 & . & . & . & \sigma_{nn} \end{bmatrix}^{-1} (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n)^{\mathrm{T}}\mathbf{C} \quad (14)$$
$$= \left[\sum_{i=1}^{n} \frac{\mathbf{v}_i \mathbf{u}_i^{\mathrm{T}}}{\sigma_{ii}}\right]\mathbf{C}$$

The calculation in eq 14 employs all n available principal components. However, only a subset, k, of the n available PCs is typically used for the computation. The k PCs are generally those accounting for the largest amount of data variance, i.e., the first few PCs. If too many PCs are used, the model formed will not only contain the general characteristics of the data, but also describe details of the calibration data that are not necessary for generalization and may degrade predictions for data not used in the calibration process. Using only k PCs, eq 14 can be rewritten as

$$\widehat{\mathbf{P}} = \mathbf{V}_k\Sigma_k^{-1}\mathbf{U}_k^{\mathrm{T}}\mathbf{C}$$
$$= (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k)\begin{bmatrix} \sigma_{11} & 0 & . & . & . & 0 \\ 0 & \sigma_{22} & . & . & . & 0 \\ . & . & . & & & . \\ . & . & & . & & \\ . & . & & & . & . \\ 0 & 0 & . & . & . & \sigma_{kk} \end{bmatrix}^{-1} (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k)^{\mathrm{T}}\mathbf{C} \quad (15)$$
$$= \left[\sum_{i=1}^{k} \frac{\mathbf{v}_i \mathbf{u}_i^{\mathrm{T}}}{\sigma_{ii}}\right]\mathbf{C}$$

where k indicates the use of a subset of k of the n available PCs.

After $\widehat{\mathbf{P}}_k$ has been computed, predicted concentrations can be obtained for unknown samples given the response of the unknown to the n sensors. If **y** denotes the $1 \times n$ response of an unknown sample, the estimate of analyte concentrations can be obtained as $\widehat{\mathbf{c}}_y = \mathbf{y}\widehat{\mathbf{P}}_k$ where $\widehat{\mathbf{c}}_y$ denotes the $1 \times s$ vector of predicted concentrations for the s analytes. Thus, PCR can be used to obtain quantitative estimates for all species present in known amounts in the calibration samples.

Since PCR is a linear technique, it will be most successful when sensor responses are known to be linear. Like LDA, PCR can produce overly optimistic results if prediction accuracy is based only on data used to generate the principal components. This problem can be somewhat alleviated by use of a leave-one-out cross-validation procedure with the training data. However, a better way to verify the predictive ability of a model is with the use of external prediction set samples.

F. Applications of PCR

Eklöv and co-workers used PCR to predict the fermentation time of sausage samples.[66] Data were collected with a 14-sensor array at various times in the fermentation process. A subset of the data was used to build a calibration model, with the remainder held out to validate the model. A PCR model using two principal components was more successful at predicting fermentation times than linear regression with any single sensor. However, the authors point out that the sensor responses and the changes in sausage characteristics over time are likely nonlinear. Therefore, PCR may not provide optimal predictions in this case.

In characterizing the storage time of fish samples, Schweizer-Berberich and co-workers used PCR to quantify various compounds producing the odors.[62] Analyses were performed over a period of days to monitor how the odor changed, and how the concentrations of particular analytes changed in relation to the odor. Using responses from an array of eight amperometric sensors for test gas mixtures, they were able to build models to predict the concentrations of sulfur- and nitrogen-containing compounds with reasonable accuracy. This allowed them to predict the concentrations of the same compounds for the fish samples. However, predictive accuracy for other compounds of interest was poor. The poor predictions seem to be more a lack of sensitivity of the sensor to the analytes of interest rather than a failing of PCR.

Bodenhöfer and co-workers demonstrated that sensors coated with chiral polymers could be used to differentiate chiral analytes with PCA.[75] In addition, they used PCR to form models to predict the enantiomeric composition of mixtures containing *R* and *S* forms of chiral analytes. Separate models were generated for each chiral analyte. Various array sizes were investigated to determine how many chiral sensors were necessary to produce accurate quantification. PCR was used on data from an 11 sensor array to predict enantiomeric composition of samples. External prediction sets were used to verify model accuracy. Prediction set errors were within 4% of the actual enantiomeric composition in all cases.

In a recent series of papers, Zellers and co-workers presented a pattern recognition method they called extended disjoint PCR (EDPCR).[84−87] In short, they developed principal components models for each group examined. An unknown could be classified by determining the goodness of fit for each of the PC models and assigning the unknown to the group corresponding to the model of best fit. Quantification of analyte concentration could be achieved once the appropriate model was determined. Monte Carlo experiments were used with EDPCR to evaluate arrays of four SAW sensors with various coatings.[84] Using the optimal coatings, analyte classification was better than 90%. In another study, temperature and humidity were varied to see how these parameters affected SAW sensor responses.[85] Additionally, the same methodology was used to examine the limit of detection for sensor arrays.[86] Monte Carlo experiments showed that the recognition rate for sensor arrays often falls off sharply near the limit of detection. Therefore, the authors suggested that a limit of recognition would be a more meaningful measure of array performance.

A very recent paper by Park, Groves, and Zellers[87] examined the ability of arrays of 2−6 SAW sensors to differentiate pure analytes and components of mixtures. The authors again used their EDPCR/Monte Carlo approach. Arrays of 2−6 sensors were used to classify 16 vapors in two concentration ranges, one near the limit of detection and one well above the limit of detection. Recognition rates were greater than 95% for arrays ranging from 3 to 6 sensors. The 16 vapors were divided into two groups, eight relatively nonpolar vapors in one group and eight relatively polar vapors in the other group. Recognition rates were similar to those obtained for all 16 vapors, but the optimal sensors selected (when considering arrays of less than six sensors) were quite different for the subsets of eight vapors. Mixtures of 2−6vapors were also examined with this array. The authors found that adding sensors did not necessarily improve array performance. Additionally, results indicated that if all possible combinations of vapors are considered, analysis of more than four vapors may be intractable. This paper carefully examines the number of sensors necessary to accomplish a given task. Some of the results presented dispute the idea that large sensor arrays are required for complex vapor sensing tasks.

G. Partial Least Squares

Partial least squares (PLS) provides information similar to that available through PCA and PCR. The calibration matrix, **X**, is decomposed into two matrices, **B** and **D**, such that the vectors $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_k$ are linear combinations of the vectors $\mathbf{u}_1$, $\mathbf{u}_2$, ..., $\mathbf{u}_k$ and the vectors $\mathbf{d}_1$, $\mathbf{d}_2$, ..., $\mathbf{d}_k$ are linear combinations of the vectors $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_k$. Algorithmic details for PLS can be found in refs 53 and 54. Recently, Lang and co-workers demonstrated mathematically the differences between PCR and PLS and described a general regression algorithm containing PCR and PLS as special cases.[88] In short, PLS utilizes more information from the matrix, **C**, of calibration concentration values in generating **B** and **D** than PCR does in generating **U** and **V**.

The columns of **B** are frequently termed PLS scores and can be used to generate plots analogous to the PC plots produced using the columns of **U**. PLS loading plots can be obtained in a similar manner from the columns of **D**. Graphical analysis of such plots provides information similar to the more traditional PCA plots.

Quantification with PLS models also requires an estimate of the regression coefficients in **P**. First, the calibration matrix, **X**, is projected using the *k* latent variables in **B** and **D**:

$$\mathbf{X}_k = \mathbf{B}_k\mathbf{B}_k^{\mathrm{T}}\mathbf{X}\mathbf{D}_k\mathbf{D}_k^{\mathrm{T}} \qquad (16)$$

An estimate of **P** can then be obtained as

$$\hat{\mathbf{P}}_k = \mathbf{X}_k^{+}\mathbf{C} \qquad (17)$$

where $\mathbf{X}_k^{+}$ denotes the Moore−Penrose generalized inverse of $\mathbf{X}_k$.[54] Concentration estimates for unknown samples can then be obtained as described in the PCR section.

Since PLS is quite similar to PCA and PCR, similar precautions should be taken when using PLS. PLS is a linear technique, but nonlinear variations are available. An external prediction set should be used to verify model accuracy, especially when PLS is used to obtain quantitative estimates.

H. Applications of PLS

PLS has been used to separate and quantitate analytes of interest based on responses from sensor array data. Carrasco and co-workers used PLS score plots to show that data from an array of 18 MOS gas sensors could be used to separate perfume samples, where the class of the perfume samples was determined by the volume of sample headspace used for injection.[42] Although complete separation of the samples was achieved, LDA and PCA were shown to be able to separate samples based on perfume type as well as injection volume. Results of such an analysis were not presented using PLS. Niebling and Müller compared models formed with an array of two sensors and models formed with an array of eight sensors.[28] The two-factor PLS model formed with data from the eight-sensor array was superior. This work provides a clear demonstration of the ability of PLS to extract relevant information and disregard redundant information.

Blixt and Borch used an array of 10 MOSFETS, four TGS, and one IR sensor to monitor vacuum-packaged beef.[89] Responses were collected over several months, and they found that only six of the available sensors were stable over this time period. PLS models were generated using all six of the sensors and subsets of the six sensors to predict the degree of spoilage as determined by a sensory panel. The most promising model for prediction of spoilage was based on responses from two MOSFET sensors. After model formation, data from additional samples were collected to validate the models. The model validation provides an excellent demonstration of how to verify the ability of PLS models to generalize.

Seeman and co-workers used reflectometric interference spectroscopy with an array of 10 polymer-coated sensors to examine three test analytes.[90] PLS scores and loadings were examined to see which sensors were contributing the most information to the responses for the three analytes. For quantitation, PLS produced poor results, and a more complex quadratic PLS model was developed that provided improved prediction accuracy. Domanský and co-workers also used nonlinear PLS for quantitation.[91] They used an array of four sensors to monitor hydrogen and ammonia gases. All four sensor responses were necessary to accurately quantitate hydrogen, while two sensors were found to be adequate for ammonia. The nonlinear PLS models formed provided excellent quantitation for both analytes. Similarly, Dickert and co-workers found that they could achieve excellent quantitation of structurally related aromatics (e.g., *o*-xylene and *m*-xylene) by performing nonlinear PLS on responses from an array of four QCMs.[92] Relative humidity was systematically varied to examine its effect on the sensor responses.

Grate and co-workers used arrays of polymer-coated acoustic wave vapor sensors to identify unknown vapors which were not represented in the calibration phase.[93] To accomplish this, models were generated to predict solvation parameters of a given vapor based on sensor response and predetermined parameters of the polymers in the array (e.g., polymer thickness). The calculated solvation parameters are then compared to a database containing parameters for many vapors. This work is unique in that it provides a potential means to identify an unknown vapor even if the unknown vapor was not explicitly represented in the calibration phase.

I. Cluster Analysis

Clustering is the operation of determining which objects are most similar to one another and grouping them accordingly. Similarity can be determined in many ways, one of the most common is by comparing distances in N-dimensional space. N may represent the number of sensor responses or the number of samples, depending upon whether a clustering of sensors or of analytes is desired, but for the purposes of this discussion, N is defined to be equal to the number of sensor responses and the analytes are to be clustered. Each sample can be represented by an N-dimensional vector of sensor responses. Many distance metrics have been reported for use in cluster analysis,[94] but the most commonly used is the Euclidean distance. A general formula for distance is as follows:

$$d_{ij} = \sqrt[p]{\sum_{k=1}^{N} (x_{ik} - x_{jk})^p} \qquad (18)$$

where d_{ij} is the distance between two samples, x_{ik} is the first sample, x_{jk} is the second sample, and p is the number which produces the desired distance metric. For $p = 2$, the distance is the Euclidean distance. For higher values of p, more weight is given to smaller distances.

