A Two-Tiered Classification Algorithm for Identification of Binary Mixtures of VOCs

Robi Polikar and Bryan Healy
Rowan University
136 Rowan Hall, 201 Mullica Hill Road, Glassboro, NJ 08028, USA
polikar@rowan.edu

Abstract

Several classification techniques have been developed with varying degrees of success for automated identification of VOCs, however, the problem becomes considerably more challenging when more then one VOC is present. This is because not only the response of the sensors to certain VOCs may be too strong to mask the response of the sensors to other VOCs in the environment, but also the responses of the sensors to VOCs may not have enough separability information if the specificity of the sensors is not adequate. We propose the following procedures for these two isssues in identification of binary mixtures of VOCs: a nonlinear cluster transformation technique is employed to increase pattern separability and a two-step classification is used to identify dominant and secondary VOCs separately. Results demonstrate the feasibility of the combined approach.

1. Introduction

Volatile organic compounds (VOC) can be found in a variety of settings, including industrial (such as wastewater) as well as residential (such as drinking water supplies, hospitals) sites. Most VOCs find their way into the environment through human causes such as pollution in industrial areas or fuel spills. These compounds can have a disastrous effect on the environment through premature degradation of the surrounding area and health hazards to people in the area that use the contaminated resources. The need for an accurate, cost–effective and objective system for detection and identification of VOCs is therefore undisputed.

Various laboratory based methods exist for the examination of water or air samples and detection of these compounds, but they are often expensive and not located near the source of the pollution causing difficulties in continuous testing. A system employing an electronic nose (Enose) can be used for on-field detection and classification saving both time and money [1]. The data generated by the Enose can then be analyzed by a pattern recognition system for automated identification of the VOC present in the environment. In fact, several such techniques have been proposed over the last decade for identification of (single) VOCs, each with varying degrees of success [2-5]. However, the problem becomes considerably more challenging, when the VOCs appear in a mixture, and the individual components of this mixture needs to be identified [6;7]. This difficulty arises from primarily two factors: First, in many cases the sensor(s) may have a very similar response to two very different compounds, a direct result of inadequate specificity of the sensor. Second, the sensor response to one of the components in the mixture may be so strong that the response to the other components may be completely masked. In this study, we consider binary mixtures of such VOCs, where we refer to them as dominant and secondary VOCs, respectively.

We present a composite approach to the above mentioned two issues. We propose a classification system that first applies a preprocessing algorithm, nonlinear cluster transformation or nonparametric discriminant analysis, in order to increase the separability of the data to aid in classification. We then use a two-level classification system: first the separability algorithm is applied to the raw data followed by a neural network classifier to determine the dominant VOC only. Then, a second level of neural network is used, based on the dominant VOC classification, to determine the secondary VOC in the mixture.

The electronic nose system used to generate the data analyzed in this study is an array of six quartz crystal microbalances (QCM), a well known chemical sensor commonly used in practice for VOC detection. Each QCM was coated with a different polymer, chosen to maximize the specificity of the sensor array for the specific VOCs of interest. As shown next, even a careful selection of polymers did not provide a well-separated feature space for the identification of VOCs.

To date, on a database that includes 24 binary combinations of five dominant and seven secondary VOCs, we have obtained performance figures as high as 87% on the dominant VOC and 81% on the secondary VOC.

