DETECTION AND IDENTIFICATION OF ODORANTS USING AN ELECTRONIC NOSE

Robi Polikar[1] , *Ruth Shinar*[2], *Vasant Honavar*[3], *Lalita Udpa*[4], *Marc D. Porter*[2]

[1] Dept. of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028
[2]Microanalytical Instrumentation Center, [3] Computer Science [4]Electrical and Computer Engineering
Iowa State University, Ames, IA 50011

ABSTRACT

Gas sensing systems for detection and identification of odorant molecules are of crucial importance in an increasing number of applications. Such applications include environmental monitoring, food quality assessment, airport security, and detection of hazardous gases. In this paper, we describe a gas sensing system for detecting and identifying volatile organic compounds (VOCs), and discuss the unique problems associated with the separability of signal patterns obtained by using such a system. We then present solutions for enhancing the separability of VOC patterns to enable classification. A new incremental learning algorithm that allows new odorants to be learned is also introduced.

1. INTRODUCTION

Gas sensing systems for detection and identification of odorants are of significant importance for many industries and organizations. Examples include food industries for testing the quality or wholesomeness of food products, military and humanitarian organizations for locating buried land mines, petrochemical and valve manufacturing companies for detecting and identifying hazardous gases, and airport security and customs inspection agencies for detecting illegal drugs and plastic bombs. Due to their ability to mimic the human olfactory system, albeit in a very limited sense, gas sensing systems are often referred to as *Electronic Nose (Enose) Systems*.

Enose systems for detection and identification of volatile organic compounds (VOCs), an important class of chemicals that can readily evaporate at room temperature, have gained considerable attention, since VOCs are encountered in many gas sensing applications. A major problem in VOC identification is the substantial similarity of patterns obtained for different VOCs, a phenomenon attributed to low selectivity of the sensing system. Most attempts to solve this problem have provided only marginal success [1], and only for specific VOCs. Furthermore, no attempt has been made to develop an algorithm to incrementally learn new odorants.

In this paper we describe a gas sensing system along with new pattern classification and incremental learning algorithms for detection and identification of VOCs. An experimental setup that can be used for various gas sensing applications is first described. Unique challenges that are encountered in processing and identifying signals obtained by using such systems are then presented, followed by an intuitive and powerful pattern separability enhancing algorithm to address these challenges. An incremental learning algorithm is then introduced, which allows the system to identify additional VOCs that have not been previously encountered.

Robi Polikar was with the Dept. of Electrical and Computer Engineering of Iowa State University while conducting this study.

2. EXPERIMENTAL SETUP FOR VOC DETECTION

Piezoelectric acoustic wave sensors comprise a versatile class of chemical sensors for the detection of VOCs. Addition or subtraction of molecular material from the surface or bulk of an acoustic wave sensor results in a change in its resonant frequency. The frequency change, Δf, caused by a deposited mass, Δm, can be described by the Sauerbrey Equation [2]. For quartz crystal microbalances (QCMs), this relationship is given by

$$\Delta f = -2.3 \times 10^6 \cdot f^2 \cdot \frac{\Delta m}{A} \qquad (1)$$

where f is the fundamental resonant frequency of the bare crystal, and A is the sensing surface area. For sensing applications, a sensitive polymer film is cast on the surface of the QCM. This film can bind the molecules of the VOC of interest, altering the resonant frequency of the device in proportion to the added mass. The QCM-based chemical sensor system typically consists of an array of several crystals, each coated with a different polymer film. The response pattern of such an array then serves as the signature for a given VOC. This array design is aimed at improving identification, which is hampered by the limited selectivity and sensitivity of individual films.

An array of six 9 MHz QCMs was used in this study. The QCMs were first coated with chromium/gold, which served as electrodes. Each QCM was then coated with a different polymer to sorb the VOCs of interest. The QCMs were mounted in a sealed test fixture and exposed to VOC vapors. The vapor generation system consisted of calibrated mass flow controllers, conventional gas bubblers containing the VOCs, and a pair of three-way switchable valves leading into the test fixture. The vapor at various concentrations was generated by flowing a carrier gas, typically dry nitrogen, through the bubbler and further diluting the vapor with nitrogen to obtain the desired concentration. The switchable valves were computer controlled to automatically expose the sensor array to various concentrations of VOCs. The frequency response was monitored using an HP8753C network analyzer, interfaced to an IEEE 488 card installed in the PC, and an HP8516A resonator measurement software. Real time data were displayed and analyzed to obtain frequency shifts (relative to the baseline) vs. VOC concentration. Typical noise levels (standard deviations of the baseline) for the QCMs were around 0.1 Hz. Figure 1 depicts the overall schematic of this setup.

