3

Sensor Array System for Gases Identification and Quantification

Walaa Khalaf
University of Calabria
Italy

1. Introduction

This chapter represents a proposed Electronic Nose (ENose) system has been designed in the laboratories of the university of Calabria (Italy), this system has the ability to detect a gas type and then to estimate its concentration. This chapter explain the design idea, the principles of the detection and quantification mathematical model, and the method of calculating the parts per million (ppm) of the gases. The fast evaporation rate and toxic nature of many Volatile Organic Compounds (VOCs) could be dangerous at high concentration levels in air and working ambient for the health of humans, therefore the detection of these compounds (i.e. VOCs) has become a serious and important task in many fields. In fact, the VOCs are also considered as the main reason for allergic pathologies, lung and skin diseases. Other applications of systems for gas detection are in environmental monitoring, food quality assessment (Zhang et al., 2008), disease diagnosis (Casalinuovo & Pierro, 2006; Gardner et al., 2000), and airport security (Lee et al., 2002).

There are many research contributions on the design of an electronic nose system based on using tin oxide gas-sensors array in combination with Artificial Neural Networks (ANN) for the identification of the Volatile Organic Compounds (VOC's) relevant to environmental monitoring, Srivastava (Srivastava, 2003) used a new data transformation technique based on mean and variance of individual gas-sensor combinations to improve the classification accuracy of a neural network classifier. His simulation results demonstrated that the system was capable to successfully identify target vapors even under noisy conditions. Simultaneous estimates of many kinds of odor classes and concentrations have been made by Daqi et al (Daqi & Wei, 2007); they put the problem in the form of a multi-input/multi-output (MIMO) function approximation problem.

In literature several different approximation models have been adopted. In particular a multivariate logarithmic regression (MVLR) has been discussed in (Cohen et al., 2003), a quadratic multivariate logarithmic regression (QMVLR) in (Penza et al., 2002), while a multilayer perceptron (MLP) has been experimented in (Lee et al., 2002). Finally, support vector machines (SVM) has been used in (Distante et al., 2003; Pardo & Sberveglieri, 2005, Wang et al., 2005).

To identify the type of analyte we use the support vector machine (SVM) approach, which was introduced by Vapnik (Vapnik, 1998) as a classification tool and strongly relies on

statistical learning theory. Classification is based on the idea of finding the best separating hyperplane (in terms of classification error and separation margin) of two point-sets in the sample space (which in our case is the Euclidean seven-dimensions vector space, since each sample corresponds to the measures reported by the seven sensors which constitute the core of our system). Our classification approach includes the possibility of adopting kernel transformations within the SVM context, thus allowing calculation of the inner products directly in the feature space without explicitly applying the mapping (Cristianini & Shawe-Taylor, 2004).

As previously mentioned, we used a multi-sensor scheme and useful information is collected by combining the readings of all the used sensors. On this work, combining the information coming from several sensors of diverse types under different heater voltages values are sent to a learning system which has the ability to identify the gas and to estimate its concentration.

This chapter is organized as follows. In Section 2 we describe our Electronic Nose (ENose), while Section 3 gives an overview of the SVM approach. Section 4 is devoted to the description of our experiments involving five different types of analytes (Acetone, Benzene, Ethanol, Isopropanol, and Methanol). While section 5 contains the results of gas identification and quantification. Finally the conclusions are drawn in Section 6.

2. Electronic Nose System

An array of gas sensors with a learning system constitutes what is called Electronic Nose (ENose), the response of this sensor array constitutes an odor pattern (Pearce et al., 2003). A single sensor in the array should not be highly specific in its response but should respond to a broad range of compounds, so that different patterns are expected to be related to different odors.