2. Increasing Pattern Separability

In general, classification algorithms work well when the underlying data distributions have adequately large intercluster distances between patterns of different classes, and small intracluster distances between patterns of the same class. For most real-world problems, however, this is rarely the case as patterns corresponding to different classes overlapping in the feature space are usually the norm then the exception. Therefore, feature extraction algorithms are typically used as a preprocessing step to classification, whose fundamental objective is to reduce the dimensionality of pattern vectors without loosing discriminatory information. The general problem of feature extraction can be formulated as one of determining a mapping of the form $\mathbf{y} = f(\mathbf{x})$, or $\mathbf{y} = W^T\mathbf{x}$, that transforms pattern vectors onto a lower dimensional space in which the corresponding feature vectors are better separable. Several well-established techniques have been used (and sometimes misused) for this purpose. For example, the principal component analysis (PCA) is one such popular technique, however, PCA does not take the separability of the patters into consideration [8]. The Fisher Linear Discriminant (FLD) also achieves dimensionality reduction, but by taking a criterion function into consideration: the ratio of intercluster to intracluster distances. The FLD projects the data onto a lower dimensional space where this criterion function is maximized. Consequently, FLD is a feature reduction algorithm that ensures maximum separability of patterns in the transformed space [8]. However, FLD has its limitations. First, regardless of the dimension of the original pattern, the FLD transforms a pattern vector onto a feature vector, whose dimension can be at most *C-1*, where *C* is the number of classes. Second, a matrix inversion used in FLD requires that N-$C > d$, where N is the number of training data and d is the dimension of the pattern vector. These limitations can sometimes be quite restrictive depending on the dimensionality, the number of classes and the number of training data available. However, a modification of FLD originally described in [9], and known as nonparametric discriminant analysis (NDA), removes the above mentioned restrictions by redefining the intercluster distances.

Another technique that can be used for increasing pattern separability is the nonlinear cluster transformation (NCT) described in this paper. NCT attempts to increase the intercluster distances while preserving the dimensionality of the pattern vectors. NCT has no limitations in terms of dimensionality, number of classes, or the total number of patterns in the database.

We describe these two techniques, NDA and NCT, as preprocessing steps for analyzing binary mixture VOC data.

3. Nonparametric Discriminant Analysis (NDA)

Consider a multi-class classification problem and let C be the number of classes. For the i^{th} class, let $\{\mathbf{X}_i\}$ be the set of patterns in this class, $\mathbf{m_i}$ be the mean of vectors $\mathbf{x} \in \{\mathbf{X}_i\}$, n_i be the number of patterns in $\{\mathbf{X}_i\}$. Let $\mathbf{m}$ be the mean of all patterns in all C classes. Then the within scatter matrix $\mathbf{S_W}$, and between scatter matrix $\mathbf{S_B}$ are defined as follows:

$$\mathbf{S}_W = \sum_{i=1}^{C} \sum_{\mathbf{x}\in X_i} (\mathbf{x}-\mathbf{m}_i)\cdot(\mathbf{x}-\mathbf{m}_i)^T$$

$$\mathbf{S}_B = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{\mathbf{x}\in X_i} w_{ijx}(\mathbf{x}-\mathbf{m}_{ijx})\cdot(\mathbf{x}-\mathbf{m}_{ijx})^T \qquad (1)$$

where $\mathbf{m}_{ijx}$ represents the mean of $\mathbf{x_i}$'s k-nearest neighbors from class j and the w_{ijx} represents the weight of the feature vector $\mathbf{x}$ from class i to class j defined as

$$w_{ijx} = \frac{\min\left(\text{dist}\left(\mathbf{x}^i_{KNN}\right), \text{dist}\left(\mathbf{x}^j_{KNN}\right)\right)}{\text{dist}\left(\mathbf{x}^i_{KNN}\right)+\text{dist}\left(\mathbf{x}^j_{KNN}\right)} \qquad (2)$$

with dist($\mathbf{x}^i_{KNN}$) being the Euclidean distance from $\mathbf{x}$ to its k-nearest neighbors in class i. In general, if a point belonging to class i is far away in the feature space from the cluster of class j instances, w_{ijx} is a small quantity. If, however, an instance of class i is close to the boundary of class j instances, then w_{ijx} is a large quantity. We note that $\mathbf{S_W}$ is a measure of the intracluster distances, and $\mathbf{S_B}$ is a measure of the intercluster distances. The transformation, the projection from the original feature space onto a lower dimensional feature space, can be expressed as

$$\mathbf{y} = \mathbf{W}^T \cdot \mathbf{x} \qquad (3)$$

where the column vector $\mathbf{y}$ is the feature vector in the projected space corresponding to pattern $\mathbf{x}$. The optimum matrix $\mathbf{W}$ is obtained by maximizing criterion function