Sensors were exposed to five VOCs, namely toluene (TL), xylene (XL), ethanol (ET), octane (OC) and tricholoroethylene (TCE) at concentrations of 70, 105, 140, 175, 245, 350 and 700 parts per million, in random order, for a duration of 10 minutes each. After each VOC exposure, the test fixture was purged with

0-7803-7041-4/01/$10.00 ©2001 IEEE

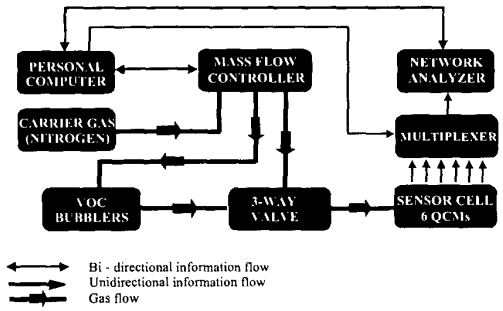

Fig. 1. Experimental setup for the Enose system.

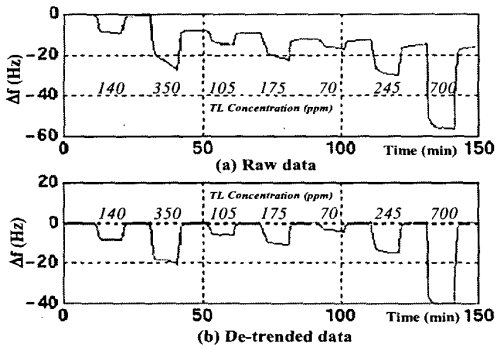

Fig. 2. (a) Raw data (b) Detrended data.

dry nitrogen to flush the VOC molecules. At each concentration, the frequency shift of each sensor was recorded to obtain a six dimensional pattern, representing the exposed VOC at that concentration. Figure 2 shows a typical response to these seven concentrations of toluene from a single sensor.

3. ISSUES ASSOCIATED WITH ODORANT PATTERNS

Several issues are associated with patterns obtained from these systems. First, as seen from Figure 2(a), sensor responses often exhibit a drift that needs to be corrected. Therefore, a drift removal algorithm was first applied, which segments the data, computes the best linear fit in the least mean square sense for each segment, and then subtracts this fit from the original signal. Figure 2(b) illustrates the output of this simple detrending scheme.

Sensor response amplitudes are linearly proportional to the VOC concentration, where the proportionality constant defines the sensitivity of the sensor for the given VOC. Since the concentration for an unknown VOC is also unknown, the identification must be based on signature patterns, and not on the concentration dependent amplitudes. Therefore the concentration information was removed by normalizing each pattern by the square root of the sum of squares of sensor responses. However, this normalization removes most of the discriminatory information, as shown in Figure 3, which illustrates responses of six sensors to toluene and xylene before and after the normalization for a given concentration. Additional information on the setup, the coatings, the database, and comparison of the Enose to mammalian olfactory system can be found in [3, 4].

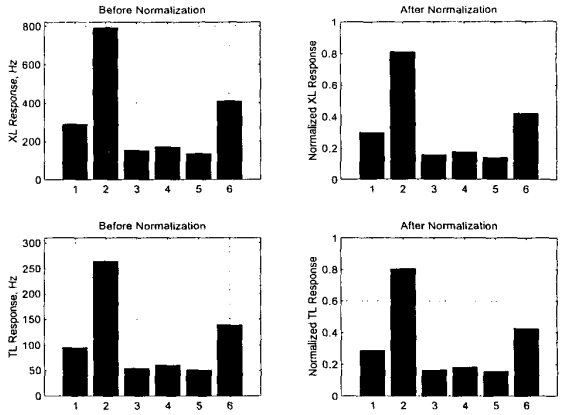

Fig. 3. Effect of normalization.

A number of classification algorithms, including neural networks, decision trees, and cluster analysis have been employed to identify VOCs using their normalized sensor responses, however none of these methods converged to a solution. This is attributed to the poor selectivity of the sensors, which resulted in overlapping class distributions in the pattern space. Therefore, a preprocessing algorithm that can identify and augment the minor discriminatory information between patterns of different classes (VOCs) is necessary. Such a scheme is introduced in the next section.