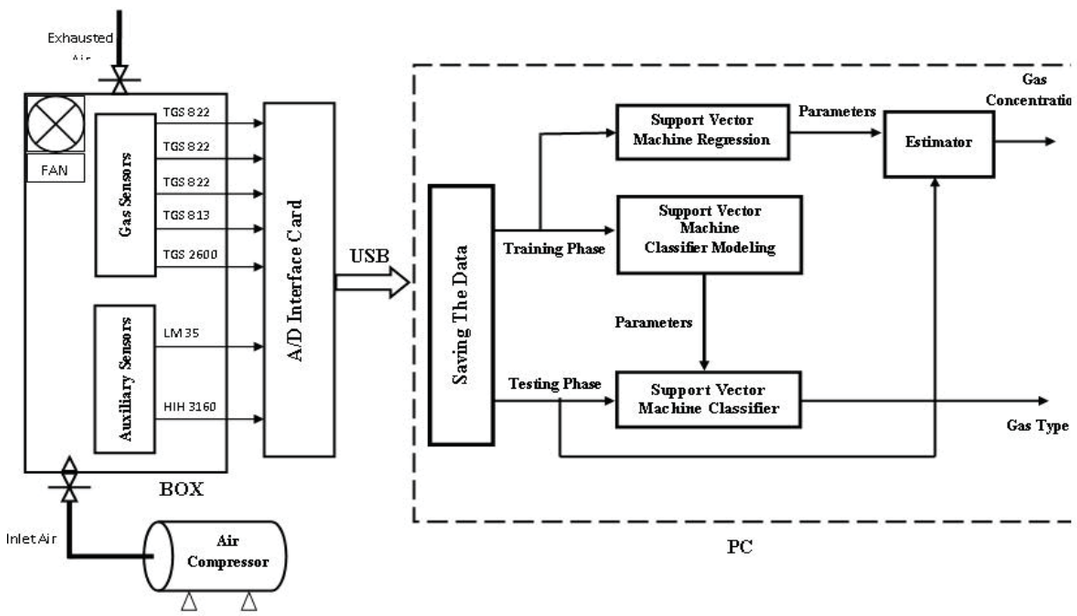

Fig. 1. Block diagram of our Electronic Nose system.

Our system (Figure 1) consists of five different types of gas sensors supplied with different heater voltages to improve the selectivity and the sensitivity of the sensors which are from the TGS class of FIGARO USA INC. The sensing element is a tin dioxide (SnO_2) semiconductor layer. In particular three of them are of TGS-822 type, each one being supplied with a different heater voltage (5.0 V, 4.8 V, and 4.6 V, respectively, see Figure 2), one of the TGS-813 type, and the last one is of the TGS-2600 type. Because the gas sensor response is heavily affected by environmental changes, two auxiliary sensors are used for the temperature (LM-35 sensor from National Semiconductor Corporation), and for the humidity (HIH-3610 sensor from Honeywell).

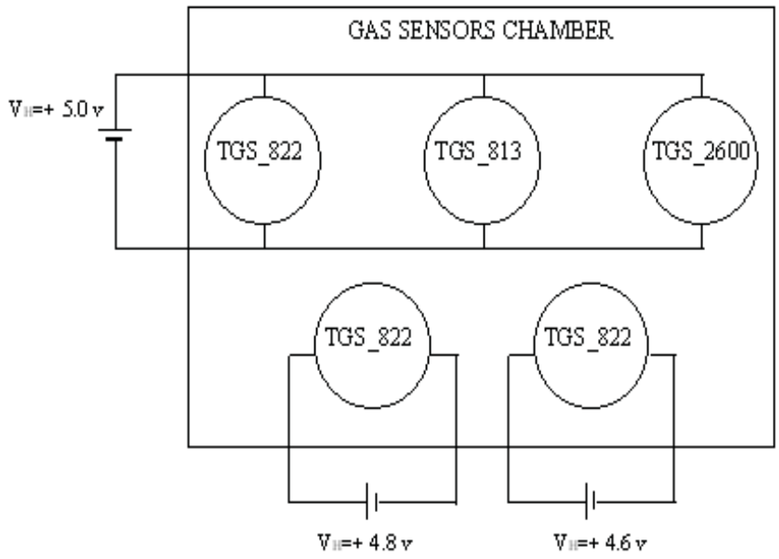

Fig. 2. Block Diagram of gas sensors chamber

The gas sensors and the auxiliary sensors are put in a box of 3000 cm^3 internal volume. Inside the box we put a fan to let the solvent drops evaporate easily. All sensors are connected to a multifunction board (NI DAQPad-6015), which is used in our system as an interface between the box and the PC. The National Instruments DAQPad-6015 multifunction data acquisition (DAQ) device provides plug-and-play connectivity via USB for acquiring, generating, and logging data; it gives 16-bit accuracy at up to 200 kS/s, and allows 16 analog inputs, 8 digital I/O, 2 analog outputs, and 2 counter/timers. NI DAQPad-6015 includes NI-DAQmx measurement services software, which can be quickly configured and allows us to take measurements with our DAQ device. In addition NI-DAQmx provides an interface to our LabWindows/CVI running on our Pentium 4 type PC.
The integrated LabWindows/CVI environment features code generation tools and prototyping utilities for fast and easy C code development. It offers a unique, interactive ANSI C approach that delivers access to the full power of C Language. Because LabWindows/CVI is a programming environment for developing measurement applications, it includes a large set of run-time libraries for instrument control, data acquisition, analysis, and user interface. It also contains many features that make developing measurement applications much easier than in traditional C language environments.