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T\mathbf{S}_B\mathbf{W}\right|}{\left|\mathbf{W}^T\mathbf{S}_W\mathbf{W}\right|} \qquad (4)$$

The columns of $\mathbf{W}$ that maximizes $J(\mathbf{W})$ are then the eigenvectors that correspond to the largest eigenvalues in the generalized eigenvalue equation [8]

$$\mathbf{S}_B\mathbf{w}_i = \lambda_i\mathbf{S}_W\mathbf{w}_i \qquad (5)$$

For nonsingular $\mathbf{S}_W$, Equation 5 can be written as

$$\mathbf{S}_W{}^{-1}\mathbf{S}_B\mathbf{w}_i = \lambda\mathbf{w}_i \qquad (6)$$

From Equation 6, we can directly compute the eigenvalues λ_i and the eigenvectors $\mathbf{w_i}$, constituting the columns of the $\mathbf{W}$ matrix, which can then be used to obtain the transformed instances $\mathbf{y}$ in the new feature space.

4. Nonlinear Cluster Transformation (NCT)

NCT is a three step procedure: in the first step, reduction of intracluster distances is achieved by eliminating the outliers. In the second step, the desired cluster separation is obtained by a simple translation of each cluster along an optimal direction. This step, in essence, generates training data pairs for determining the NCT function for the third step. In this last step, the data generated in step two is used to train a generalized regression neural network (GRNN) to approximate the function mapping between original clusters and the translated clusters. The feature vectors are then input to a classifier of choice. The details of these steps are explained below

4.1 Outlier Removal

The patterns in each class i in the training database are first normalized according to

$$\mathbf{x} = \mathbf{x} \Big/ \sqrt{\sum_{k=1}^{d}\left(x^k\right)^2} \qquad (7)$$

where x^k is the k^{th} element of the pattern $\mathbf{x}$, and d is the dimensionality of the patterns. Outlier removal is performed next, based on the Mahalanobis distances of patterns from the cluster centers. For each cluster i, the Mahalanobis distance of pattern $\mathbf{x}$ in class i is computed as

$$M_D = (\mathbf{x}-\mathbf{m}_i)^T\mathbf{C}_i{}^{-1}(\mathbf{x}-\mathbf{m}_i) \qquad \mathbf{x}\in\{\mathbf{X}_\text{i}\} \qquad (8)$$

where $\mathbf{C}_i$ is the covariance matrix of instances of the i^{th} class, and $\mathbf{m}_i$ is the mean of this population. M_D can be used as a measure of dispersion within the cluster. A suitable threshold is chosen based on the data, and instances with an M_D larger then this threshold are removed.

4.2 Cluster Translation

This step addresses the problem of closely packed and possibly overlapping clusters. The idea is to translate the clusters appropriately in order to physically separate them. Conceptually, all clusters are thought of as like charged particles, and the magnitude and direction of the translation vector are then derived using the concept of a repulsive force exerted by each cluster *i* on other clusters.

Consider a two-class problem with (possibly) overlapping clusters, whose centers are located at $\mathbf{m_1}$ and $\mathbf{m_2}$. The distance between these two clusters can be increased if class I patterns are translated along the vector $\mathbf{S}_1 = -\left(\mathbf{m}_2 - \mathbf{m}_1\right)$, and class II patterns are translated along $\mathbf{S}_2 = -\mathbf{S}_1 = \left(\mathbf{m}_2 - \mathbf{m}_1\right)$. This idea can be extended to multi-class problems of arbitrary dimensionality, where patterns of class C_i can be translated along $\mathbf{S_i}$, where the optimal direction $\mathbf{S_i}$ can be computed as

$$\mathbf{S_i} = -\sum_{j \neq i}^{C} \left(\mathbf{m}_j - \mathbf{m}_i\right) \tag{9}$$