4. ENHANCING PATTERN SEPARABILITY

Many of the existing schemes for enhancing pattern separability do not specifically target increasing intercluster distances, but rather try to obtain the smallest set of features with the most discriminatory information, through a mathematical transformation or a set of rules. One method that specifically targets increasing pattern separability is Fisher's linear discriminant method, which also reduces the dimensionality to $C-1$, where C is the number of classes [5]. This mandatory reduction in dimensionality, however, can work against pattern separability, since there may not be enough discriminatory information left in $C-1$ features.

In this paper, we propose a new scheme where enhancing pattern separability is achieved through *nonlinear cluster transformation (NCT)*, a three-step supervised procedure that attempts to increase the intercluster distances and reduce the intracluster distances, while preserving the dimensionality.

In the first step, reduction of intracluster distances is achieved by eliminating the outliers, using the Mahalanobis distance metric. In the second step, the desired cluster separation is obtained by translation of each cluster along an optimal direction, away from all other clusters. This step, generates training data pairs for determining the NCT function. In the third step, the data generated in the second step is used to train a generalized regression neural network (GRNN) to approximate the function mapping between original and translated clusters.

The cluster translation step addresses the problem of closely packed and possibly overlapping clusters. The underlying idea is to move clusters away from each other in order to physically separate them. 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 $\mathbf{S}_1 = -(\mathbf{m}_2 - \mathbf{m}_1)$, and class II patterns are translated along $\mathbf{S}_2 = -\mathbf{S}_1 = -(\mathbf{m}_1 - \mathbf{m}_2)$. This idea can be extended to multi-class problems of arbitrary dimensionality, where patterns of class C_i can be translated along $\mathbf{S}_i$. The optimal direction $\mathbf{S}_i$ can be computed as

$$\mathbf{S}_i = -\sum_{j \neq i}^{C} (\mathbf{m}_j - \mathbf{m}_i) \tag{2}$$

where C is the total number of clusters. All patterns in cluster i are moved along the direction of $\mathbf{S}_i$, and the translated patterns can then be obtained as $\tilde{\mathbf{X}}_i = \mathbf{x}_i + (\mathbf{S}_i / \|\mathbf{S}_i\|) \cdot dist_i$, where $\tilde{\mathbf{X}}_i$ is the new location of pattern $\mathbf{x}_i$, and $dist_i = 1/|\mathbf{m} - \mathbf{m}_i|$ is a normalizing constant that controls the amount of translation.

It can be shown that the directions of these translation vectors are optimal [3], since these directions maximize the overall intercluster distance D, defined as the summation of intercluster distances between all cluster pairs:

$$D = \sum_{i,j=1}^{C} D_{ij} = \sum_{i,j=1}^{C} (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i - \mathbf{m}_j) \tag{3}$$

However, in order to translate a test pattern, we need to learn how to translate patterns without knowing to which class they belong. This problem can be thought of as a function approximation problem, where the function to be approximated is the one that maps original patterns to their new locations. A GRNN was used to accomplish this function approximation. GRNNs, special cases of radial basis function (RBF) neural networks, have been used with significant success in multidimensional function approximation. GRNNs do not require iterative training, and they can approximate any arbitrary multidimensional function defined between a set of input and output vectors. Detailed information on the use of GRNNs can be found in [3, 6].

Figure 4 illustrates the effect of NCT on blind data which was not used for training. Only three sensor responses were used in Figure 4 for easy visualization, whereas computations were made in six dimensions. Note that patterns corresponding to different VOCs are very closely packed and overlapping before processing, and they are separated considerably after the NCT processing. Once the patterns have been preprocessed, the complexity of the classifier can be significantly reduced. In fact, a single layer multilayer perceptron (MLP) trained with 220 patterns corresponding to five VOCs at various concentrations, was able to correctly classify 92% of 164 test patterns which were not used during training. Recall that no classifier, including single or double hidden layer MLPs, RBFs, or decision trees, was able to converge to a solution, let alone correctly classify the majority of the test patterns when trained with unprocessed signals.