For support vector machine (SVM) training and testing in multi-class classification we use LIBSVM-2.82 package (Chang & Lin, 2001). LIBSVM-2.82 uses the one-against-one approach (Knerr et al., 1990) in which, given **k** distinct classes, **k(k -1)/2** binary classifiers are constructed, each one considering data from two different classes. LIBSVM provides a parameter selection tool for using different kernels and allows cross validation. For median-sized problems, cross validation might be the most reliable way for parameter selection. First, the training data is partitioned into several folds. Sequentially a fold is considered as the validation set and the rest are for training. The average of accuracy on predicting the validation sets is the cross validation accuracy (Gallant et al., 1993). In particular the leave-one-out cross validation scheme consists of defining folds which are singleton, i.e. each of them is constituted by just one sample.

3. Support Vector Machine (SVM)

Support vector machines (SVM) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. This family of classifiers has both abilities: to *minimize* the empirical classification error and to *maximize* the geometric margin. Hence it is also known as *maximum margin classifier approach* (Abe, 2005).

An important feature of the SVM approach is that the related optimization problems are convex because of *Mercer's conditions* on the kernels (Cristianini & Shawe-Taylor, 2004). Consequently, they haven't local minima. The reduced number of non-zero parameters gives the ability to distinguish between these system and other pattern recognition algorithms, such as neural networks (Cristianini & Shawe-Taylor, 2000).

3.1 The Optimal Separating Hyperplane

A separating hyperplane is a linear function that has the ability of separating the training data without error as shown in Figure 3.

Suppose that the training data consists of n samples $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$,…$(\mathbf{x}_n, y_n)$, $x \in \Re^d$, $y \in \{-1, +1\}$ that can be separated by a hyperplane decision function

$$\mathrm{D}(\mathrm{x}) = < \mathrm{w} \,.\, \mathrm{x} > + b = 0 \tag{1}$$

with appropriate coefficients w and $\boldsymbol{b}$ (Cherkassky & Mulier, 1998; Hastie et al., 2001; Vapnik, 1998). Notice that the problem is *ill-posed* because the solution may be not unique and then some constraint has to be imposed to the solution to make the problem *well-posed* (Distante et al., 2003).

A separating hyperplane satisfies the constraints that define the separation of the data samples:

$$\begin{aligned} &< \mathrm{w} \,.\, x_i > + b \geq +1 \quad \text{if } y_i = +1 \\ &< \mathrm{w} \,.\, x_i > + b \leq -1 \quad \text{if } y_i = -1, \qquad i = 1, 2, ..., n \end{aligned} \tag{2}$$

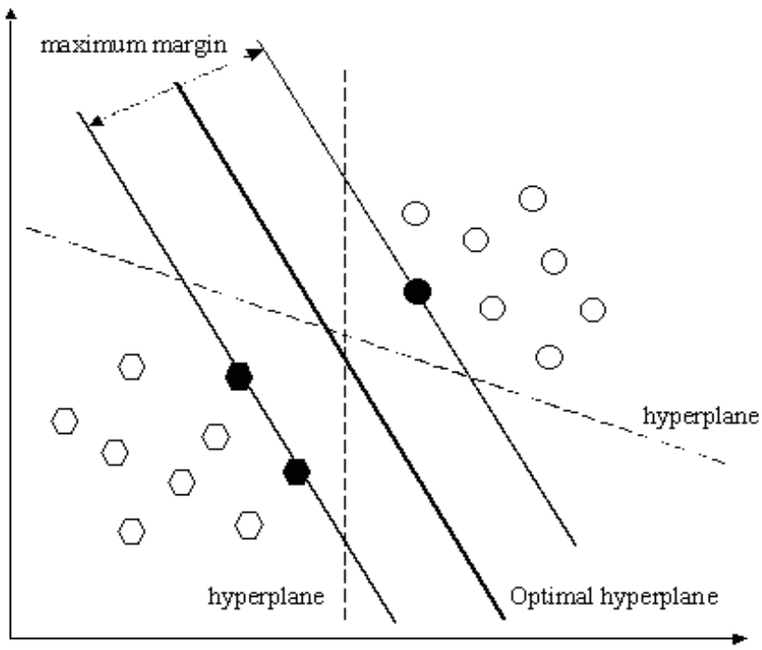

Fig. 3. The optimal hyperplane for two data sets

Or in more compact form (notation)