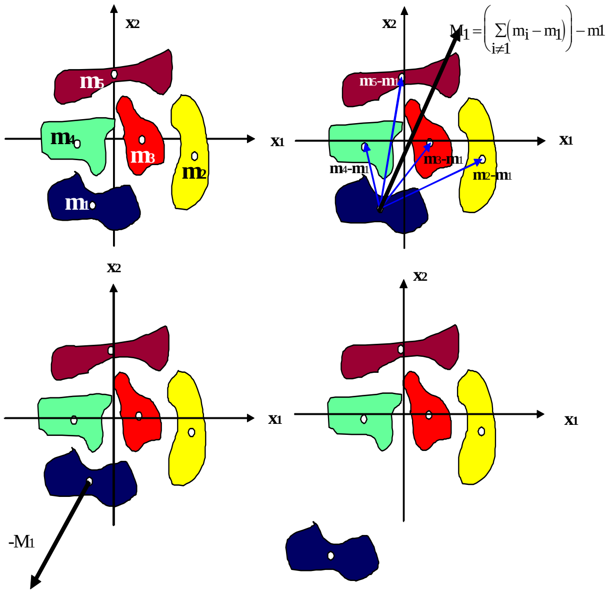
and where $\mathbf{m}_i$ and $\mathbf{m}_j$ are the cluster centers of cluster *i* and cluster *j*, respectively, and C is the number of clusters. The resultant translation vector for cluster *i* is $\mathbf{S_i} = -\mathbf{M_i}$, where

$$\mathbf{M_i} = \left(\sum_{j \neq 1} (\mathbf{m}_j - \mathbf{m}_i) \right) \tag{10}$$

All patterns in cluster *i* are moved along the direction of -$\mathbf{M_i}$, and the translated patterns can be obtained by

$$\mathbf{x}_{Si} = \mathbf{x}_i + \left(-\mathbf{M_i} / \left\| \mathbf{M_i} \right\|\right) \cdot dist_i \tag{11}$$

where $\mathbf{x}_i$ is a pattern from cluster *i*, $dist_i = 1 / \left| \mathbf{m} - \mathbf{m}_i \right|$ is a normalizing constant that controls the amount of translation, and $\mathbf{x}_{Si}$ is the new location of the pattern $\mathbf{x}_i$. It is straightforward to show mathematically that these translation directions maximize intercluster distances [10]. Note that $\mathbf{S}_i$ points in the opposite direction of the resultant vector that combines the cluster center of cluster *i* to the centers of all other clusters, that is, it points away from all other clusters. The procedure is illustrated in Figure 1.



x2
m5
m4
m3
m2
m1
x1
x2
$M_1 = \left(\sum_{i \neq 1} \left(m_i - m_1\right) \right) - m1$
m5-m1
m4-m1
m3-m1
m2-m1
x1
x2
x1
-M1
x2
x1

Figure 1. Nonlinear cluster transformation.

The cluster transformation described here can also be expressed in a matrix form. Let $i=1,2,\ldots,C$, where C is the number of classes, $n=1,2,\ldots,N_i$ where N_i is the number of patterns in class *i*, $X_n^{\,i}$ be the n^{th} pattern of the i^{th} class, and $Y_n^{\,i}$ be the corresponding pattern after translation. Then,

$$\begin{pmatrix} Y_1^{\,i} \\ Y_2^{\,i} \\ . \\ . \\ Y_{Ni}^{\,i} \end{pmatrix} = -dist_i \cdot \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \\ & \vdots & \\ & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{m}_1 - \mathbf{m}_i \\ \mathbf{m}_2 - \mathbf{m}_i \\ . \\ . \\ \mathbf{m}_C - \mathbf{m}_i \end{pmatrix} + \begin{pmatrix} X_1^{\,i} \\ X_1^{\,i} \\ . \\ . \\ X_{Ni1}^{\,i} \end{pmatrix} \tag{12}$$

This equation can be implemented on the training data sets to generate a second dataset that can be used to train a GRNN to learn the overall transformation function.