There are several considerations to keep in mind when working with clustering techniques. The first is that as is frequently the case for neural network analyses, cluster analysis typically requires a large number of training samples. This requirement ensures that the samples from each class will fill the associated N-dimensional space. Second, markedly different results often occur depending upon the choice of clustering method.

Cluster analysis has been used both as an exploratory and as a pattern recognition technique. Relationships between analytes and sensors have been examined. Clustering has frequently been used as a method for selecting out of a large set of sensors the optimal sensor array for classifying a particular set of samples. The performance of sensor arrays for the purpose of identifying analytes has been evaluated based upon the utility of the data collected from those arrays for correctly clustering various analytes.

1. Mapping and Display

Sensor array data can be displayed graphically for visual inspection to give the user qualitative information about the data set. Typically this cannot be done directly as the data sets are usually greater than three-dimensional (meaning that there are more than three sensor responses per sample). The data can be reduced to two or three dimensions for graphical display. The method most commonly reported for this purpose in the analysis of data from sensor arrays is principal components analysis, which has been described earlier in this paper. Another technique, nonlinear mapping, also known as Sammon mapping,[95,96] is included in the data analysis package provided with some commercial instruments.[97,98]

In Sammon mapping, the object is to find a two-dimensional representation which best preserves the original relationships between the data points. The interpoint distances in N dimensions are calculated using a distance metric, typically the Euclidean distance as described above. Then the distances between the points in two dimensions are given by

$$d^*_{ij} = [(y_{i1} - y_{j1})^2 + (y_{i2} - y_{j2})^2]^{1/2} \qquad (19)$$

where the y's are the scores calculated from a principal components analysis, i and j refer to points, and 1 and 2 refer to the coordinates.

The goal is to change iteratively the y for each point to minimize the error function, E

$$E = \frac{1}{\sum_{i>j} (d_{ij})^p} \sum_{i>j} \frac{(d_{ij} - d^*_{ij})^2}{(d_{ij})^p} \qquad (20)$$

The exponent, p, is included to allow weighting of small or large distances. The effect of different values of p on the projection obtained is discussed by Bender and Kowalski.[96] The error function must be minimized using a nonlinear minimization method such as steepest descent.

Two recent studies illustrate the use of this mapping approach. A study was performed using a commercially available instrument with conducting polymer sensors for the purpose of determining whether the instrument could distinguish water vapor from dairy products and slightly varying dairy products from one another.[97] Data from 32 sensors was taken, and the Euclidean distance between each vector was calculated. Sammon mapping was used to project the data points from 32-dimensional space into two or three dimensions for visual inspection. Sammon mapping was also used with a commercially available electronic nose to examine a data set of vapors from balsamic vinegars.[98]

In studies involving many analytes in complex data sets, these graphs may not yield a good separation of the analyte classes. This is not conclusive proof that the analytes cannot be distinguished from one another. In such instances, it may be the case that there are too few dimensions shown to give a satisfactory separation.

2. Hierarchical Clustering

Hierarchical clustering is a technique frequently used in sensor array applications. In hierarchical clustering, the multivariate distances between all pairs of points are first calculated. The results of the clustering depend on the method of clustering and the similarity measure applied to the data set. The most commonly used distance metric in sensor array applications is the Euclidean distance described above, but there are many alternatives.

Sometimes, rather than using the raw distances, similarity values, S_{ij}, are calculated as follows

$$S_{ij} = 1 - \frac{d_{ij}}{d_{ij}(\max)} \qquad (21)$$

where d_{ij} is the distance between points i and j and $d_{ij}(\max)$ is the largest distance between any two samples. Thus, the most distant points in a data set will have $S_{ij} = 0$ and the identical points will have $S_{ij} = 1$.

After distances and similarity values have been calculated, clustering may be performed using one of several methods.[99] Single linkage computes the nearness of a point relative to a cluster according to the distance between the point and the nearest point in the cluster. The complete linkage method determines the distance between a cluster and a point according to the distance between the point and the furthest point in the cluster. The centroid method uses the distance between the point and the center of gravity of points in a cluster. Additionally, the calculation may be weighted by the number of points in the cluster.

Any of these clustering methods can be used to build up a hierarchy of points. Starting with the points forming the data set, the two most similar points are joined. Then the most similar third point is joined, then the fourth, and so on. A hierarchical arrangement is formed which represents the overall degree of clustering of the data points.

Good results can also be obtained with divisive methods. In a divisive method, the set of all objects is first divided into two clusters. Each cluster is successively divided until finally all objects are separated. Each method presents certain computational advantages.[94]

Minimal spanning tree (MST) is another method. The MST is found by connecting a set of data points such that each forms a node of the tree and the sum of all the line segments over the data set is a minimum. Then, clusters are pruned from the tree on the basis of the length of a line segment in comparison to other nearby segments.

Following the formation of clusters, a dendogram may be drawn. A dendrogram is a two-dimensional plot in which the points are assigned equally spaced locations arrayed along the foot of the plot and the vertical axis is related to the degree of similarity of the points. A vertical line originates from each point along the foot of the plot, and the line for any particular point ends when that point has joined a cluster. This allows a simple, visual examination of the overall set of clustering relationships within a data set. Note that clusters are not automatically formed by this method. They may be assigned according to distance or another criterion.

Hierarchical clustering is an unsupervised method. It is best suited to qualitative investigations of relationships in data. Hierarchical clustering has been used in a variety of qualitative applications with artificial nose data.

An array of 32 conducting polymer sensors was exposed to polychlorinated phenol vapors, and the change in electric resistance was measured.[100] Euclidean distance and similarity matrices were calculated from normalized data. Clustering was performed using the complete linkage method to investigate the effect of chemical structure on sensor response. Hierarchical clustering has been used by several groups to select a set of sensors out of a larger set of possible sensors. Holmberg and co-workers used a hybrid array of 15 sensors consisting of 10 MOSFET sensors, four Taguchi gas sensors, and one infrared CO_2 sensor to take data from the odors of five cardboard papers and air.[74] They used hierarchical clustering with a Euclidean distance metric with average between groups linkage to examine the data. After several iterations of analysis, they found that data from four of the original sensors, two MOSFETs and two TGS, produced a clustering which correctly classified all 96 samples. No prediction was done, however, so the result should be considered in terms of the sensors it selected rather than the classification results. To screen candidate sensing layers for a sensor array, Domansky and co-workers deposited 10 thin sensing layers on stainless steel sample electrodes.[91] A Kelvin probe was used to measure the work function, which is a measure of the interaction of the sensing layer with the analyte. Hydrogen and ammonia at three concentrations were presented to the sensing layers. The response time, recovery, reproducibility, and baseline stability for each sensor's signal as a function of time was qualitatively examined. For each sensor and gas combination, calibration curves were made and the responses evaluated on the basis of their sensitivity and linearity with the log of concentration.

Using information gathered from these tests about the linearity, sensitivity, and stability/reproducibility of each sensing layer to both hydrogen and ammonia, a hierarchical clustering analysis was performed. The clustering method chosen was K-means, in which the centroid average of linked samples replaces those samples in further calculations. The data were mean-centered and treated with principal components analysis. One component was selected. The clustering analysis resulted in two layers with good linearity and one with poor linearity, all with high sensitivity to hydrogen, clustering together. Three layers which were clustered together and selective for ammonia also featured one layer with poor linearity, while the other two exhibited good linearity. The remaining four layers formed another cluster. These results were consistent with the qualitative evaluations made previously.

The relationship between sensors has been examined using hierarchical clustering. Dickinson and co-workers fabricated four fiber optic sensors using varying ratios of two monomers which were then photopolymerized.[101] The data set consisted of the response at 40 time points of two of each type of sensor to the presentation of benzene vapor. A hierarchical cluster analysis using Euclidean distance showed that the sensors did not cluster as one would have expected if their responses were simply linearly related to the relative amounts of the two different monomers used to fabricate the sensors.

3. K-Nearest Neighbor Classification

The *K*-nearest neighbor (KNN) technique is simple yet powerful. An unknown is classified in this method according to the majority classification of its *K* nearest neighbors in a training set in *N*-dimensional space. Each of the *N* dimensions corresponds to a sensor response or to a sample, depending upon the desired application. Ties can either be avoided by limiting *K* to odd numbers only or by giving greater weight to closer neighbors. Nearness is determined using a distance metric.

KNN is algorithmically very simple.[102] First the distance matrix between all pairs of points is computed. To determine the nearest neighbors, either the upper or lower part of the diagonally symmetric distance matrix is scanned for the smallest *K* distance values. Whichever class an observation's *K* nearest neighbors belong to is taken to be its class. The algorithm is frequently used with leave-one-out[41] cross-validation, in which the class membership is determined for one member at a time using the remainder of the data set to examine for nearest neighbors. The percent error which is a measure of the predictive ability for the whole data set is then the number of observations correctly classified (one at a time) divided by the total number of observations. Despite its simplicity, KNN classification can be used very effectively with cross-reactive sensor array data.

KNN is a supervised method. This means that knowledge of the dependent variable is used. Parameters of interest include the total number of training samples used, the number of training samples in each class, and the training method used (leave-one-out or leave-*n*-out cross-validation). To evaluate the performance of a KNN model after it has been created, validation should be performed. Methods for validation include using the model to predict the classes of observations not used in the training and scrambling the dependent variable and re-doing the analysis. For the scrambling experiment, the desired outcome is a low correct classification rate, preferably near that expected to be attained by random class assignment.

KNN was used to classify sample responses from an array of four coated quartz crystals. The data set was comprised of the lowest and highest frequencies exhibited during the presentation of each of the seven organic solvents. The data were normalized to eliminate concentration effects.[103] No details were provided about the training procedure, validation method (if any), or number of correct vs incorrect classifications, so it is difficult to evaluate the results. Nakamura and co-workers derived a sensor response model from autoregressive model[104] coefficients.[35] Data were taken for five organic solvents using an array of six AT-cut quartz crystals coated with a plasma polymer film. A principal component analysis was done using the normalized saturation-mass both by itself and with the time constant. KNN was used to quantify the improvement in the separation of analyte classes when both parameters were included. This paper provides an example of using PCA to make two-dimensional plots of the data and then using KNN to quantify the analyte separation by the PCs. It appears that the authors considered only the first two PCs in their analysis. While it is convenient to use only the first two or three PCs for graphical display, KNN can accommodate any number of PCs. There is a possibility that the authors may have had better classification rates using more PCs or different

PCs. As was the case with the previous paper, this paper provided few of the details of the computations.

4. Visually–Empirical Region of Influence

A unique new clustering method has been developed by Osbourn et al. at Sandia National Laboratories. The development of visually–empirical region of influence (VERI) clustering was motivated by shortcomings of other known clustering techniques.[105] Human clustering judgments in two dimensions extended to k-dimensional data are used as the clustering criteria. No input is required from the user beyond the data. Provision is made for both outlier and multiclass data points.