$$y_i[< \mathrm{w} \ . \ x_i > + b] \geq 1 \qquad i = 1, 2, \ldots, n \tag{3}$$

For a given separable training data set, all possible separating hyperplanes can be represented in the form of equation 3. The formulation of the separating hyperplanes allows us to solve the classification problem directly. It does not require estimation of density as an intermediate step (Abe, 2005). When $D(\mathrm{x})$ is equal to 0, this hyperplane is called *separating hyperplane* as shown in Figure 4.
Let d*i* be the signed distance of the point x*i* from the separating hyperplane

$$d_i = \frac{< \mathrm{w} \ . \ x_i > + b}{\| \mathrm{w} \|} \tag{4}$$

where the symbol $| | \mathrm{w} | |$ denotes the norm of w. From this equation follows that

$$d_i \ \| \mathrm{w} \| = < \mathrm{w} \ . \ x_i > + b \tag{5}$$

and using the constraints (3), we have

$$y_i d_i \ \| \mathrm{w} \| \geq 1 \tag{6}$$

So for all $\mathbf{x}_i$ the following inequality holds:

$$\frac{1}{\|\mathrm{w}\|} \leq y_i d_i \tag{7}$$

Notice that $y_i d_i$ is always positive quantity. Moreover, $1 / \|\mathrm{W}\|$ is the lower bound on the distance between the points $\mathbf{x}_i$ and the separating hyperplane (w, b).

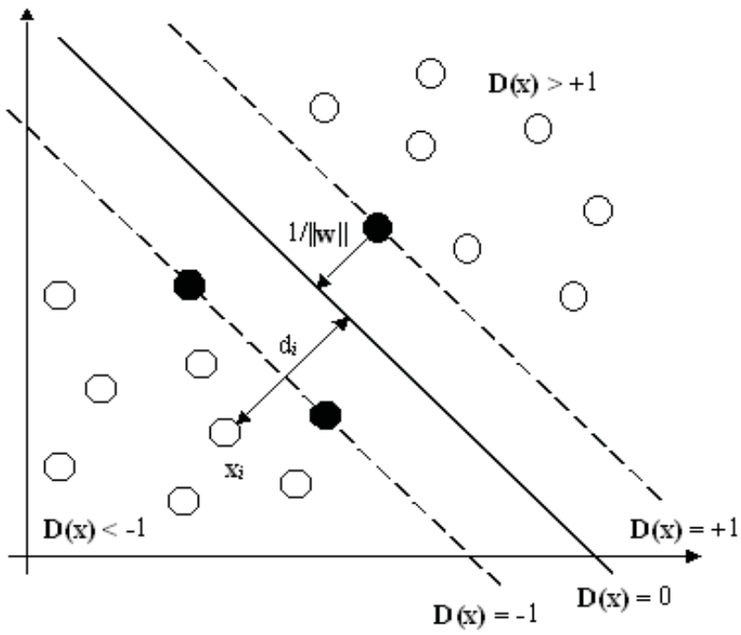

Fig. 4. Separating hyper plane for two data sets as well as the two support vectors

The purpose of the "1" in the right hand side of inequality (equation 3) for establishing a *one-to-one* correspondence between separating hyperplanes and their parametric representation. This is done through the notion of canonical representation of a separating hyperplane (Pontil & Verri, 1998).
The optimal hyperplane is given by maximizing the margin, γ, subject to the constraints (3). The *margin* is given by (Cristianini & Shawe-Taylor, 2000),

$$\begin{aligned}
\gamma(\mathrm{w}, b) &= \min_{y_i=-1} d_i + \min_{y_i=+1} d_i \\
&= \min_{y_i=-1} \frac{< \mathrm{w}.x_i > +b}{\|\mathrm{w}\|} + \min_{y_i=+1} \frac{< \mathrm{w}.x_i > +b}{\|\mathrm{w}\|} \\
&= \frac{1}{\|\mathrm{w}\|}\left(\min_{y_i=-1} (< \mathrm{w}.x_i > +b) + \min_{y_i=+1} (< \mathrm{w}.x_i > +b) \right) \\
&= \frac{2}{\|\mathrm{w}\|}
\end{aligned} \tag{8}$$

Thus the optimal hyperplane is the one that minimizes

$$\Phi(\mathrm{w}) = \frac{1}{2}\|\mathrm{w}\|^2 \tag{9}$$

Because Φ(w) is independent of *b*, changing *b* moves it in the normal direction to itself, and hence the margin remains unchanged but the hyperplane is no longer optimal in that it will be nearer to one class than the other.