5. INCREMENTAL LEARNING OF VOC PATTERNS

One of the main challenges in using Enose systems is to be able to increase the number of odorants that can be identified over time. From a pattern classification point of view, this requires an algorithm that is capable of incremental learning of new classes, without forgetting the previously acquired knowledge. Furthermore, the training database that was originally used to train the system, may not be available by the time new training datasets become available. Therefore, the algorithm should not require access to previously used databases when learning new information. Commonly used classification algorithms such as MLPs, RBFs, or

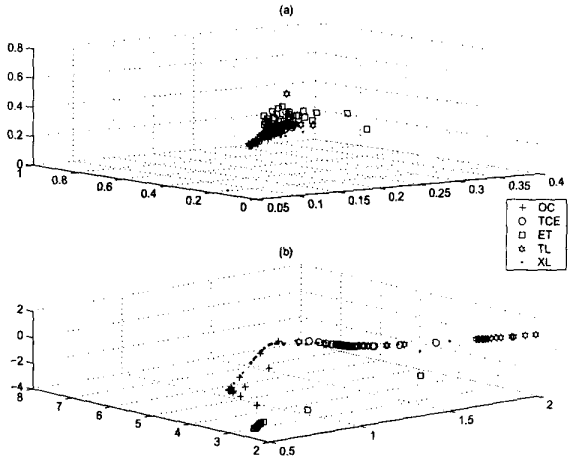

Fig. 4. (a) Before (b) after NCT processing.

wavelet networks are not capable of incremental learning, since they need to be reinitialized and retrained with the combined old and new data to learn the additional information. This causes all previously acquired knowledge to be lost, a phenomenon known as *catastrophic forgetting*.

LEARN++, which we first introduced in ICASSP 2000 [7], is an algorithm that addresses all of the issues mentioned above. In this section, we summarize the major points of LEARN++, and introduce new features that improve classification performance. The details and theoretical analysis of LEARN++ can be found in [3].

LEARN++ is based on generating a number of classifiers using different distributions of the training data, and then combining the outputs of these classifiers using a weighted majority voting scheme. The algorithm keeps track of the performance of each classifier on each training instance, and generates a new training subset based on the performances of all previous classifiers. In particular, a weight is given to each instance, and this weight is increased if the instance is misclassified. The updated weight is then used to determine whether this instance should be included in the next training set. A distribution is formed from these weights according to which the next training set is chosen. Misclassified instances are more likely to be selected into the next training set. A new classifier is then trained with the new training set, added to the pool of classifiers generated earlier, and the combined classification performance of all classifiers is then used to determine the next training set. Multiple classifiers are generated for any given database, and as new databases become available, new classifiers are added. This scheme ensures that instances, particularly from new classes, are learned efficiently, since these instances are most likely to be misclassified by previous classifiers. Furthermore, since all classifiers are retained, previously learned information is not lost. The final classification for each instance is then based on the weighted majority voting of all classifiers.

For each training instance (x_i, y_i), the weight distribution update rule from iteration t to $t+1$ is given by

$$D_{t+1} = \frac{D_t}{\sum_i D_t(i)} \cdot \begin{cases} B_t, & \text{if } H_t(x_i) = y_i, \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

where $H_t(x_i)$ is the *composite classifier*, computed by the weighted majority voting of all previous t classifiers for the instance x_i, D_t is the weight distribution at the t^{th} iteration, y_i is the desired clas-

sification, and $0 < B_t < 1$ is percent error of misclassification of the t^{th} classifier. For given t classifiers, the weighted majority voting simply computes the class that receives the highest vote, from voters (individual classifiers) whose votes are weighted according to their individual performance. That is,

$$H_t = \arg\max_y \sum_{k=1}^{K} \sum_{t:H_t(x)=y} \log(1/B_t) \qquad (5)$$

where K is the total number of databases used to train t classifiers.

Alternatively, the inverse of the Mahalanobis distance between each instance and the t^{th} training dataset can also be used as the weight of each classifier during voting. Instances with small Mahalanobis distances are likely to come from the database which was used to train the current classifier, and hence that classifier is more likely to classify this instance correctly. Using the inverse of the Mahalanobis distance to assign a different weight to each classifier, ensures that the classifier weights are dynamically updated, which results in a better estimate of the optimal classifier weight for each instance [3]. The disadvantage of this approach is that it requires the mean and covariance matrices of the training datasets to be saved, increasing the space complexity of the algorithm. To use a Mahalanobis distance based voting scheme, the term B_t in Equation 5 is replaced by

$$\frac{1}{MW_{\mathbf{x}_i}(t)} = \frac{1}{\left((\mathbf{x}_i - \mathbf{m}_t)^T \cdot \mathbf{C}_t^{-1} \cdot (\mathbf{x}_i - \mathbf{m}_t)\right)} \qquad (6)$$

where $1/MW_{\mathbf{x}_i}(t)$ is the weight of the t^{th} classifier for instance x_i, and $\mathbf{m}_t$ and $\mathbf{C}_t$ are the mean and covariance matrix of the instances in the t^{th} training dataset.