VERI is based on the concept of a region of influence (ROI). In two dimensions, a region is defined around each pair of points such that each pair is clustered only if no other data points lie within the region of influence. The shape of the region has been empirically determined from experiments with human subjects, but the scale and orientation of the ROI are adjusted for each pair of points. To extend the method to k-dimensions, all combinations of three points are considered. Only if two points lie within the ROI and one does not do the two points form a cluster. Thus, a k-dimensional problem is reduced to a series of two-dimensional problems.

The clustering can also be extended to a pattern recognition method.[106] The first step is to normalize the values for each feature. Next, leave-one-out processing of the data should be performed to identify outliers and points in multiple classes. The class of an unknown, or test point, is classified according to which training points it clusters with, if any. Points which do not fall within any defined class are considered outliers.

VERI has been used to identify the best array of j sensors chosen from a larger pool of k possible sensors. The method has been used to choose the optimal two-sensor pair from six possibilities for the simultaneous determination of CO_2 and H_2O.[107] Ricco and co-workers used VERI to analyze all possible combinations of six SAW sensors to form the optimal arrays with between three and six sensors.[108] The data set consisted of the isotherms for 15 organic analytes and water. The best three through six sensor arrays had correct identification rates of 84%, 95.8%, 97.9%, and 98.3%, respectively. The authors noted that the reason the best four- through six-sensor arrays had less than 100% correct identification was that a few points were classified as outliers. The authors proposed that quantification could be done by having a look-up table of the original data and averaging concentrations for the analytes in the table with data most similar to that of the unknown.

Further, one of the best six-sensor arrays identified previously was used to analyze a data set comprised of seven analytes and their 21 binary mixtures at varying concentrations. A 98% correct identification rate was achieved. Finally, the effects of reproducibility, noise, and drift upon VERI pattern recognition were studied. It was found that the addition of noise did not significantly degrade the pure component results up to 28% noise, but 8% noise had a significant effect upon mixtures, and 12% noise had very significant effects.

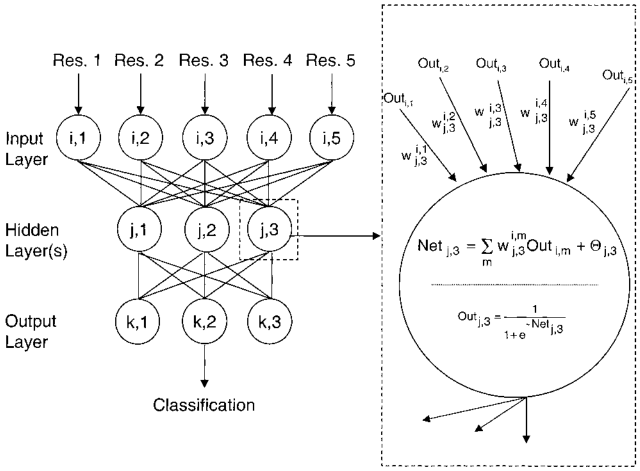


Figure 3. Diagram of a fully connected three-layer feed-forward computational neural network. The notation in the neuron which is shown in detail is as follows: $\mathrm{Out}_{j,3}$ is the output of the neuron which will be passed on to the neurons in the output layer. $\mathrm{Net}_{j,3}$ is the weighted sum of the inputs which is subjected to the sigmoid transfer function which results in $\mathrm{Out}_{j,3}$. The weights between the neurons are denoted as $w^{i,m}_{j,n}$. Finally, $\Theta_{j,3}$ is the bias term.

J. Computational Neural Networks

Computational neural networks (NNs) are a popular technique for sensor array data processing. While there are myriad NN methods in existence, only those used with sensor array data will be described in detail here. Hertz, Krogh, and Palmer included in their book a brief history of NNs and provide references to important papers.[109]

1. Feed-Forward Neural Networks

The feed-forward (FF) NN is the most often used NN technique for the analysis of data from sensor arrays.[4] These networks are also known as multilayer perceptrons and back-propagation NNs (after the most popular method of optimizing the network's weights and biases). These networks are used both as classifiers and as quantitators.

The FF network consists typically of an input layer, a hidden layer, and an output layer, as shown in Figure 3. For sensor arrays, the number of neurons in the input layer is frequently equal to the number of sensors in the array, or the optimal number of sensors in the array, times the number of pieces of information taken from each sensor, which is frequently but not always one. The data fed to the input layer are also frequently scaled.

Each input is passed along to the hidden layer along with a weight. In the hidden layer, a weighted sum of the inputs is computed and a bias term is added, which is equivalent to having an additional input with the fixed value of one. This value is then subjected to a transfer function, which is typically a sigmoid (as illustrated in Figure 3) or hyperbolic tangent function. The optimal number of neurons to include in the hidden layer must be experimentally determined.

The output from the hidden layer is then passed to the output layer. The number of neurons in the output layer usually depends on the application. If the network is being used as a classifier, the number of neurons is typically equal to the number of classes. The same operations are performed in the output neurons as in the hidden layer neurons. The result is then scaled (the reverse of the scaling process at the beginning of the network) and the user gets an answer.

Training FF networks amounts to adjusting the weights and biases. Various algorithms have been developed for this purpose. Arguably the most popular is the back-propagation method.[110] The use of the back-propagation technique requires the optimization of a learning rate and a momentum term. Other training algorithms include quasi-Newton methods[109] such as Broyden−Fletcher−Goldfarb−Shanno (BFGS).[111−114] Such methods offer the advantages of fewer parameters to adjust (no learning rate or momentum), and they are second-order methods, which means more information about the error response surface is taken into account during their training.

Regardless of the algorithm chosen, the training process is an iterative procedure. Observations are presented to the network one by one, and the weights and biases are adjusted after each observation in a direction which minimizes the difference between the true value and the value predicted by the network (or the classification, if the network is being used as a classifier). Some training methods involve adjusting the weights and biases after each pass through the entire training set, i.e., after each epoch. This process continues until training is complete. The method for determining when training is complete is dependent on the optimization algorithm chosen to adjust the weights and biases. Typically an external test set is then used to determine the utility of the network for predicting observations which were not used in training the network.

The performance of the NN after training is dependent upon the starting weights and biases. One set of randomly chosen weights may yield a network with a low error rate, and another set may produce a network which gives inferior results. One way to overcome the starting-point dependence is to train many networks from different random starting points and select one which gives the best performance. Another option is to use an optimization algorithm to choose a reasonable set of starting weights and biases.

Another factor to consider when using FF NNs is overtraining. This occurs when the network memorizes the features of the training data and loses its ability to generalize. One method used to avoid overtraining is the use of a cross-validation (CV) set to monitor the training of the network. After every epoch, the CV set is predicted and training is stopped at the point in training at which the CV set error begins to increase.

Several papers have been written which give guidelines for the proper use of FF NNs to avoid getting results based on chance rather than a true relationship in the data.[115−117] One disadvantage of using a FF NN for classification purposes is that the number of adjustable parameters quickly increases as the number of sensors or classes increases. The ratio of the number of training observations to the number of adjustable parameters in a NN should be kept above 2.[117] The number of adjustable parameters, P, is given by

$$P = (I \times H) + (H \times O) + H + O \qquad (22)$$

where I is the number of units in the input layer, H is the number of neurons in the hidden layer, and O is the number of units in the output layer. For instance, a modest 3−2−3 network has 17 adjustable parameters, necessitating at least 34 training observations, while a modest increase in the number of sensors to 5 and the number of classes to 6, assuming 4 hidden units, results in 54 adjustable parameters, which means at least 108 observations should be in the training set. It is not uncommon to see papers with at least six sensors and eight classes, and if we again assume four hidden neurons (although the actual optimum number in a particular application will vary and must be determined by the user), we get 68 adjustable parameters. Changing to five hidden neurons means an increase to 83 adjustable parameters. Anyone not wishing to take large numbers of training set data observations will probably not wish to use FF NNs for classification.

In addition to the number of training observations required, additional samples should be taken for a cross-validation set. As discussed above, this is used to prevent overtraining the network to the point where it has memorized the training data, which results in overfitting so that the network has no predictive ability. The training process is monitored by predicting the output of the network after each epoch for the cross-validation set. The cross-validation set error will decrease during the course of training the network, until it reaches the point where the network begins to overtrain. At that point, the cross-validation set error will begin to increase, and it is right before that error begins to increase that the training should be terminated.

Following the training of the NN, some form of validation is desirable to show that the model arises from a true relationship between the data and the dependent variable, rather than by chance. A set of observations not used for either training or cross-validation should be held out of the data set and used for this purpose. The model created is then used to predict the dependent variable for the members of the prediction set. If the prediction set error is close to the training and cross-validation set error, the model has validated. If the prediction set error is very large, while the training set error was very good, it is likely that the network has been overtrained and has simply memorized the training set data. Another possibility in that case is that too many adjustable parameters have been used and, again, the training set has simply been memorized and the network has no predictive ability.

One additional method of validating the model is by Monte Carlo experiments. The dependent variable is scrambled, and then the network is trained using this dependent variable and the descriptors used in the original model. If this results in a poor prediction set error, this provides further evidence that the model is valid and did not arise by chance correlations.

In order for the reader to be able to evaluate the computational work done in a paper, the authors should include relevant computational details. This includes the number of layers, the number of neurons in each layer, the number of training observations, whether a cross-validation set was used and if so how many members it contained, and whether a prediction set was used and the number of members in it. If some other form of validation was used, that should be included. For a treatment of FF NNs beyond that presented here, the reader is referred to a review by Burns and co-workers.[116] The review gives an explanation and many examples of applications of the method and, most importantly, further addresses the pitfalls researchers face in using FF networks.

2. Applications of Feed-Forward Neural Networks

Many authors described the application of FF NNs to the analysis of data from sensor arrays. A summary of these applications appears in Table 2. In the table, the type of sensor, the analytes, and a description of the data from the array which is used as input to the NNs is given. Also given is whether the application was for classification or quantitation, the architecture used, and the results reported.

Commercially available instruments were used in a wide variety of applications with FF computational NNs chosen for data analysis. A commercial array of 32 conducting polymer sensors was used to investigate the utility of electronic noses for arson investigation in the detection of accelerants.[118] The instrument was used to analyze neat samples of common accelerants and of strong and weak residues from fire debris with the objective of classifying samples according to type of accelerant and strength (neat, weak, or strong). The network was trained using a fuzzy back-propagation algorithm supplied by the instrument manufacturer. It was found that while initial results were promising, effects resulting from the substrate to which the sample was applied and the treatment during and after the fire should be investigated. While the application is interesting, few of the relevant computational details were supplied.

In another interesting application, He and co-workers used FF networks and sensor array data to correctly classify the smoke from four combustible materials: wood, plastic, grain, and paper.[44] They found that they were able to use the network to obtain a correct classification rate of 100% with the NN approach, which out-performed stepwise discriminant analysis, with 98% correct for the training set and 83% correct for the test set. These results must be interpreted with caution, however, as even though validation was done for the NN, the ratio of training observations to adjustable parameters is low.