3.2 Support Vector Classification

Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression of multi dimensional data sets (Gutierrez-Osuna, 2003; Pearce, 2003). They belong to the family of generalized linear classifiers. This family of classifiers has both the abilities of minimizing the empirical classification error and maximizing the geometric margin. In fact a SVM is also known as maximum margin classifier (Distante et al., 2003).

SVM looks for a separating hyperplane between the two data sets. The equation of such hyperplane is defined by

$$f(\mathrm{x}) = \mathrm{w}^T\mathrm{x} + b = 0 \tag{10}$$

where **w** is the weight vector which defines a direction perpendicular to the hyperplane, **x** is the input data point, and ***b*** is the bias value (scalar), for a proper normalization. The margin is equal to $||\mathrm{w}||^{-1}$. Therefore maximizing the margin is equivalent to minimizing $||\mathbf{w}||$. The advantage of this maximum margin criterion is both robustness against noise and uniqueness of the solution.

In many practical cases the data are not linearly separable, therefore the hyperplane tries to both maximize the margin and minimize the sum of classification errors at the same time. The error ξ_i of a point (x_i, y_i) ($y_i \in \{-1,+1\}$ represents the class membership) with respect to a target margin γ and for a hyperplane defined by *f* is:

$$\xi_i = \xi((x_i, y_i), f(x_i), \gamma) = \max(0, \gamma - y_i f(x_i)) \tag{11}$$

where ξ_i is called the margin slack variable which measures how much a point fails to have margin . If y_i and $f(\mathrm{x}_i)$ have different signs the point x_i is misclassified because

$$\xi_i > \gamma > 0 \tag{12}$$

The error ξ_i is greater than zero if the point x_i is correctly classified but with margin smaller than γ .

$$\gamma > \xi_i > 0 \tag{13}$$

Finally, the more x_i falls in the wrong region, i.e. satisfies equation 12, the bigger is the error. The cost function to be minimized is:

$$\frac{1}{2} || \mathrm{w} ||^2 + C \sum_i \xi_i \tag{14}$$

where C is a positive constant, which determines the trade off between accuracy in classification and margin width (Burges, 1998). Therefore, this constant can be regarded as a *regularization parameter*. When C has a small value, the optimal separating hyperplane tends to maximize the distance with respect to the closest point, while for large values of C; the optimal separating hyperplane tends to minimize the number of non-correctly classified points.

If the original patterns are not linearly separable, they can be mapped by means of appropriate kernel functions to a higher dimensional space called *feature space*. A linear separation in the feature space corresponds to a non-linear separation in the original input space (Wang et al., 2005). Kernels are a special class of functions that permit the inner products to be calculated directly in the feature space, without explicitly applying the mapping. The family of kernel functions adopted in machine learning range from simple linear and polynomial mappings to sigmoid and radial basis functions (Mouller et al., 2001). In this paper linear kernel is used.

4. Experimental

In our experiments we used five different types of volatile species with different concentrations. They are acetone, methanol, ethanol, benzene, and isopropanol. The data set for these volatile species is made up of samples in R^7 space where each sample correspond to the outputs of the gas and auxiliary sensors.

Analyte Concentration (ppm)	Volume of Pure Analyte (cm^3)
10	0.03
50	0.15
100	0.30
200	0.60
400	1.20
800	2.40
1000	3.00
2000	6.00

Table 1. Analyte concentration vs. Analyte volume

Our box (see Figure 5) contains the PCB (Printed Circuit Board) where we fixed two different types of sensors i.e. gas sensors and auxiliary sensors. It also contains a fan for circulating the analyte inside during the test. The system encompasses one input for inlet air coming from an air compressor which has been used to clean the box and the gas sensors after each test. One output is used for the exhaust air. The inner dimensions of the box are 22 cm length, 14.5 cm width, and 10 cm height, while the effective volume is 3000 cm^3.

The amount of volatile compounds needed to create the desired concentration in the sensor chamber (our box) was introduced in the liquid phase using high-precision liquid chromatography syringe. Since temperature, pressure and volume were known, the liquid needed to create the desired concentration of volatile species inside the box could be calculated using the ideal gas theory, as we explain below. The analyte concentration versus analyte volume injected is shown in Table 1.