4.3 Function Mapping

In order to translate each cluster away from each other, the cluster information is required, which obviously is not available for a test pattern. We therefore need to *learn* how to translate patterns without knowing the class information. This problem can be thought of as a function approximation problem, where the function to be approximated is a function that maps *d* dimensional original patterns to their new locations. A generalized regression neural network (GRNN) was used to accomplish this function approximation. GRNNs can be though of as a special case of radial basis function neural networks (RBFNN). GRNNs do not require iterative training, and they can approximate any arbitrary multidimensional function defined between a set of input and output vectors. GRNN is based on the theory of nonlinear regression analysis, commonly used as a statistical function estimation scheme. For brevity and due to their widespread use, GRNN architecture is not reviewed here and interested readers are referred to [11].

5 Results

We have evaluated NDA and NCT in the binary mixture VOC identification problem, a real world example, which proved to be intractable for neural network classifier without the preprocessing. The database used in this study contained 24 binary mixtures of VOCs. Each mixture contained two of the following VOCs: Octane (OC), acetonitrile (ACN) xylene (XL), ethanol (ET), toluene (TL), trichloroethane (TCA), trichloroethylene (TCE), methylethylketone (MEK), and hexane (HX). Six sensors, each coated with a different polymer, were exposed to these mixtures at all combinations of 150, 300, 500 and 700 parts per million (ppm), giving 16 combinations of concentrations for each mixture. The polymers were Apiezon (APZ), Polyisobutelene (PIB), Poly (diethyleneglycoladipate)(DEGA),Solgel (SG), Siloxane (OV275), and Poly (diphenoxylphoshate) (PDPP). Figure 2 shows typical patterns obtained from four mixtures of XL, where the vertical axis shows relative frequency change of the QCM sensor. We note the similarity of the patterns for different mixtures. The main reasons of this similarity are the dominance of XL compared to others, as well as the overlapping nature of instances in the feature space.

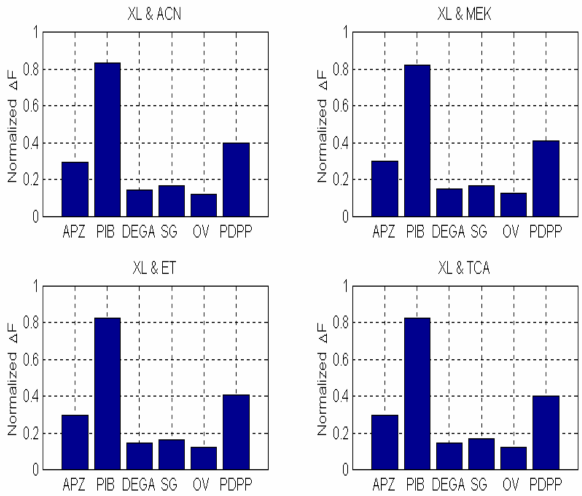


Figure 2. Responses of four mixtures of XL.

When a dominant VOC is present in the mixture, the responses of sensors to other VOCs become partially, or completely, masked by the response to the dominant VOC. XL, TL, ET, TCE and OC are known to be dominant and hence the main goal was to identify these VOCs first. We were unable to find any NN architecture to distinguish these signals according to their dominant VOCs. Though once the dominant VOC was identified, an MLP was able to identify the secondary VOCs).