Monitoring beer flavor is another sensor array application in which FFNNs have been applied. Gardner and co-workers used a network with an 18:6:3 architecture to classify samples of carrier gas, ethanol, and ethanol tainted with diacetyl. Five samples of each analyte (at the same concentration each time) were presented to the array, leading to a data set of 15 observations. Caution must be used in interpreting the author's report of a 93% successful classification rate for several reasons. First, the network was trained for 100 000 iterations without cross-validation to determine the optimal stopping point. Second, although what the authors termed 3-fold cross-validation was done, the number of adjustable parameters in the network is very large (135) compared to the total number of observations (15). Finally, it is possible that the good results for the 3-fold cross-validation are due to the fact that each of the five samples for each class was identical. Due to the small number of training observations, a different method (such as PCA and/or KNN, for example) would have been more appropriate. Also, it would likely be beneficial, whatever the analysis method, to reduce the number of sensors used from 18 to the minimum number needed to give good classification.

In Sweden, grain is checked by inspectors who try to detect 'off' odors. Borjesson and co-workers compared the performance of an array of MOSFET sensors with NN processing to that of two or three inspectors.[119] Four SnO_2 sensors, 10 MOSFET sensors, and an infrared CO_2 detector made up the array. Grains were classified in one of two ways, either as good or bad or as belonging to one of the following four classes: moldy/musty, acid/sour, burnt, or normal. In addition, the mean intensity of samples classified as having a moldy/musty odor was predicted. The array successfully classified 75% of the samples in the four-class case and 90% of the samples in the good/bad two-class case. Inspectors agreed in the four-class case for 70% of the samples and in the two-class case for 80% of the samples. The correlation coefficient between the values of moldy/musty mean intensity perceived by inspectors and that predicted by the electronic nose was 0.89. The authors concluded that the electronic nose and human inspectors are about equally efficient in classifying grains according to whether they possess an 'off' smell.

In the above example, the ratio of adjustable parameters in the NNs to the number of training set observations was above two, as recommended to avoid chance correlations. As a further measure to ensure the models found were valid, a test set was used following the training. The networks were trained to a specified mean square error or to a maximum of 10 000 iterations. It is possible that training using a cross-validation set, instead, to monitor the performance of the network may have resulted in less training iterations required and in better over all prediction results. The use of cross-validation to monitor network training is nicely illustrated in a paper by Dickert et al.[92] They used a QCM array to simultaneously classify and quantify volatile organic compounds, including xylene isomers. This paper also is a good example of a model with a

Table 2. Feed-Forward Computational Neural Network Applications in Sensor Array Data Analysis

sensors	analytes	data (no. of samples in training set)[c]	usage[a]	architecture (no. of adj. params)[d]	results[b] (validation?)[e]	ref
commercial instrument, 32 conducting polymer sensors	neat and burned samples containing arson accelerants (gasoline, kersosene, mineral spirits, motor oil, diesel fuel, lacquer thinner)	average sensor response in specified time interval (?)	C	?:?:?	initial results promising (y)	118
three organic thin film sensors	water, 2-propanol, ethyl acetate, acetone	four measurements of change in conductivity taken at specific times for three sensors (?)	C	12:?:4	water and propanol successfully classified (y)	152
4 SnO_2, 7 MOSFET, infrared CO_2 detector	grains (wheat, barley, and oats)	voltage shift (200 or 257)	C,Q	12:6:4:M $M = 2$ (116) or 4 (126)	75% 4 class, 90% 2 class (y)	119
four stationary phase coated AT-cut quartz crystals	acetone, benzene, chloroform, pentane and mixtures	lowest frequency from each sensor (?)	C	4:3:4 (31)	all analytes and mixtures successfully identified (n)	153
five semiconducting oxide sensors	CO, methane, propane/butane, methanol	voltage shift (50)	Q	64 neurons in the SOM, 64:6:4 (418)	mean absolute error (MAE) from hybrid network 12.5; MAE for feed-forward network alone 14.3 (y)	128
six piezoelectric quartz crystals	binary mixtures: *n*-octane/chloroform, *n*-octane/*n*-propanol, chloroform/*n*-propanol	frequency change (46)	C	6:4:3 (43) combined with 8 × 8 SOM	>80% classification (n)	129
four metal oxide semiconductor sensors	two red wines from different vineyards	voltage shift (12)	C	4:5:2 (37)	better than standard chemical analysis (y)	154
four coated quartz microbalances	mixtures of *m*- and *p*-xylene, *o*- and *m*-xylene, tetrachloroethylene and toluene, in the presence of humidity	frequency shift (150)	C,Q	4:3:3 (27)	better than 3 ppm accuracy for organics (y)	92
Pt-MOSFET sensor	hydrogen and ethanol mixtures at varying concentrations	49 descriptors from transient response curve (?)	Q	?:?:?	forward selection gives the best results (y)	122
10 MOSFETs, six TGS, one CO_2 sensor	data from *E. coli* batch cultivation to estimate biomass and specific growth rate	five descriptors per sensor (?)	Q	?:?:?	PCA and PLS scores as inputs to the neural network give the best results (y)	
24 conducting polymer sensors	carrier gas, ethanol, diacetyl taint	fractional change in conductance (12)	C	18:6:3 (135)	93% success (y)	155
four thin film oxide semiconductor sensors	CH_3SH, $(CH_3)_3N$, C_2H_5OH, CO	fractional change in resistance (60)	C	4:8:12 (148)	100% correct (n)	156
two semiconducting MOS, 1 humidity sensor	CO, CH_4, humid air	output voltage (231)	C,Q	3:22:22:3 (663)	5% relative error (y)	127
10 CHEMFETs four TGS, one infrared CO_2 sensor	five cardboard packaging papers	drain current, and resistances (80)	C	4:7:5 (75) or 7:7:7:5 (152)	100% correct classification (y)	74
seven coated AT-cut piezoelectric quartz crystals	smoke of four combustible materials (wood, plastic, grain, paper)	frequency change (52)	C	7:6:4 (76)	100% correct classification, NN performed better than discriminant analysis (y)	44
10 MOSFETs, four semiconducting SnO_2, one optical CO_2 sensor	ethanol batch cultivation headspace samples	change in voltage, change in resistance, IR absorption (99)	Q	5:4:1 (29)	4.6% rms error for ethanol predictions (y)	69
eight commercial metal oxide sensors	six types of aviation fuel	change in resistance (87)	C	5:7:5 (82) 5:5:2 (42)	9/9 and 4/4 correct prediction for one pset, 7/9 and 2/4 for a different split of the data to make a pset (y)	157
19 polymer-coated fiber optics with an immobilized dye	nine organic vapors	change in dye fluorescence over time (210)	C, Q	nine individual networks ranging from 7:4:1 (37) to 11:8:1 (105) for classification, one 9−3−1 (34) network for quantification	90% correct prediction set classification, 97% correct prediction set quantification (y)	30
12 commercially available TGS	coffee samples	fractional change in steady-state sensor conductance	C	12:3:3 (51)	86% back-propagation, 93% fuzzy weights	158

Table 2 (Continued)

sensors	analytes	data (no. of samples in training set)[c]	usage[a]	architecture (no. of adj. params)[d]	results[b] (validation?)[e]	ref
four commercially available TGS	89 total patterns (80 or 81)			81/51=1.6	75% BP, 85% FNN	
	water samples			4:6:6 (72)	*n*-fold cross-validation (y)	
	60 total patterns (54)			72/54 = 1.3		
10 reflectometric interference spectroscopy sensors	tetrachloroethene, toluene, octane	change in optical thickness (198)	Q	one net per analyte, 3:3:1, 3:2:1, and 5:2:1, respectively (not fully connected)	RE < 5% (y)	90
21 conducting polymer sensors	flavor notes in beer: control, dimethyl sulfide, diacetyl, hop essence	resistance (?)	C,Q	21:3:4 (82)	flavor classification unsuccessful for two analytes and 54% and 87% for two others (y)	45
				21:2:4 (56)	organoleptic score predicted with accuracy of ± 1.4 (out of 10) with 95% confidence (y)	
10 MOSFET sensors and four TGS	fermented sausage samples	gate voltage at constant drain current and voltage drop (?)	Q	13:2:1 (31)	rmse of 2.7 when predicting fermentation time (?)	66
commercially available array of 32 conducting polymer sensors	multiple classes of:	percent change in resistance over time	C	NN/GANN:	NN/GANN % correct:	124
	fragrances	(172)		4:7:5/9:7:5 (75/110)	87/95 (y)	
	colas	(420)		4:6:2/2:6:2 (44/32)	100/100 (y)	
	hog farm odors	(85)		24:8:6/12:8:6 (254/158)	85/90 (y)	
eight polymer-coated SAW	two analytes: diethyl sulfide, isooctane (previously published data)	frequency shift ratio (?)	C	varied no. of input units from 4 to 8:5:2 (37 to 57 adj. param.) or both back-propagation trained NN and NN trained with a genetic algorithm (GA)	100% correct classification for 4 or 5 inputs, 50% for 6−8 inputs for back-propagation trained network	125
					100% classification for 4−8 input neurons for GA trained NN (y)	
eight TGS	three loose-leaf teas	fractional voltage distance (48)	C	208:80:3 (16963)	90% correct test set classification (y)	120
16 conducting polymer sensors	bacterial and yeast culture headspace	parameters calculated from dynamic sensor response (bacteria = 112, yeast = 128)	C	112:90:13 (11363)	bacteria 93.4% correctly classified (n)	70
				112:90:4 (10534)	yeast 96.9% correctly classified (n)	
four commercially available TGS	ethanol, toluene, *o*-xylene at three different concentrations	steady-state conductance change and conductance rise time (35)	C,Q	8:5:3 (63)	100% classification (y)	82
				4:3:3 (27)	84% toluene, 100% *o*-xylene and toluene quantification (y)	
eight reflectometric interference spectroscopy sensors	cyclohexane, toluene, and nitrobenzene, and binary mixtures thereof	relative change in optical film thickness (108)	C,Q	6:3:3 (33)	107/108 correctly classified (y)	58

[a] C = classification, Q = quantification. [b] Quantitative results are given if reported by the authors; otherwise the qualitative results reported by the authors (e.g., successful classification) are given. [c] Number of observations used for training (if given). [d] Number of adjustable parameters in the neural network, see text for details. [e] Indicates whether any type of cross-validation or prediction was reported.

large ratio of observations to adjustable parameters and the use of a prediction set for validation.

Llobet et al. investigated the use of transient information from a sensor as well as steady-state responses. Although leave-one-out validation was done, the results must be interpreted cautiously due to the small ratio of observations to adjustable parameters. Nonetheless, this paper is an example of the value of the use of dynamic data as well as steady-state data in FFNN analyses.[82] The use of dynamic data has also been investigated in an application involving the classification of three types of tea[120] and for the classification of bacteria and yeast headspace.[70]

A FFNN model was developed to classify varying concentrations of CH_3SH, $(CH_3)_3N$, C_2H_5OH, and CO. The authors report 100% successful classification; however, caution must be used in interpreting those results as the size of the training set was never specified, the use of a prediction set was not mentioned, and the network was trained for 10 000 iterations without a specified stopping criterion. In addition, the network architecture used was 4:8:12, which corresponds to 148 adjustable parameters. This means that the training set should have at least 296 observations. While the number of samples used for training is not explicitly given in the paper, it is apparent from the text and a principal components figure that the number is probably 60, which gives a ratio of observations to adjustable parameters of 0.4, far under the recommended minimum of 2.0.[117]

The selection of which features to use from a sensor array is one aspect of data analysis. It is frequently overlooked, and the steady-state responses from all sensors are used as inputs to the FFNN. However, most classification algorithms perform better when only the features relevant to an application are included.[121] So, for instance, features which are solely noise and features which do not contribute to the separation of classes should be eliminated. The same principle also applies to quantitative analyses.