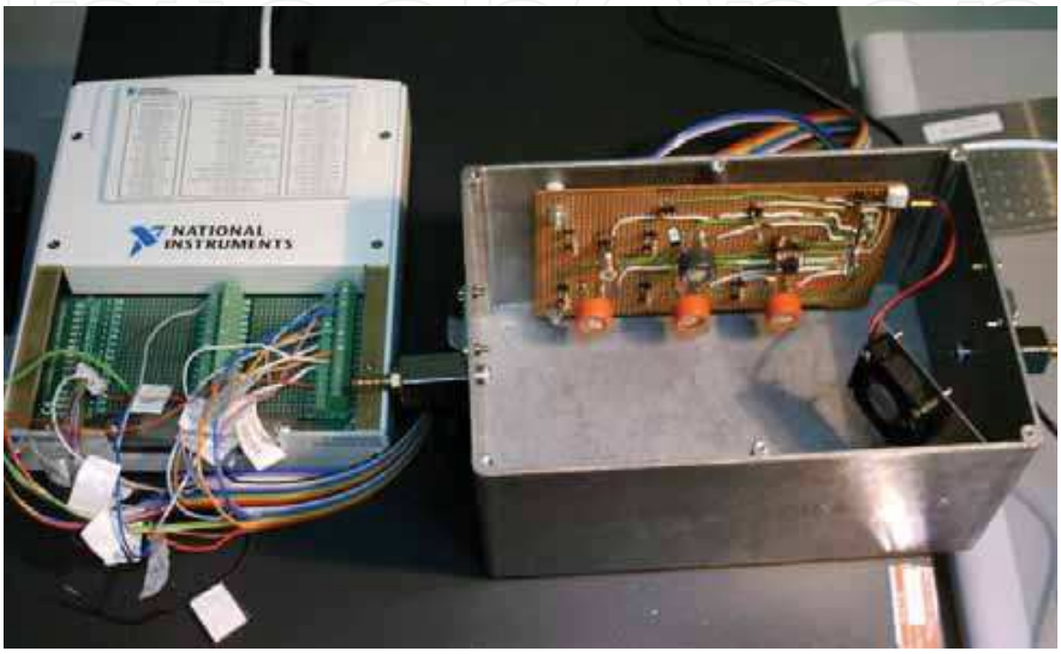

Fig. 5. Our box including the sensors and the fan, connected to the Interface card.

A syringe of 10 μ*l* is used for injecting the test volatile compounds. In the experiments we used five different types of volatile compounds, acetone, methanol, ethanol, benzene, and isopropanol. We take methanol as an example for calculating the ppm (parts-per-million) for each compound. Methanol has molecular weight *MW* = 32.04 g/mol and density ρ = 0.7918 g/cm^3. The volume of the box is 3000 cm^3; therefore, for example, to get 100 ppm inside the box, from Table 1, we used 0.3 cm^3 of methanol, or, equivalently, 0.3 *ml*.

Methanol Concentration (ppm)	Methanol quantity (μ*l*)
40	0.2
100	0.5
200	1.0
400	2.0
1000	5.0
1400	7.0
2000	10.0

Table 2. Methanol concentration vs. methanol quantity

The density of methanol is

$$\partial = \frac{P \times MW}{R \times T} \tag{15}$$

where
∂ = the density of the gas of Methanol in g/l,
P = the Standard Atmospheric Pressure (in atm) is used as a reference for gas densities and volumes (equal 1 atm),
MW = Molecular Weight in g/mol,
R = universal gas constant in atm/mol.K (equal 0.0821 atm/mol.K),
T = temperature in Kelvin ($T_K = T_C + 273.15$).
As a result we get d = 1.33 g/l.

$$Mass = v_{gas} * \partial = v_{liq} * \rho \tag{16}$$

where v_{gas} is the volume occupied by the gas of methanol which is equal to $0.3*10^{-3}$ l, ∂ is the density of the gas of methanol as calculated before, ρ is the constant density of methanol, therefore;

$v_{\text{liq}} = (v_{gas} \text{ x } \partial) \ / \ \rho \Rightarrow v_{\text{liq}} = (0.3 * 10^{-3} * 1.33) \ / \ 0.7918$

the volume (v_{liq}) is $0.503*10^{-6}l$ which provides 100 ppm of methanol. This means that if we want to get 100 ppm of methanol we must put 0.503 μl of methanol as liquid in the box by using the syringe. Table 2 shows different concentrations of methanol (in ppm) versus its quantities (in μl).