6. RESULTS AND DISCUSSION

In this section we present the results of applying NCT preprocessing followed by LEARN++ classification of VOC patterns in an incremental manner. The database consisted of 384 six-dimensional signals, 220 of which were used for training. This database was divided into three training datasets $S_1 \sim S_3$ and one test dataset, $TEST$. S_1 had instances from ET, OC, and TL, S_2 had instances mainly from TCE (and very few from the previous three), and S_3 had instances from XL (and very few from the previous four). $TEST$ set included instances from all classes. A single hidden layer MLP was used as the base classifier. Note however that LEARN++ is independent of the classifier, and can be used with any supervised learning algorithm. Table 1 presents the data distribution, and Table 2 presents the results.

Each column in Table 2 shows the performance of the composite classifier obtained by computing the Mahalanobis weighted majority of all classifiers generated up to that point. Each row shows the performance on a particular dataset. During training session one, only S_1 was used for training (5 classifiers), during session two, only S_2 was used for training (10 classifiers), and so on. As expected, the performances of the classifiers on their own training data were very high. We note that the performance on the $TEST$ dataset improves as incremental learning progresses and the system learns new classes. This is also expected, since $TEST$ set had instances from all five classes, and instances from all classes were not introduced to classifiers until the last session. We note that when the last five classifiers (which have seen instances from all classes) were evaluated on test dataset, the performance was around 60%, indicating that all training sessions

Table 1. Data class distribution for the VOC database.

	ET	TCE	OC	XL	TL
S_1	20	-	10	-	40
S_2	10	25	10	-	10
S_3	10	15	10	40	10
$TEST$	24	24	24	40	52

Table 2. Classification results on the VOC database.

Session → Dataset ↓	Training 1 (5)	Training 2 (10)	Training 3 (5)
S_1	98.8 %	86.3 %	75.0 %
S_2	-	89.9 %	90.1 %
S_3	-	-	94.1 %
$TEST$	56.7 %	64.0 %	86.6 %

are indeed necessary for the final classification. The performance improvement on the $TEST$ data as new datasets are introduced demonstrates the incremental learning capability of the algorithm.

7. CONCLUDING REMARKS

In this paper we introduced an electronic nose system for odor detection and identification along with various signal processing and classification algorithms. In particular, nonlinear cluster translation was introduced for increasing pattern separability, and LEARN++ for incremental learning. We note that both algorithms were also tested for identification of a larger number of VOCs, individual components of mixtures of VOCs, as well as non gas-sensing applications, and very promising results were obtained [3]. This demonstrates the effectiveness and feasibility of the algorithms for a broad spectrum of pattern separability and classification problems. Current work is in progress, where the voting mechanism of LEARN++ is used to estimate the reliability of the final classification and the confidence limits of the performance.

8. REFERENCES

[1] B. Kermani, S. Schiffman, H. Nagle, "Using neural networks and genetic algorithms to enhance performance in an electronic nose," *IEEE Tran. on Biomedical Eng.*, vol. 46, no. 4, pp. 429-439, 1999.

[2] G. Z. Sauerbrey, "Use of vibrating quartz for thin film weighting and microweighing (in German)," *Zeitchrift für Physik*, vol. 155, pp. 206-222, 1959.

[3] R. Polikar, "Algorithms for enhancing pattern separability, feature selection and incremental learning with applications to gas sensing electronic nose systems" *Ph.D. Dissertation*, Iowa State University, August 2000. Also available at www.public.iastate.edu/~rpolikar/RESEARCH/PhDdis.pdf

[4] www.public.iastate.edu/~rpolikar/RESEARCH/vocdata.html

[5] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, John Wiley & Sons: New York, NY, 2000.

[6] D.F. Specht, "A general regression neural network," *IEEE Trans. on Neural Networks*, vol.2 pp. 568-576, 1991.

[7] R. Polikar, L. Udpa, S. Udpa, V. Honavar, "LEARN++: An incremental learnig algorithm for multilayer perceptron neural networks," *Proceedings of ICASSP 2000*, vol. 6, pp. 3414-3417, 2000.