Various means of feature selection are available, including using forward selection or choosing an optimal set of sensors to use. This problem becomes more important to solve in applications where dynamic information is used and there are several parameters available from each sensor. Also, choosing only the features necessary will reduce the number of adjustable parameters in a NN model and thus reduce the chances of creating a model with poor predictive ability.

Eklov and co-workers investigated methods of finding the best inputs to a FF NN selected from a large pool of available features.[122] They discussed several methods of feature selection. Two data sets were investigated. Descriptors were calculated from the raw data for both data sets, and two methods were used to select inputs which were then used to create NN models. The first method of feature selection involved choosing inputs by forward-selection and evaluating subsets based on the root-mean-square (rms) error of a multiple linear regression model. The second method involved the calculation of PCA and PLS scores to be used as inputs to the NN. The best models for a data set of hydrogen and ethanol mixtures resulted from using PCA and PLS scores as inputs to a NN. A more noisy set of data from an *E. coli* batch cultivation was modeled best using inputs to a NN chosen by forward selection. This is a good demonstration of the fact that there are many possible ways to approach problems in sensor array data processing and the best method to use may depend on the application. Further, the best method may not be known apriori, and various experiments may be tried to find the right method for a particular application.

Another method of choosing features to use in a NN model is to use an optimization algorithm such as the genetic algorithm or simulated annealing. An array of 19 fiber optic sensors, each coated with a polymer containing the immobilized dye Nile red, was used to classify nine organic vapors.[30] The change in fluorescence intensity of the dye over 60 time points (about 20 s) due to the presentation of the organic vapors was monitored. The data for the 19 sensors were examined, and those sensors producing noisy or inconsistent data were eliminated from further analysis. The data were then autoscaled to remove concentration effects. For each of the remaining 10 sensors, 15 descriptors were calculated. These descriptors included the average sensor response over six evenly distributed sets of time points and the most positive and most negative slopes. For each of the nine analytes, a separate NN was constructed. Feature selection was performed using simulated annealing[123] to choose the optimal subset of descriptors with multiple linear regression as the fitness evaluator. The selected features were then submitted to NNs, which were trained using a BFGS optimization method to adjust the weights and biases. The system of NNs correctly classified 100% of the training set data and 90% of the prediction set data. The interesting feature of this work is the calculation of descriptors from the response curves and the use of a simulated annealing optimization algorithm to choose optimal subsets of descriptors.

Kermani et al. investigated the use of a genetic algorithm (GA) combined with NN to select relevant features and NN parameters.[124] They compared the results obtained using the GANN with NN results. They found that while NNs gave satisfactory results, combining them with the GA enhanced their performance.

The use of genetic algorithms has not been confined to feature selection. Srivastava et al. used a GA for the purpose of training a neural network.[125] The purpose of the GA is to optimize the weights. Although they demonstrated its use on a relatively simple two-class problem, the results were very good.

In addition to classification, FF networks have been used for quantification. An ethanol batch cultivation was monitored using an electronic nose.[69] Headspace samples from the cultivation were analyzed to determine the ethanol concentration present in the broth. The best five sensors were determined using PCA. A 5:4:1 FF NN correctly quantified the ethanol concentrations with a root-mean-square error of 4.6%.

In many manufacturing industries, detecting the concentrations of gases such as CO, CH_4, and H_2 is important. Moore and co-workers used an array of six sensors to collect data on the four gases.[126] They used a three-layer network with nine units in the hidden layer but it was only partially connected.

The quantification of carbon dioxide and methane in humid air has been attempted.[127] One-half the data set was used for training, and the entire data set was used for testing. The results for the testing phase may be overly optimistic due to the presence of data used in training the network. To get a more realistic validation, only the half of the data unused in the training phase should be used for testing.

Quantitative analysis with FFNNs includes not only the prediction of analyte concentrations, but also other properties of interest. Organoleptic scores have been predicted for beer samples.[45] The fermentation time of sausage samples has been predicted.[66]

Kraus and Gauglitz[58] compared the performance of FFNNs and self-organizing maps (SOM) (discussed in a later section) in simultaneously classifying and quantifying three volatile organic hydrocarbons and their binary mixtures using data from eight reflectometric interference spectroscopy sensors. They found that both NNs performed well, with the FFNN correctly classifying 107/108 prediction observations and the SOM correctly classifying 106/108 prediction samples. This paper is notable because rather than making replicate measurements of each analyte at each concentration, each measurement of an analyte was at a different concentration (or ratio of concentrations, in the case of the binary mixtures). Also, the prediction measurements were made in the same fashion, using concentration values not used for the training data. This is more likely to give a realistic estimate of network performance in a real-world situation than the use of replicate measurements at certain concentrations. Also, the practice of using replicate measurements has raised objections from reviewers.

Self-organizing maps (SOM) have been used as preprocessing units to FF NNs both for quantification and classification. Data from four pollutant gases taken with five semiconductor oxide sensors were used as input to a 64 neuron SOM.[128] The outputs from the SOM were then used as inputs to a FF NN. The hybrid NN yielded a lower overall mean absolute error of quantification than did a FF network alone. A hybrid network of this type has also been used to identify binary mixtures of organic solvents.[129]

3. Self-Organizing Maps

Self-organizing maps (SOM),[130] also known as Kohonen networks, are unsupervised pattern recognition NNs. Training begins with one observation, which is a vector of dimension N, which corresponds usually to the number of sensors in the array. Each element of the vector, x_i, represents an input to the network. The output units are arranged in an array and connected by weights, w_{ij}, to the inputs. The input observation is compared to each output vector's weight vector, and the output vector with the weight nearest the observation is determined to be the winner.

The weights are updated by a learning rule

$$\Delta w_{ij} = \eta \Lambda(i, i^*)(x_j - w_{ij}) \tag{23}$$

for all i observations and j elements. η is the learning rate. i^* is the winning output unit. $\Lambda(i,i^*)$ is the neighborhood function which is 1 for $i = i^*$ and decreases as the observation becomes further from the outputs. Both η and $\Lambda(i,i^*)$ are user-specified parameters which are typically adjusted as training proceeds. Those units close to the winner have their weights changed significantly, while those further away undergo an increasingly insignificant change. In this way, the weight vectors of the winning neuron and its neighbors are made more similar to the input vector. As is the case with supervised pattern recognition methods, validation of the results should be done by comparing the results obtained with the network for the training set with the results from a test set not used in training the network.

The most common application of the SOM has been for the correction of drift. Marco and co-workers[131] let the SOM continue to learn throughout the operation of the sensor array. The learning rate was kept to a minimum and only the winning neuron was allowed to update its weights. They found that with simulated sensor array drifts of up to 20%, the SOM's success rate remained above 80%; however, if during the drift the SOM was allowed to continue learning, the network experienced almost no loss of performance. SOMs were also used to classify analytes using sensor array data. The classification of combustion gases using data from six commercial tin-oxide sensors was accomplished with less than a 3% error rate. A final supervised step was performed after the SOM was trained to make classification possible. The training patterns were presented again to the network and the winning neurons were labeled according to the class of the corresponding observation. A procedure was implemented for the case where a conflict between observations arose. While the authors did state that they divided the data set into a training and a test set, the total number of observations and the number of observations used for prediction in each experiment were not clearly stated. Another simulation of sensor array data with drift also found the SOM effective in compensating for the effects of the drift.[132] Sensor array data with simulated drift was used to empirically determine the optimal SOM parameters for operating under such conditions.[133]

A self-organizing classifier (SOC), similar to a SOM except with only one neuron representing each class and no neuron interconnections, was investigated for its utility in counteracting the effects of drift and was compared with the use of a Box−Jenkins model of the sensors.[134] The authors found that under conditions where noise is limited and all gases are measured frequently, the SOC was capable of predicting with 100% accuracy.

Finally, the performances of a sensory panel and an electronic nose for the analysis of milk and tomato pastes have been compared.[67]

4. Learning Vector Quantization

Learning vector quantization (LVQ) is a version of a supervised SOM proposed by Kohonen.[130] The algorithm begins with the creation of starting weight vectors for each class, the number of which is specified by the user. The weight vectors can be initialized as the centroids of the data plus a small random deviation (to prevent them all from being equal). Next, an observation is presented to the network and a weighted Euclidean distance is calculated between the observation and each starting weight vector. The weight vector nearest the observation is called the winning neuron. The weight vector second nearest to the observation is termed the runner-up neuron.

If the winning neuron and the observation belong to the same class, only that neuron is updated. If the winning neuron is not of the same class as the observation and the runner-up is, then both neurons are updated. The updates are performed according to the learning function

$$w_{ij}^{new} = w_{ij}^{old} + \delta\alpha(t)[w_{ij}^{old} - x_j] \qquad (24)$$

where w_{ij}^{new} is the jth element of the weight vector for the ith neuron after learning, w_{ij}^{old} is the jth element of the weight vector for the ith neuron before learning, $\alpha(t)$ is a monotonically decreasing learning rate, and x_j is the jth descriptor value of the presented observation. δ is equal to +1 if the winning neuron is of the same class as the observation and −1 if it is not. For the runner-up neuron, δ is zero unless it is of the same class as the observation in which case δ is +1. The updates result in making the neuron more similar to the observation if the output is correct and to the neuron being made less similar if it is incorrect. Neurons which are rarely or never selected in the training process are re-initialized during the training process to the centroid of the poorly identified class plus a small random deviation.

After all of the training data has been presented once to the network, the current performance is evaluated using a cost function.[32] This is typically the percentage of observations correctly classified or some variation thereof. If the cost for this iteration is an improvement of the previous cost, it is accepted. If it is detrimental, it can be accepted based on a certain probability. Training continues until the specified number of epochs (complete presentations of the training data) have elapsed. Like FF computational NNs, LVQ is capable of overtraining; one way to prevent overtraining is by using a cross-validation set. During the prediction phase, the unknown vector is compared to the weight vectors and the one most similar to the unknown vector is the winner and determines the class of the unknown.

Also as with FF computational NNs, validation is necessary to determine that the model was not obtained purely by chance. A prediction set should be used as described for FF NNs.

The performance of LVQ has been compared with that of other NNs. In both cases described below, the LVQ technique was found to out perform FF NNs.

A polymer-coated fiber optic sensor array was used to collect data for 20 organic vapors.[31] The data were autoscaled to remove concentration effects, and descriptors were calculated. Classification was performed using both individual FF networks for each analyte and LVQ. Feature selection for the LVQ models was performed using a genetic algorithm[135,136] with LVQ as the fitness evaluator. Cross-validation was used to monitor the training of the network and to determine a termination point before the occurrence of over-training. The superior performance of LVQ was confirmed by correctly classifying 90% of the observations in an external prediction set, while the individual FF networks collectively classified only 73% of the prediction set observations correctly. The performances of LVQ, FF networks, and fuzzy adaptive resonance theory map (ARTmap) were compared in the task of determining the ripeness of bananas using commercial tin−oxide sensor array data.[137] LVQ correctly classified 92% of banana samples into classes consisting of seven stages of ripeness, while fuzzy ARTmap and FF networks were found to have correct classification rates of 90% and 83%, respectively. Fuzzy ARTmap alone, however, was shown to efficiently learn new samples without degrading its performance on previous samples. In addition, thorough studies were done comparing the generalizability, ability to incrementally learn, and resistance to noise of each of the types of networks.