5. Results

In our experiments we used 22 concentration samples for acetone, 22 for benzene, 20 for ethanol, 23 for isopropanol, and 21 for methanol. For each concentration the experiment was repeated twice, thus a total number of 216 classification calculations was performed. We put the desired quantity of solvent (in ppm) previously calculated inside the BOX, switching ON the fan which is inside the box to let the solvent drops evaporate easily. The program starts reading the data that are coming from the seven sensors which form our system.
After few seconds, when the signals start to be stable, we switch OFF the fan and then we save the data in a file that indicates also the class label of the current sample. After that we must clear the BOX and the sensors by supplying a compressed air coming from an Air Compressor. We repeat twice this procedure for each gas type and for each concentration.
In the first analysis, we used a SVM with linear kernel, and we applied a multi-class classification by using the LIBSVM-2.82 package (Chang & Lin, 2001). The optimal regularization parameter C was tuned experimentally by minimizing the leave-one-out cross-validation error over the training set.
In fact the program was trained as many times as the number of samples, each time leaving out one sample from training set, and considering such omitted sample as a testing sample check the classification correctness. The classification correctness rate is the average ratio of

the number of samples correctly classified and the total number of samples. The results are shown in Table 3 for different values of *C*. By using linear kernel we got 100.00% classification correctness rate for C = 1000 adopting a leave-one-out cross-validation scheme. We remark that such results are better than those obtained by supplying all sensors by the same heater voltage (in such case, in fact, the best classification correctness rate was 94.74%).

C values	Classification Correctness Rate %
10	91.24
50	96.31
100	96.77
500	98.62
800	99.08
1000	100.00
2000	99.54

Table 3. Multiple C values with linear kernel

Once the classification process has been completed, the next step is to estimate the concentration of the classified analyte. To this aim, we use again the support vector machine approach but in this time as a regression system.
Finally we considered (Table 4) the correlation coefficient (*C.C*) as a measure for the estimation accuracy (Penza et al., 2002), the correlation coefficient is a number between 0.0 and 1.0. If there is no relationship between the predicted values and the actual values the correlation coefficient is 0.0 or very low (the predicted values are no better than random numbers). As the strength of the relationship between the predicted values and actual values increases so does the correlation coefficient. A perfect fit gives a coefficient of 1.0.

Analyte Type	C.C for SVM regression method
Acetone	0.982431
Benzene	0.989445
Ethanol	0.974048
Isopropanol	0.985179
Methanol	0.973584

Table 4. Gases concentration estimation results

Thus the higher correlation coefficient (near to 1.0) the better is the regressor (Cohen et al., 2003). Correlation coefficient is calculated as follows:

$$C.C = \frac{\sum_{i=1}^{n} X_i \hat{X}_i - \frac{\sum_{i=1}^{n} X_i \sum_{i=1}^{n} \hat{X}_i}{n}}{\sqrt{(\sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n})(\sum_{i=1}^{n} \hat{X}_i^2 - \frac{(\sum_{i=1}^{n} \hat{X}_i)^2}{n})}} \quad (17)$$

where *C.C* is the correlation coefficient, X are the actual values, $\hat{X}$ are the predicted values, and n is the number of data points.
Table 5 shows the real concentrations with respect to the results of the suport vectore regression method of benzene.

Real Concentrations	Estimated Concentrations
18	17.99
36	46.69
54	53.98
72	72.13
90	82.87
108	101.15
126	118.84
144	146.65
162	161.98
180	179.99
234	260.09
270	279.95
324	344.26
360	377.28
414	427.67
468	468.00
540	520.62
630	671.43
720	719.99
810	755.55
900	798.71
1080	953.05

Table 5. Real and Estimated Concentrations of Benzene.

6. Conclusion

The results demonstrate that our system has the ability to identify the type of analyte and then estimate its concentration. The best correctness rate was 100.00%. Also the values obtained in terms of concentration estimates appear quite satisfactory. Supplying three similar sensors (TGS-822) with different heater voltages, improved the performance of the system. Future work will be devoted to decrease the number of gas sensors and to use another type of sensors, like QCM sensors.

7. References

Abe, S. (2005). Support Vector Machines for Pattern Classification, Springer-Verlag.
Burges, C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, Vol. 2, No.2, 1998, 121-167.

Chang, C. & Lin, C (2001). Libsvm: A Library for Support Vector Machines. [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm.

Cherkassky, V. & Mulier, F. (1998). Learning From Data, Concepts, Theory, and Methods. John Wiley and Sons.

Cohen, J.; Cohen, P.; West, S. G.; & Aiken, L. S. (2003). Applied Multiple Regression/Correlation Analysis for the Behavioural Sciences, 3rd Ed.; Hillsdale, NJ: Lawrence Erlbaum Associates.