The sample patterns shown in Figure 2 were not necessarily the worst, but rather typical patterns, illustrating the difficulty of this database. The entire database consisted of 384 six-dimensional patterns. For dominant VOC identification, there were five classes, namely, OC, ET, XL, TL and TCE. The database was partitioned into two parts, T_{VOC} for training and E_{VOC} for evaluation, each with 192 instances. The training database T_{VOC} and its preprocessed version (with NDA or NCT) were used to train the GRNN to learn the function mapping. GRNN was then evaluated on E_{VOC}, which was not seen during training. Due to functional similarities between GRNN and RBF networks, an RBF network was also tried in place of a GRNN. While unable to converge with unprocessed data, an MLP type NN was easily trained with the preprocessed data. Once the dominant VOC was identified, a separate MLP was trained for each of the five secondary VOCs, creating a two-tier classification system. The generalization performances on the test data for each technique are provided in Table 1.

Algorithm	Dominant	Secondary
NDA	86.84%	81.58%
NCT-GRNN	86.40%	73.24%
NCT-RBF	83.55%	69.89%

Table 1. Generalization performances

6. Conclusions and Discussion

Two pattern separability techniques, NDA and NCT, have been applied to mixture VOC data. NDA aims to maximize the intercluster to intracluster distance ratios, whereas NCT tries to increase the intercluster distances while keeping intracluster distances constant. Preprocessing allowed improved performances of subsequent classification algorithms, and in fact, made training possible for the VOC database.

For identification of binary mixtures of VOCs, where the response to a secondary VOC is masked by the response to a dominant VOC, we proposed a two-tiered classification procedure. The dominant VOC is identified first, and based on this information a second level of classifiers are used – one for each dominant VOC – to identify the secondary VOCs. Attempting to identify both components at once had earlier proven to be impossible.

Both NDA and NCT had similar performances for the identification of dominant VOCs, but NDA performed better for the identification of secondary VOCs. Furthermore, using GRNN with NCT performed significantly better then using RBF networks

Acknowledgments

This work is supported by the New Jersey Water Resources Research Institute at Rutgers University, NJ.

References

[1] T.C.Pearce, *et al.*, *Handbook of Machine Olfaction, Electronic Nose Technology*. Wiley 2003.

[2] A. K. Srivastava, "Detection of volatile organic compounds (VOCs) using SnO_2 gas-sensor array and artificial neural network," *Sen. and Act. B,* vol. 96, no. 1-2, pp. 24-37, 2003.

[3] M. Sriyudthsak, *et al.*"Radial basis neural networks for identification of volatile organic compounds," *Sen. and Act. B*, vol. 65, no. 1, pp. 358-360, 2000.

[4] R. Polikar, *et al.*, "Detection and identification of odorants using an electronic nose," in *IEEE Int. Con. Acou., Speech and Sig. Proc.*, pp. 3137-3140, 2001.

[5] A. Ortega, S. Marco, T. Sundic, and J. Samitier, "New pattern recognition systems designed for electronic noses," *Sensors and Actuators, B: Chemical*, vol. 69, no. 3, pp. 302-307, 2000.

[6] M. Penza and G. Cassano, "Application of PCA and ANNs to recognize the individual VOCs of methanol/2-propanol in a binary mixture by SAW multi-sensor array," *Sen. and Act. B,*, vol. 89, no. 3, pp. 269-284, 2003.

[7] C. Di Natale, *et al.*, "Composed neural network for the recognition of gas mixtures," *Sens. and Act. B*, vol. B25, no. 1-3 pt 2, pp. 808-812, 1995.

[8] R.O.Duda, P.E.Hart, and D.G.Stork, *Pattern Classification*, 2 ed. New York, NY: Wiley, 2001.

[9] K.Fukunaga, *Statistical Pattern Recognition*, 2 ed. New York, NY: Academic Press, 1990.

[10] R. Polikar, L. Udpa, and S. S. Udpa, "Nonlinear cluster transformations for increasing pattern separability," in *IEEE Int. J. Con. Neural Networks*, pp. 4006-4011, 1999.

[11] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568-576, 1991.