5. Adaptive Resonance Theory-Based Methods

Compared to the NN methods described up to this point, adaptive resonance theory-based methods offer significant advantages. In contrast to the FF NNs, ART networks require less training data, because the number of adjustable parameters does not grow as rapidly. ART networks train significantly faster than LVQ networks. While the training time difference is acceptable when one wishes to train using only a few sets of features, if a more extensive feature selection requiring many trainings to compare sets of features is desired, the computational time is significantly higher for LVQ than for ART networks.

Adaptive resonance theory (ART) was developed by Grossberg to address what he termed the stability-plasticity dilemma. Briefly, other algorithms such as LVQ will adjust their weight vectors when presented with outlier data to accommodate the new data point, which may result in the degradation of the ability of the network to identify existing classes. In analogy to LVQ, ART networks compare the new observation to the winning neuron, and if it is not similar enough, the network architecture is able to adapt.[32] Many variants exist, including ART1, ART2,[99] fuzzy ART, ARTmap, and fuzzy ARTmap.[138] Just as the other NN methods, these NNs also require cross-validation and prediction to prevent overtraining (in supervised methods) and for model validation, respectively.

Data from a fiber optic sensor array was used with a fuzzy ARTmap NN to identify TCE from among other analytes and to identify the presence or absence of TCE in binary mixtures.[32] The data were autoscaled to remove concentration effects, and descriptors to encode features of the response traces were calculated. Feature selection for the ARTmap models was performed using a genetic algorithm with a fuzzy ARTmap as the fitness evaluator. A cross-validation set was used to prevent overtraining. The final model consisted of 12 descriptors from four sensors. Validation of the model occurred when it successfully classified 94% of the prediction set as to the presence or absence of TCE.

Using previously published data, Shukla and coworkers investigated the utility of an unsupervised ART network for classifying sensor array data.[139] The change in resistance of four doped tin oxide sensors was measured when the array was exposed to varying concentrations of test vapors. Acetone, ethyl methyl ketone, carbon tetrachloride, and xylene were the test vapors. The ratios of three of the sensor outputs to the fourth sensor output were used as input to the ART network. The results were 100% correct classification for the ART network, which is better than the 90% correct classification obtained using FF NNs with the same data. While the authors stated that an independent test set was used, it was not clear whether results reported were for the training set or the test set.

6. Other Neural Network Techniques

Various other NN techniques have been used as well. Some novel techniques are described in papers in this category. An array of 15 sensors, nine MOSFETs, four Taguchi-type sensors, one CO_2 sensor, and one O_2 sensor, was used to analyze the headspace of bacteria for the purpose of classifying them.[140] Classification trees were used to analyze the data, which was correctly classified by the sensor system in 76% of the cases.

The responses of the sensors were monitored over time. Features were extracted from the data by fitting functions to approximate the shape of the true response curves. The six features used from each sensor came from the fitted approximations rather than from the response curves themselves. Seven sensors were eliminated from the analysis on the basis of having signals which were not significantly different from noise. Forty-eight features from the remaining eight sensors were passed to classification trees for further processing. Four features were chosen by the classification tree and yielded a 68% correct classification for a validation set not used in making the model. Classification trees separate data into classes by asking simple questions about the data, which can be answered yes or no.

Nakamoto and co-workers presented a creative strategy for identifying mixture components.[141] The sensor array is used to take an unknown mixture sample. Next, the sensor array is exposed to a sample consisting of a known ratio of the same components appearing in the mixture. The ratio of the known sample is adjusted until the same pattern observed for the unknown is generated for the known sample. The ratio of components in the unknown sample is then determined to be equal to the ratio in the known sample. They applied this technique to two- and three-component mixtures. While this is novel, the drawback is that it can obviously only be applied when the components present in a mixture are known in advance. Because the ratio of components in the known sample must be iteratively adjusted, waiting for steady-state responses would be time-consuming. Therefore, the authors used a method based upon adaptive control theory to make continuous measurements as the ratio is adjusted.

White and co-workers used a delay line NN with a fiber optic sensor array to investigate the functional consequences of the neural processing of sensor data to get an insight into the biological equivalent.[142] They used it to test the analyte classification ability of their computer model of the olfactory bulb of the salamander.

ChemNets are NNs incorporating information from sensor responses and information underlying the theory of the sensors used.[143] Approximately 95% of the final network is based upon theory. The remaining part of the network is a FF computational NN and is thought by the authors to account for the uncertainty encountered in real world situations. ChemNets were constructed on the basis of Taguchi sensor theory. The authors applied the ChemNets to benzene/toluene mixtures.

Several authors used NN methods based on fuzzy logic. Ali and co-workers[144] used a QCM sensor array and a neuro-fuzzy NN to classify six organic solvents. A fuzzy NN was used by Vlachos and Avaritsiotis[145] with an array of tin oxide sensors. The array successfully classified CO, ethanol, and methane and quantified CO in the presence of various humidity levels. The performance was compared with that of back-propagation NNs.

Pie and Jun describe a network consisting of a radial basis function (RBF) NN combined with a fuzzy c-means algorithm (FCMA).[146] They use it both on simulated data and on data from a sensor array. The interesting part of this paper is the combined network description. Classification results for a real data set of four gases are reported; however, it is not specified if the results are for a training or prediction set. They describe the same system and very similar experiments in another paper.[147]

Lazzerini and co-workers proposed what they termed a 'linguistic fuzzy method' for the classification of samples using data from a sensor array.[148] The use of an autoassociative neural network for preprocessing data which is then used as input to a FFNN has been described.[149] The use of recurrent and time-delay NNs has been investigated.[150] The results of analyzing a data set with the abductory induction mechanism (AIM) have been compared with previously published FFNN results for the same previously published data set by Sommer et al.[151]

VII. Summary and Conclusions

The uses of cross-reactive arrays of chemical sensors coupled with pattern recognition methods have been described. Several types of sensors have been developed including metal oxide semiconductors, metal oxide semiconducting field effect transistors, quartz crystal microbalances, surface acoustic wave devices, polymer-coated sensors, and optical sensors. Numerous applications have been demonstrated using these types of sensors in an array format. Additionally, several commercial products have been developed.

The analysis of the data from a sensor array involves selecting a preprocessing approach (if necessary) and a pattern recognition method. The optimal preprocessing method depends on the type of sensors used and the goal of the analysis. For example, a common preprocessing technique used to remove concentration effects from analyte responses is normalization. This is useful when classification of analytes is of interest but detrimental for quantitative applications. Several available preprocessing approaches have been presented and discussed, along with representative examples employing these techniques.

If the steady-state responses from each sensor are used as the inputs to whatever method is chosen, little feature selection is usually required. However, if the time-varying nature of the signals from the sensors are taken into account, the dimensionality of the data must be reduced using, for example, the calculation of principal components or a preprocessing method such as the calculation of descriptors.

Methods such as linear discriminant analysis, principal component analysis, and partial least squares are useful for reducing the dimensionality of a data set so that it can be examined in a two- or three-dimensional plot. For many of the applications presented, the separation provided by these methods was sufficient. Since LDA, PCA, and PLS are all linear methods, they are best used in cases where sensor arrays are known to respond linearly. These methods can be used with nonlinear data; however, care should be taken in interpreting results from such an analysis. An advantage of these approaches is that little calibration data is required for model generation.

Cluster analysis and computational neural networks offer alternatives to the linear techniques mentioned above. These methods offer the potential to accurately model any nonlinearity that may be present. A potential disadvantage of NN approaches is that no statistical information can be generated to describe the models. Additionally, as the number of adjustable parameters for a NN increases, so do the number of training samples required and the amount of training time required.

As a general rule, as the number of analytes in an application increases, so does the complexity of the algorithm used to analyze the data. A notable exception to this rule of thumb is the KNN classification scheme, which has been shown to perform as well as more complicated methods for various applications, although for certain purposes it may be more desirable to use another technique. For example, the computational time increases with the square of the number of observations, for KNN, so for a very large data set, a different classification scheme, such as a fuzzy ARTmap, may be necessary due to time considerations. This difficulty can be mitigated by, for example, using a leave-*n*-out training scheme rather than the customary leave-one-out. Also, because the algorithm is so simple, it cannot take into account observations from an unknown class.

As demonstrated in this review, numerous methods exist for computational analysis of data obtained from arrays of chemical sensors. With advances in array fabrication (e.g., miniaturization of sensor allowing more responses to be recorded in a shorter time), demands on computational methods will be increased. Additionally, computational aspects will play a key role in allowing array based sensors to be used in monitoring a wide range of chemically similar analytes. Past success using cross-reactive arrays of chemical sensors coupled to computational techniques to identify and quantify analytes of interest ensures that this area of research will continue to receive substantial attention in the future.

VIII. References

(1) Dickinson, T. A.; White, J.; Kauer, J. S.; Walt, D. R. *Trends Biotechnol.* **1998**, *16*, 250−258.
(2) Ziegler, C.; Gopel, W.; Hammerle, H.; Hatt, H.; Jung, G.; Laxhuber, L.; Schmidt, H. L.; Schutz, S.; Vogtle, F.; Zell, A. *Biosens. Bioelectron.* **1998**, *13*, 539−571.
(3) Janata, J.; Josowicz, M.; Vanysek, P.; Devaney, D. M. *Anal. Chem.* **1998**, *70*, 179R−208R.
(4) Gardner, J. W.; Bartlett, P. N. *Electronic Noses. Principles and Applications*; Oxford University Press: Oxford, England, 1999.
(5) Diamond, D. *Principles of Chemical and Biological Sensors*; Wiley-Interscience: New York, 1998.
(6) Kress-Rogers, E. *Handbook of Biosensors and Electronic Noses. Medicine, Food, and the Environment*; CRC Press: Boca Raton, FL, 1997.
(7) Nagle, H. T.; Gutierrez-Osuna, R.; Schiffman, S. S. *IEEE Spectrum* **1998**, 22−34.
(8) Byfield, M. P.; May, I. P. *GEC J. Res.* **1996**, *13*, 17.
(9) Ballantine, D. S., Jr., et al. *Acoustic Wave Sensors: Theory, Design, and Physicochemical Applications*; Academic Press: San Diego, 1997.
(10) Grate, J. W.; Abraham, M. H.; McGill, R. A. In *Handbook of Biosensors and Electronic Noses. Medicine, Food, and the Environment*; Kress-Rogers, E., Ed.; CRC Press: Boca Raton, FL, 1997.
(11) Rose-Pehrsson, S. L.; Grate, J. W.; Ballantine, D. S., Jr.; Jurs, P. C. *Anal. Chem.* **1988**, *60*, 2801.
(12) Lonergan, M. C.; Severin, E. J.; Doleman, B. J.; Beaber, S. A.; Grubbs, R. H.; Lewis, N. S. *Chem. Mater.* **1996**, *8*, 2298−2312.
(13) Doleman, B. J.; Sanner, R. D.; Severin, E. J.; Grubbs, R. H.; Lewis, N. S. *Anal. Chem.* **1998**, *70*, 2560.
(14) Severin, E. J.; Sanner, R. D.; Doleman, B. J.; Lewis, N. S. *Anal. Chem.* **1998**, *70*, 1440.
(15) Dickinson, T. A.; White, J.; Kauer, J. S.; Walt, D. R. *Nature* **1996**, *382*, 697−700.
(16) Hierlemann, A.; Schweizer-Berberich, M.; Weimar, U.; Kraus, G.; Pfau, A.; Göpel, W. In *Sensors Update*; Baltes, H., Göpel, W., Hesse, J., Eds.; VCH Publishers: New York, 1996.
(17) Craven, M. A.; Gardner, J. W.; Bartlett, P. N. *Trends Anal. Chem.* **1996**, *15*, 486-493.
(18) Vaihinger, S.; Gopel, W. In *Sensors. A Comprehensive Survey*; Gopel, W., Hesse, J., Zemel, J. N., Eds.; VCH Publishers: New York, 1991; Vol. 2.
(19) Sharaf, M. A.; Illman, D. L.; Kowalski, B. R. *Chemometrics*; John Wiley & Sons: New York, 1986.
(20) Gardner, J. W.; Hines, E. L. In *Handbook of Biosensors and Electronic Noses. Medicine, Food, and the Environment*; Kress-Rogers, E., Ed.; CRC Press: Boca Raton, FL, 1997.
(21) Gardner, J. W.; Craven, M.; Dow, C.; Hines, E. L. *Meas. Sci. Technol.* **1998**, *9*, 120−127.
(22) Ide, J.; Nakamura, Y.; Nakamoto, T.; Moriizumi, T. *IEICE Trans. Electron.* **1998**, *E81C*, 1057−1063.
(23) Maricou, H.; Pereira, D.; Verschuere, L.; Philips, S.; Verstraete, W. *Water Air Soil Pollut.* **1998**, *107*, 423−442.
(24) Nakamura, M.; Sugimoto, I.; Kuwano, H. *Sens. Actuators B* **1996**, *33*, 122−127.
(25) Buhlmann, K.; Schlatt, B.; Cammann, K.; Shulga, A. *Sens. Actuators B* **1998**, *49*, 156−165.
(26) Stuetz, R. M.; Fenner, R. A.; Engin, G. *Water Res.* **1999**, *33*, 442−452.
(27) Stuetz, R. M.; Fenner, R. A.; Engin, G. *Water Res.* **1999**, *33*, 453−461.
(28) Niebling, G.; Muller, R. *Sens. Actuators B* **1995**, *24*−*25*, 805.
(29) Roussel, S.; Forsberg, G.; Steinmetz, V.; Grenier, P.; Bellon-Maurel, V. *J. Food Eng.* **1998**, *37*, 207−222.
(30) Sutter, J. M.; Jurs, P. C. *Anal. Chem.* **1997**, *69*, 856−862.
(31) Johnson, S. R.; Sutter, J. M.; Engelhardt, H. L.; Jurs, P. C.; White, J.; Kauer, J. S.; Dickinson, T. A.; Walt, D. R. *Anal. Chem.* **1997**, *69*, 4641−4648.
(32) Johnson, S. Ph.D. Thesis, The Pennsylvania State University, 1999.
(33) Ratton, L. K., T.; McAvoy, T.; Fuja, T.; Cavicchi, R.; Semancik, S. *Sens. Actuators B* **1997**, *41*, 105.
(34) Nakata, S. A., S.; Nakasuji, M.; Yoshikawa, K. *Anal. Chem.* **1996**, *68*, 2067.
(35) Nakamura, M.; Sugimoto, I.; Kuwano, H.; Lemos, R. *Sens. Actuators B* **1994**, *20*, 231−237.
(36) Shaffer, R. E.; Rose-Pehrsson, S. L.; McGill, R. A. *Field Anal. Chem. Technol.* **1998**, *2*, 179−192.
(37) Shaffer, R. E.; Rose-Pehrsson, S. L.; McGill, R. A. *Anal. Chim. Acta* **1999**, *384*, 305−317.
(38) Sutter, J. M.; Dixon, S. L.; Jurs, P. C. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 77−84.
(39) Luke, B. T. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1279−1287.
(40) Johnson, R. A.; Wichern, D. W. *Applied Multivariate Statistical Analysis*; Prentice-Hall: Englewood Cliffs, NJ, 1982.

(41) *Multivariate pattern recognition in chemometrics*; Brereton, R. G., Ed.; Elsevier: Amsterdam, 1992; Vol. 9.

(42) Carrasco, A.; Saby, C.; Bernadet, P. *Flavour Fragrance J.* **1998**, *13*, 335−348.

(43) Doleman, B. J.; Lonergan, M. C.; Severin, E. J.; Vaid, T. P.; Lewis, N. S. *Anal. Chem.* **1998**, *70*, 4177−4190.

(44) He, X.-W.; Xing, W.-L.; Fang, Y.-H. *Talanta* **1997**, *44*, 2033.

(45) Pearce, T. C.; Gardner, J. W. *Analyst* **1998**, *123*, 2057−2066.

(46) Maul, F.; Sargent, S. A.; Balaban, M. O.; Baldwin, E. A.; Huber, D. J.; Sims, C. A. *J. Am. Soc. Hortic. Sci.* **1998**, *123*, 1094−1101.

(47) Bourrounet, B.; Talou, T.; Gaset, A. *Sens. Actuators B* **1995**, *26−27*, 250−254.

(48) Annor-Frempong, I. E.; Nute, G. R.; Wood, J. D.; Whittington, F. W.; West, A. *Meat Sci.* **1998**, *50*, 139−151.

(49) Keshri, G.; Magan, N.; Voysey, P. *Lett. Appl. Microbiol.* **1998**, *27*, 261−264.

(50) Namdev, P. K.; Alroy, Y.; Singh, V. *Biotechnol. Prog.* **1998**, *14*, 75−78.

(51) Stuetz, R. M.; Engin, G.; Fenner, R. A. *Water Sci. Technol.* **1998**, *38*, 331−335.

(52) Fenner, R. A.; Stuetz, R. M. *Water Environ. Res.* **1999**, *71*, 282−289.

(53) Malinowski, E. R. *Factor Analysis in Chemistry*, 2nd ed.; John Wiley & Sons: New York, 1991.

(54) Kalivas, J. H.; Lang, P. M. *Mathematical Analysis of Spectral Orthogonality*; Marcel Dekker: New York, 1993.

(55) *Sensors Update*; Baltes, H.; Göpel, W.; Hesse, J., Eds.; VCH Publishers: New York, 1996; Vol. 2.

(56) Unpublished results done in collaboration with K. Albert, S. Stitzel, and D. Walt at Tufts University.

(57) Freund, M. S.; Lewis, N. S. *Proc. Natl. Acad. Sci. U.S.A.* **1995**, *92*, 2652.

(58) Kraus, G.; Gauglitz, G. *Chemom. Intell. Lab. Sys.* **1995**, *30*, 211−221.

(59) Nanto, H.; Kondo, K.; Habara, M.; Douguchi, Y.; Waite, R. I.; Nakazumi, H. *Sens. Actuators B* **1996**, *35−36*, 183.

(60) Di Natale, C.; Davide, F. A. M.; D'Amico, A.; Sberveglieri, G.; Nelli, P.; Faglia, G.; Perego, C. *Sens. Actuators B* **1995**, *24−25*, 801.

(61) Wijesundera, C.; Walsh, T. *Aust. J. Dairy Technol.* **1998**, *53*, 141−141.

(62) Schweizer-Berberich, P. M.; Vaihinger, S.; Gopel, W. *Sens. Actuators B* **1994**, *18−19*, 282−290.

(63) Di Natale, C.; Macagnano, A.; Davide, F.; D'Amico, A.; Paolesse, R.; Boschi, T.; Faccio, M.; Ferri, G. *Sens. Actuators B* **1997**, *44*, 521.

(64) Ulmer, H.; Mitrovics, J.; Noetzel, G.; Weimar, U.; Gopel, W. *Sens. Actuators B* **1997**, *43*, 24.

(65) Mitrovics, J.; Ulmer, H.; Weimar, U.; Gopel, W. *Acc. Chem. Res.* **1998**, *31*, 307−315.

(66) Eklov, T.; Johansson, G.; Winquist, F.; Lundstrom, I. *J. Sci. Food Agric.* **1998**, *76*, 525−532.

(67) Di Natale, C.; Macagnano, A.; Paolesse, R.; Mantini, A.; Tarizzo, E.; D'Amico, A.; Sinesio, F.; Bucarelli, F. M.; Moneta, E.; Quaglia, G. B. *Sens. Actuators B* **1998**, *50*, 246−252.

(68) D'Amico, A.; Di Natale, C.; Macagnano, A.; Davide, F.; Mantini, A.; Tarizzo, E.; Paolesse, R.; Boschi, T. *Biosens. Bioelectr.* **1998**, *13*, 711−721.

(69) Liden, H.; Mandenius, C. F.; Gorton, L.; Meinander, N. Q.; Lundstrom, I.; Winquist, F. *Anal. Chim. Acta* **1998**, *361*, 223−231.

(70) Gibson, T. D.; Prosser, O.; Hulbert, J. N.; Marshall, R. W.; Corcoran, P.; Lowery, P.; Ruck-Keene, E. A.; Heron, S. *Sens. Actuators B* **1997**, *44*, 413.

(71) Mandenius, C. F.; Eklov, T.; Lundstrom, I. *Biotechnol. Bioeng.* **1997**, *55*, 427−438.

(72) Marth, M.; Maier, D.; Honerkamp, J.; Rapp, M. *J. Chemom.* **1998**, *12*, 249−259.

(73) Lane, A. J. P.; Wathes, D. C. *J. Dairy Sci.* **1998**, *81*, 2145−2150.

(74) Holmberg, M.; Winquist, F.; Lundstrom, I.; Gardner, J. W.; Hines, E. L. *Sens. Actuators B* **1995**, *26−27*, 246−249.

(75) Bodenhofer, K.; Hierlemann, A.; Seeman, J.; Gauglitz, G.; Christian, B.; Koppenhoefer, B.; Gopel, W. *Anal. Chem.* **1997**, *69*, 3058−3068.

(76) Hong, H.-K.; Shin, H. W.; Yun, D. H.; Kim, S.-R.; Kwon, C. H.; Lee, K.; Moriizumi, T. *Sens. Actuators B* **1996**, *35−36*, 338.

(77) Lau, K. T.; Micklefield, J.; Slater, J. M. *Sens. Actuators B* **1998**, *50*, 69−79.

(78) Inoue, H.; Park, S. S.; Bescher, E. P.; Xu, Y.; Mackenzie, J. D. *J. Sol-Gel Sci. Technol.* **1998**, *11*, 67−86.

(79) Barko, G.; Hlavay, J. *Anal. Chim. Acta* **1998**, *367*, 135−143.

(80) Grate, J. W.; Patrash, S. W.; Kaganove, S. N.; Wise, B. M. *Anal. Chem.* **1999**, *71*, 1033.

(81) Auge, J.; Hauptmann, P.; Hartmann, J.; Rosler, S.; Lucklum, R. *Sens. Actuators B* **1995**, *26−27*, 181−186.

(82) Llobet, E.; Brezmes, J.; Vilanova, X.; Sueiras, J. E.; Correig, X. *Sens. Actuators B* **1997**, *41*, 13.

(83) Gardner, J. W.; Bartlett, P. N. *Sens. Actuators B* **1996**, *33*, 60−67.

(84) Zellers, E. T.; Batterman, S. A.; Han, M.; Patrash, S. J. *Anal. Chem.* **1995**, *67*, 1092.

(85) Zellers, E. T.; Han, M. *Anal. Chem.* **1996**, *68*, 2409.

(86) Zellers, E. T.; Park, J.; Hsu, T.; Groves, W. A. *Anal. Chem.* **1998**, *70*, 4191.

(87) Park, J.; Groves, W. A.; Zellers, E. T. *Anal. Chem.* **1999**, *71*, 3877−3886.

(88) Lang, P. M.; Brenchley, J. M.; Nieves, R. G.; Kalivas, J. H. *J. Multivar. Stat.* **1998**, *65*, 58−70.

(89) Blixt, Y.; Borch, E. *Int. J. Food Microbiol.* **1999**, *46*, 123−134.

(90) Seeman, J.; Rapp, F.-R.; Zell, A.; Gauglitz, G. *Fresenius J. Anal. Chem.* **1997**, *359*, 100.

(91) Domansky, K.; Baldwin, D. L.; Grate, J. W.; Hall, T. B.; Li, J.; Josowicz, M.; Janata, J. *Anal. Chem.* **1998**, *70*, 473.

(92) Dickert, F. L.; Hayden, O.; Zenkel, M. E. *Anal. Chem.* **1999**, *71*, 1338−1341.

(93) Grate, J. W.; Wise, B. M.; Abraham, M. H. *Anal. Chem.* **1999**, *71*, 4544−4553.

(94) Massart, D. L.; Kaufman, L. *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*; John Wiley and Sons: New York, 1983.

(95) Sammon, J. W. *IEEE Trans. Comput.* **1969**, *C-18*, 401.

(96) Kowalski, B. R.; Bender, C. F. *J. Am. Chem. Soc.* **1973**, *95*, 686.

(97) Visser, F. R.; Taylor, M. *J. Sens. Stud.* **1998**, *13*, 95−120.

(98) Anklam, E.; Lipp, M.; Radovic, B.; Chiavaro, E.; Palla, G. *Food Chem.* **1998**, *61*, 243−248.

(99) Vandeginste, B. G. M.; Massart, D. L.; Buydens, L. M. C.; De Jong, S.; Lewi, P. J.; Smeyers-Verbeke, J. *Handbook of Chemometrics and Qualimetrics: Part B*; Elsevier: Amsterdam, 1998.

(100) Masila, M.; Sargent, A.; Sadik, O. A. *Electroanalysis* **1998**, *10*, 312−320.

(101) Dickinson, T. A.; Walt, D. R.; White, J.; Kauer, J. S. *Anal. Chem.* **1997**, *69*, 3413−3418.

(102) Cover, T. M.; Hart, P. E. *IEEE Trans. Inform. Theory* **1967**, *IT-13*, 21.

(103) Barko, G.; Papp, B.; Hlavay, J. *Talanta* **1995**, *42*, 475.

(104) Kay, S. M.; Marple, S. L. *Proc. IEEE* **1981**, *69*, 1380.

(105) Osbourn, G. C.; Martinez, R. F. *Pattern Recognit.* **1995**, *28*, 1793.

(106) Osbourn, G. C.; Bartholomew, J. W.; Ricco, A. J.; Frye, G. C. *Acc. Chem. Res.* **1998**, *31*, 297−305.

(107) Hoyt, A. E.; Ricco, A. J.; Bartholomew, J. W.; Osbourn, G. C. *Anal. Chem.* **1998**, *70*, 2137.

(108) Ricco, A. J.; Crooks, R. M.; Osbourn, G. C. *Acc. Chem. Res.* **1998**, *31*, 289−296.

(109) Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction to the Theory of Neural Computation*; Addison-Wesley: Redwood City, CA, 1991.

(110) Rumelhart, D. E.; McClelland, J. L. Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Volume 1: Foundations; MIT Press: Cambridge, MA, 1986.

(111) Broyden, C. G. *J. Inst. Maths. Appl.* **1970**, *6*, 76.

(112) Fletcher, R. *Comput. J.* **1970**, *13*, 317.

(113) Goldfarb, D. *Math. Comput.* **1970**, *24*, 23.

(114) Shanno, D. F. *Math. Comput.* **1970**, *24*, 647.

(115) Hanson, S. J. In *Backprogpagation: Theory, Architectures, and Applications*; Chauvin, Y., Rumelhart, D. E., Eds.; Lawrence Erlbaum Associates: Hillsdale, NJ, 1995.

(116) Burns, J. A.; Whitesides, G. M. *Chem. Rev.* **1993**, *93*, 2583−2601.

(117) Livingstone, D. J.; Manallack, P. T. *J. Med. Chem.* **1993**, *36*, 1295−1297.

(118) Barshick, S. A. *J. Forensic Sci.* **1998**, *43*, 284−293.

(119) Borjesson, T.; Eklov, T.; Jonsson, A.; Sundgren, H.; Schnurer, J. *Anal. Technol. Instrum.* **1996**, *73*, 457−461.

(120) Corcoran, P.; Lowery, P.; Anglesea, J. *Sens. Actuators B* **1998**, *48*, 448−455.

(121) Duda, R. O.; Hart, P. E. *Pattern Classification and Scene Analysis*; Wiley: New York, 1973.

(122) Eklov, T.; Martensson, P.; Lundstrom, I. *Anal. Chim. Acta* **1999**, *381*, 221−232.

(123) Sutter, J. M.; Jurs, P. C. In *Data Handling in Science and Technology. Adaptation of Simulated Annealing to Chemical Optimization Problems*; Kalivas, J. H., Ed.; Elsevier: Amsterdam, 1995; Vol. 15.

(124) Kermani, B. G.; Schiffman, S. S.; Nagle, H. T. *IEEE Trans. Biomed. Eng.* **1998**, *46*, 429−439.

(125) Srivastava, A. K.; Shukla, K. K.; Srivastava, S. K. *Microelectron. J.* **1998**, *29*, 921−931.

(126) Moore, S. W.; Gardner, J. W.; Hines, E. L.; Gopel, W.; Weimar, U. *Sens. Actuators B* **1993**, *15−16*, 344−348.

(127) Huyberechts, G.; Szecowka, P.; Roggen, J.; Licznerski, B. W. *Sens. Actuators B* **1997**, *45*, 123.

(128) Brudzewski, K.; Osowski, S. *Sens. Actuators B* **1999**, *55*, 38−46.

(129) Di Natale, C.; Davide, F. A. M.; D'Amico, A.; Hierlemann, A.; Mitrovics, J.; Schweizer, M.; Weimar, U.; Gopel, W. *Sens. Actuators B* **1995**, *25*, 808.

(130) Kohonen, T. *Self-Organizing Maps*, 2nd ed.; Springer: Berlin, 1997.

(131) Marco, S.; Ortega, A.; Pardo, A.; Samitier, J. *IEEE Trans. Instrum. Meas.* **1998**, *47*, 316−321.
(132) Davide, F. A. M.; Di Natale, C.; D'Amico, A. *Sens. Actuators B* **1994**, *18−19*, 244−258.
(133) Di Natale, C.; Davide, F. A. M.; D'Amico, A. *Sens. Actuators B* **1995**, *26−27*, 237−241.
(134) Holmberg, M.; Winquist, F.; Lundstrom, I.; Davide, F.; Di Natale, C.; D'Amico, A. *Sens. Actuators B* **1996**, *35−36*, 528.
(135) Hibbert, D. B. *Chemom. Intell. Lab. Sys.* **1993**, *19*, 227.
(136) Lucasius, C. B.; Kateman, G. *Chemom. Intell. Lab. Sys.* **1993**, *19*, 1.
(137) Llobet, E.; Hines, E. L.; Gardner, J. W.; Franco, S. *Meas. Sci. Technol.* **1999**, *10*, 538−548.
(138) Kasuba, T. *AI Expert* **1993**, *8*, 18.
(139) Shukla, K. K.; Das, R. R.; Dwivedi, R. *Sens. Actuators B* **1998**, *50*, 194−203.
(140) Holmberg, M.; Gustafsson, F.; Hornsten, E. G.; Winquist, F.; Nilsson, L. E.; Ljung, L.; Lundstrom, I. *Biotechnol. Tech.* **1998**, *12*, 319−324.
(141) Nakamoto, T.; Okazaki, T.; Mornzumi, T. *Sens. Actuators B* **1997**, *41*, 183.
(142) White, J.; Dickinson, T. A.; Walt, D. R.; Kauer, J. S. *Biol. Cybernetics* **1998**, *78*, 245−251.
(143) Wang, Z.; Hwang, J. N.; Kowalski, B. R. *Anal. Chem.* **1995**, *67*, 1497−1504.
(144) Ali, Z.; O'Hare, W. T.; Sarkodie-Gyan, T.; Theaker, B. J. *J. Therm. Anal. Calorim.* **1999**, *55*, 371−381.
(145) Vlachos, D.; Avaritsiotis, J. *Sens. Actuators B* **1996**, *33*, 77−82.
(146) Ping, W.; Jun, X. *Sens. Actuators B* **1996**, *37*, 169−174.
(147) Ping, W.; Jun, X. *Meas. Sci. Technol.* **1996**, *7*, 1707−1712.
(148) Lazzerini, B.; Maggiore, A.; Marcelloni, F. *Electron. Lett.* **1998**, *34*, 2229−2231.
(149) Gnani, D.; Guidi, V.; Ferroni, M.; Faglia, G.; Sberveglieri, G. *Sens. Actuators B* **1998**, *47*, 77−83.
(150) Schweizer-Berberich, M.; Goppert, J.; Hierlemann, A.; Mitrovics, J.; Weimar, U.; Rosenstiel, W.; Gopel, W. *Sens. Actuators B* **1995**, *26−27*, 232−236.
(151) Sommer, V.; Tobias, P.; Kohl, D.; Sundgren, H.; Lundstrom, I. *Sens. Actuators B* **1995**, *28*, 217−222.
(152) Barker, P. S.; Chen, J. R.; Agbor, N. E.; Monkman, A. P.; Mars, P.; Petty, M. C. *Sens. Actuators B* **1994**, *17*, 143−147.
(153) Barko, G.; Hlavay, J. *Talanta* **1997**, *44*, 2237.
(154) Di Natale, C.; Davide, F. A. M.; D'Amico, A.; Nelli, P.; Groppelli, S.; Sberveglieri, G. *Sens. Actuators B* **1996**, *33*, 83−88.
(155) Gardner, J. W.; Pearce, T. C.; Friel, S.; Bartlett, P. N.; Blair, N. *Sens. Actuators B* **1994**, *18−19*, 240−243.
(156) Hong, H.-K.; Shin, H. W.; Park, H. S.; Yun, D. H.; Kwon, C. H.; Lee, K.; Kim, S. T.; Moriizumi, T. *Sens. Actuators B* **1996**, *33*, 68−71.
(157) McCarrick, C. W.; Ohmer, D. T.; Gilliland, L. A.; Edwards, P. A. *Anal. Chem.* **1996**, *68*, 4264.
(158) Singh, S.; Hines, E. L.; Gardner, J. W. *Sens. Actuators B* **1996**, *30*, 185−190.

CR9800964