Cristianini, N. & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press.

Cristianini, N. & Shawe-Taylor, J. (2004). Kernel Methods for Pattern Analysis, Cambridge University Press.

Daqi, G. & Wei, C. (2007). Simultaneous Estimation of Odor Classes and Concentrations Using an Electronic Nose with Function Approximation Model Ensembles. Sensors and Actuators B, No. 120, 2007, 584-594.

Distante, C.; Ancona, N. & Siciliano, P. (2003). Support Vector Machines for Olfactory Signals Recognition. Sensors and Actuators B, Vol. 88, No. 1, 2003, 30-39.

Gallant, S. I. (1993). Neural Network Learning and Expert Systems, MIT press.

Gardner, J.; Shin, H.; & Hines, E. (2000). An Electronic Nose System to Diagnose Illness. Sensors and Actuators B, No. 70, 2000, 19-24.

Gutierrez-Osuna, R. (2003). Pattern Analysis for Machine Olfaction: A Review. IEEE Sensors Journal, Vol. 2. No.3, June 2003, 189-202.

Hastie, T.; Tibshirani, R. & Friedman, J. (2001). The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Springer-Verlag, New York.

Knerr, S.; Personnaz, L. & Dreyfus, G. (1990). Single-Layer Learning Revisited: a Stepwise Procedure for Building and Training a Neural Network. Neurocomputing: Algorithms, Architectures and Applications, Springer-Verlag, 1990, 169-184.

Lee, D.; Ban, S.; Lee, M. & Kim, Y. (2002). SnO_2 Gas Sensing Array for Combustible and Explosive Gas Leakage Recognition. Sensors Journal IEEE, Vol. 2, No. 3, June 2002, 140-149.

Mouller, K.; Mika, S.; Ratsch, G.; Tsuda, K. & Scholkopf, B. (2001). An Introduction to Kernel-Based Learning Algorithms. IEEE Trans. on Neural Networks, Vol. 12, No. 2, March 2001, 181-201.

Pardo, M. & Sberveglieri, G. (2005). Classification of Electronic Nose Data with Support Vector Machines. Sensors and Actuators B, Vol. 107, 2005, 730-737.

Pearce, T. C.; Schiffman, S. S.; Nagle, H. T. & Gardner, J. W. (2003). Handbook of Machine Olfaction: Electronic Nose Technology, WILEY-VCH.

Penza, M.; Cassano, G. & Tortorella, F. (2002). Identification and Quantification of Individual Volatile Organic Compounds in a Binary Mixture by Saw Multisensor Array and Pattern Recognition Analysis. Measurement Science and Technology, Vol. 13, 2002, 846-858.

Pontil, M. & Verri, A. (1998). Properties of support vector machines, Neural Computation, Vol. 10, No. 4, May 1998, 955-974.

Srivastava, A. K. (2003). Detection of Volatile Organic Compounds (VOCs) Using SnO_2 Gas-Sensor Array and Artificial Neural Network. Sensors and Actuators B, Vol. 96, 2003, 24-37.

Vapnik, V. N. (1998). Statistical Learning Theory, John Wiley and Sons.

Wang, X.; Zhabg, H. & Zhang, C. (2005). Signals Recognition of Electronic Nose Based on Support Vector Machines. Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 3394-3398, China, 18-21 August 2005, Guangzhou.

Zhang, H.; Chang, M.; Wang, J. & Ye, S. (2008). Evaluation of Peach quality Indices Using an Electronic Nose by MLR, QPST, and BP Network. Sensors and Actuators B, Vol. 134, No. 1, 2008, 332-338.

Recent Advances in Technologies

Edited by Maurizio A Strangio

ISBN 978-953-307-017-9

Hard cover, 636 pages

Publisher InTech

Published online 01, November, 2009

Published in print edition November, 2009

The techniques of computer modelling and simulation are increasingly important in many fields of science since they allow quantitative examination and evaluation of the most complex hypothesis. Furthermore, by taking advantage of the enormous amount of computational resources available on modern computers scientists are able to suggest scenarios and results that are more significant than ever. This book brings together recent work describing novel and advanced modelling and analysis techniques applied to many different research areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Walaa Khalaf (2009). Sensor Array System for Gases Identification and Quantification, Recent Advances in Technologies, Maurizio A Strangio (Ed.), ISBN: 978-953-307-017-9, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-technologies/sensor-array-system-for-gases-identification-and-quantification

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen