Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 26 (2005) 699–706
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Gas identification using density models

Sofiane Brahim-Belhouari *, Amine Bermak

Hong Kong University of Science and Technology, EEE Department, Clear Water Bay, Kowloon, Hong Kong

Received 18 July 2003; received in revised form 28 July 2004
Available online 2 November 2004

Abstract

In this paper we compare the accuracy of a range of advanced density models for gas identification from sensor array signals. Density estimation is applied in the construction of classifiers through the use of Bayes rule. Experiments on real sensors' data proved the effectiveness of the approach with an excellent classification performance. We compare the classification accuracy of four density models, Gaussian mixture models, Generative topographic mapping, Probabilistic PCA mixture and K nearest neighbors. On our gas sensors data, the best performance was achieved by Gaussian mixture models.

Keywords: Classification; Gas sensor array; Mixture models; Pattern recognition

1. Introduction

There is an urgent need to develop sensors and systems that can selectively detect and determine various kinds of combustible gases. In fact, the ability to monitor and precisely measure leakages of combustible and explosive gases is crucial in preventing the occurrence of accidental explosions. In order to detect combustible gases, different kinds of sensors have been used. Unfortunately, present gas sensors suffer from a number of shortcomings such as non-selectivity, non-linearities of the sensor's response and long-term drift.

While waiting for the development of a new generation of sensors, strategies and algorithms of pattern recognition must be developed in order to overcome the drawbacks of the actual sensors. A gas sensor array permits to improve the selectivity of the single gas sensor, and shows the ability to classify different odors. In fact, an array of different gas sensors is used to generate a unique signature for each odor. The feature vector that results from the preprocessing stage is often not suitable to be processed by a classification algorithm due its high dimensionality and redundancy. Therefore,

* Corresponding author. Tel.: +852 2358 8844; fax: +852 2358 1485.
E-mail addresses: eebelhou@ust.hk (S. Brahim-Belhouari), eebermak@ust.hk (A. Bermak).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.09.020

a dimensionality reduction stage is required in most cases, either by feature extraction or feature selection. The resulting low-dimensional feature vector is used to solve a given classification problem, which consists of identifying an unknown sample as one from a set of previously learned gases.

Significant work has been devoted to design a successful pattern analysis system for machine olfaction (Gutierrez-Osuna, 2002). Various kinds of flexible pattern recognition algorithms have been used for classifying chemical sensor data. Most notably neural networks have been exploited, in particular multi-layer perceptrons (MLP), radial basis functions (RBF) and self-organizing maps (SOM) (Gutierrez-Osuna, 2002). Other methods based on the class-conditional density estimation have been used, such as quadratic and K nearest neighbors (KNN) classifiers. These parametric and non-parametric density estimation methods have their merits and limitations. An attractive approach, which achieves the best of both methods, has been proposed. It is a semi parametric method based on mixture distributions. By defining a very general class of functional forms, they can model, with enough components, any density function (McLachalan and Basford, 1988). Mixture models have been applied with a significant success to speech recognition (Zhang and Alder, 1994) and image retrieval (Vasconcelos and Lippman, 2000), generating a common framework for dealing with image and sound.

In this paper we present a gas classification approach based on class-conditional density estimation using different density models, which has been successfully applied to gas identification using an experimentally obtained dataset. We compare the performance of four density models operating on PCA, LDA and Neuroscale projections. As illustrated in Fig. 1, PCA, LDA and Neuroscale are used as feature extractor while the four density models are used as gas classifier. Our aim is the identification of combustion gases with an array of eight micro-electronic gas sensors. Gaussian mixture model is shown to outperform KNN, probabilistic PCA mixture and generative topographic mapping classifiers. PCA projection is shown to improve considerably the classification accuracy.

The paper is organized as follows: Section 2 presents feature extraction techniques used in this study. Section 3 describes the different density models to build a probabilistic classifier. Section 4 shows the feasibility of the approach for a real gas identification problem. Section 5 presents some concluding remarks.

2. Feature extraction

The recognition procedure involves preprocessing and dimensionality reduction, learning from data and validation stage. In order to avoid problems associated with high dimensionality and redundancy, the initial feature vector is projected

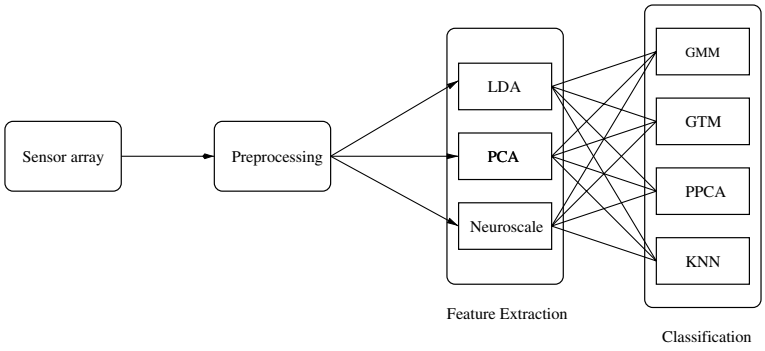

Fig. 1. Scheme of the gas identification system. The performances of four different density models operating on PCA, LDA and Neuroscale (total of 12 classification procedures) are compared.

onto a lower dimensional space. The goal of feature extraction is to find a low-dimensional vector $\boldsymbol{z}$ that preserves most of the information in the original feature vector $\boldsymbol{x}$:

$$\boldsymbol{f} : \boldsymbol{x} \in R^d \rightarrow \boldsymbol{z} \in R^q (q < d) \tag{1}$$

Hence, a projection method is needed for mapping data to a lower dimension space. The aim of such technique is to determine $\boldsymbol{f}$ which optimizes a given criterion. In the case of dimensionality reduction techniques belonging to the linear transformation family, the mapping is given by

$$\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x} \tag{2}$$

Most feature extraction techniques for electronic nose applications are based on linear techniques, mainly Principal Components Analysis (PCA) and Fisher's Linear Discriminant Analysis (LDA).

2.1. Principal components analysis

PCA is a linear transformation that preserves as much data variance as possible. PCA chooses $\boldsymbol{T}$ that minimizes the mean squared distance between original data and those reconstructed from reduced data. It is shown that:

$$\boldsymbol{T}_{\mathrm{PCA}} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1/2} \tag{3}$$

where $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ are, respectively, the eigenvectors matrix and the diagonal eigenvalues matrix of the data covariance matrix.

2.2. Linear discriminant analysis

LDA provides a linear projection of the data with $(c-1)$ dimensions, by taking into account the scatter of data within each class and across classes. Projection directions are those that maximize the inter-class separation of the projected data. The LDA transformation matrix is given by

$$\boldsymbol{T}_{\mathrm{LDA}} = \boldsymbol{S}_w \boldsymbol{\Lambda}_w^{-1/2} \boldsymbol{S}_B \tag{4}$$

where $\boldsymbol{S}_w$ and $\boldsymbol{\Lambda}_w$ are, respectively, the eigenvectors matrix and the diagonal eigenvalues matrix of the within-class scatter $\boldsymbol{W}$. $\boldsymbol{S}_B$ is the eigenvectors matrix of the between class scatter $\boldsymbol{B}$.

2.3. Neuroscale

Neuroscale is a non-linear topography (i.e. distance preserving) projection method. The concept of data topography is assumed to be captured by the inter-point distances, usually measured with an Euclidean metric:

$$D_{ij}^* = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \tag{5}$$

Each data point $\boldsymbol{x}_i$ is projected by Radial Basis Function (RBF) to a point $\boldsymbol{z}_i$, which minimizes the Sammon stress metric (Lowe and Tipping, 1996):

$$E = \sum_{i=1}^{N} \sum_{j>i}^{N} (D_{ij} - D_{ij}^*) \tag{6}$$

where the distance between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is denoted by D_{ij}.

The points $\boldsymbol{z}$ are generated by the RBF given the data point inputs. Thus, $\boldsymbol{z}_i = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{W})$, where $\boldsymbol{f}$ is the non-linear transformation effected by the RBF with parameters (weights) $\boldsymbol{W}$. We use a non-linear optimization algorithm to find the optimal weights minimizing the Sammon stress metric. This method presents the advantage of preserving the data structure, as well as the possibility of incorporating subjective information. In fact, one useful change can be made to the distance measure in order to generate a feature space that separates classes. For example, if each data point $\boldsymbol{x}_i$ belongs to a known class C_i, a dissimilarity measure s_{ij} can be defined by: $s_{ij} = 1$ if $C_i \neq C_j$ and 0 elsewhere. This can then be incorporated into the distance measure:

$$\delta_{ij} = (1 - \alpha) D_{ij}^* + \alpha s_{ij} \tag{7}$$

where the parameter $\alpha \in [0, 1]$ controls the degree of supervisory information in the mapping. Intermediate values of α allow the points to be projected so as to retain the distance structure with extra separation from the classes.

3. Density models

The objective of pattern recognition is to set a decision rule, which optimally partitions the data

space into c regions, one for each class C_k. The boundaries between regions are the separating surfaces or decision boundaries. A pattern classifier generates a class label for an unknown feature vector $\boldsymbol{x} \in R^d$ from a discrete set of previously learned classes. The most general classification approach is to use the posterior probability of class membership $\wp(C_k|\boldsymbol{x})$. To minimize the probability of misclassification one should consider the maximum a posterior rule and assign $\boldsymbol{x}$ to class $C_{\hat{k}}$:

$$\begin{aligned} C_{\hat{k}} &= \arg\max_{\{1,\ldots,c\}} [\wp(C_k|\boldsymbol{x})] \\ &= \arg\max_{\{1,\ldots,c\}} [\wp(\boldsymbol{x}|C_k)\wp(C_k)] \end{aligned} \tag{8}$$

where $\wp(\boldsymbol{x}|C_k)$ is the class-conditional density and $\wp(C_k)$ is the prior probability. In the absence of prior knowledge, $\wp(C_k)$ can be approximated by the relative frequency of examples in the dataset. One way to build a classifier is to develop a model that estimates the posterior probabilities directly, where the boundaries are learnt from data. An alternative is to estimate the class-conditional densities by using representation models for how each pattern class populates the feature space.

In this approach, classifier systems are built by considering each of the classes in turn, and estimating the corresponding class-conditional densities $\wp(\boldsymbol{x}|C_k)$ from data. Methods for estimating probability density can be divided into parametric and non-parametric methods. In parametric methods a specific functional form for the density is assumed, such as unimodal Gaussian density. This contains a number of parameters which are then estimated from training data. Such approaches might be incapable of providing an accurate representation of the true density.

The most widely used method of non-parametric density estimation is the K nearest neighbors (KNN). KNN rule is a powerful technique that can be used to generate highly non-linear classification with limited data. Despite the simplicity of the algorithm, it often performs very well and is an important benchmark method. However, one drawback of KNN is that all the training data must be stored, and a large amount of processing is needed to evaluate the density for a new input pattern. An alternative is to combine the advantages of both parametric and non-parametric methods, by allowing a very general class of functional forms in which the number of adaptive parameters can be increased to build more flexible models. This leads us to a powerful technique for density estimation, called mixture model (Titterington et al., 1985). More extensive discussions of density estimation can be found in (McLachalan and Basford, 1988; Titterington et al., 1985). In our work we focus on semiparametric models based on mixture distributions. We present briefly three density models namely: (i) Gaussian mixture models, (ii) generative topographic mapping and (iii) probabilistic PCA mixture.

3.1. Gaussian mixture models

Gaussian mixture model (GMM) can be classified as a semiparametric density estimation method since it defines a very general class of functional forms for the density model. In a mixture model, a probability density function is expressed as a linear combination of basis functions. A model with M components is described as mixture distribution (Titterington et al., 1985):

$$\wp(\boldsymbol{x}) = \sum_{j=1}^{M} \wp(j)\wp(\boldsymbol{x}|j) \tag{9}$$

where $\wp(j)$ are the mixing coefficients and the parameters of the component density functions $\wp(\boldsymbol{x}|j)$ vary with j. Each mixture component is defined by a Gaussian parametric distribution in d dimensional space:

$$\begin{aligned} \wp(\boldsymbol{x}|j) = &\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} \\ &\times \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^{\top}\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j)\right\} \end{aligned} \tag{10}$$

The parameters to be estimated are the mixing coefficients $\wp(j)$, the covariance matrix $\boldsymbol{\Sigma}_j$ and the mean vector $\boldsymbol{\mu}_j$. The method for training mixture model is based on maximising the data likelihood. The log likelihood of the dataset $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, which is treated as an error function, is defined by

$$l = \sum_{i=1}^{n} \log \wp(\boldsymbol{x}_i) \tag{11}$$

A specialized method is commonly used to produce optimum parameters, known as the expectation-maximization (EM) algorithm (Bishop, 1995). The EM algorithm iteratively modifies the model parameters starting from the initial iteration $k = 0$. For GMM, the EM optimization can be carried out analytically with a simple set of equations (Bishop, 1995), where the mixing coefficients are estimated by

$$\wp^{k+1}(j) = \frac{1}{n}\sum_{i=1}^{n}\wp^{k}(j|\boldsymbol{x}_i) \tag{12}$$

and the estimate for the means for each component is given by

$$\boldsymbol{\mu}_j^{k+1} = \frac{\sum_{i=1}^{n}\wp^{k}(j|\boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{n}\wp^{k}(j|\boldsymbol{x}_i)} \tag{13}$$

and, finally, the update equation for the covariance matrix is

$$\boldsymbol{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{n}\wp^{k}(j|\boldsymbol{x}_i)(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{k+1})(\boldsymbol{x}_i-\boldsymbol{\mu}_j^{k+1})^{\top}}{\sum_{i=1}^{n}\wp^{k}(j|\boldsymbol{x}_i)} \tag{14}$$

Although the class-conditional distributions in feature space are generally non-Gaussian, the resulting multi-modal approximation is remarkably accurate.

3.2. Generative topographic mapping

In many classification problems we have to deal with high dimensional data. Therefore we would like to model the distribution $\wp(\boldsymbol{x})$ parametrized by latent variables $\boldsymbol{z}$ in low-dimensional space. After estimating $\wp(\boldsymbol{x}|\boldsymbol{z})$, the dependence on $\boldsymbol{z}$ has to be integrated out to obtain the density in data space $\wp(\boldsymbol{x})$, where

$$\wp(\boldsymbol{x}) = \int \wp(\boldsymbol{x}|\boldsymbol{z})\wp(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \tag{15}$$

The generative topographic mapping (GTM) (Bishop et al., 1996) is one of the more popular methods for dealing with this situation. It is a mixture model, which means Eq. (15) is approximated by a sum over M Gaussians:

$$\wp(\boldsymbol{x}) = \frac{1}{M}\sum_{j=1}^{M}\wp(\boldsymbol{x}|\boldsymbol{z}_j) \tag{16}$$

$\wp(\boldsymbol{z})$ is assumed to be a uniform distribution and each mixture component is a spherical Gaussian with variance σ^2, and the jth centre is given by a parametrized mapping $\boldsymbol{y}(z_j,\boldsymbol{W})$. Eq. (16) can be rewritten as

$$\wp(\boldsymbol{x}|\boldsymbol{W},\sigma) = \frac{1}{M(2\pi\sigma^2)^{d/2}} \times \sum_{j=1}^{M}\exp\left\{-\frac{\|\boldsymbol{y}(\boldsymbol{z}_j,\boldsymbol{W})-\boldsymbol{x}\|^2}{2\sigma^2}\right\} \tag{17}$$

It is a constrained mixture model because the centres are not independent but are related by the mapping $\boldsymbol{y}$. In GTM method, the mapping from $\boldsymbol{z}$ to $\boldsymbol{x}$ is modeled with a RBF:

$$\boldsymbol{y}(\boldsymbol{z},\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{\Phi}(\boldsymbol{z}) \tag{18}$$

where $\boldsymbol{\Phi}(\boldsymbol{z})$ are K fixed basis functions $\boldsymbol{\Phi}_i(z)$ and $\boldsymbol{W}$ is a $d\times K$ matrix of the adjustable network weights. The log likelihood for a dataset $\boldsymbol{x}_i$, $i = 1,\ldots,n$ is given by

$$l(\boldsymbol{W},\sigma) = \sum_{i=1}^{n}\log\left(\frac{1}{M}\sum_{j=1}^{M}\wp(\boldsymbol{x}_i|\boldsymbol{z}_j,\boldsymbol{W},\sigma)\right) \tag{19}$$

This log likelihood is then maximized in terms of $\boldsymbol{W}$ and σ using an expectation-maximization (EM) algorithm.

3.3. Probabilistic PCA mixture

Classical PCA is made into a density model by using a latent variable approach, in which the data $\boldsymbol{x}$ is generated by a linear combination of a number of variables $\boldsymbol{z}$ of low-dimension ($q < d$). The mapping from $\boldsymbol{z}$ to $\boldsymbol{x}$ is given by

$$\boldsymbol{y}(\boldsymbol{z},\boldsymbol{W}) = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu} \tag{20}$$

$\boldsymbol{\mu}$ represents the data mean. The probability model of PCA can be written as a combination of the conditional distribution (Tipping and Bishop, 1999):

$$\wp(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{W},\sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}} \times \exp\left\{-\frac{\|\boldsymbol{x}-\boldsymbol{W}\boldsymbol{z}-\boldsymbol{\mu}\|^2}{2\sigma^2}\right\} \tag{21}$$

and the latent variable distribution:

$$\wp(\boldsymbol{z}) = \frac{1}{(2\pi)^{q/2}} \exp\left\{-\frac{\boldsymbol{z}\boldsymbol{z}^{\top}}{2}\right\} \tag{22}$$

By integrating out the latent variables $\boldsymbol{z}$, we obtain the distribution of the observed data, which is also Gaussian:

$$\wp(\boldsymbol{x}|\boldsymbol{W},\sigma) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{C}|^{1/2}} \times \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{C}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \tag{23}$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^{\top} + \sigma^2\boldsymbol{I}$. The covariance matrix is the sum of two terms: one is diagonal in a q-dimensional subspace spanned by the first q principal components and the other is spherical. A mixture of PPCA has the same form as Eq. (9), where each component density function is given by a probabilistic PCA. Hence, the training of such a model can be done in the maximum likelihood framework using an EM algorithm.

4. Experimental results

4.1. Data description

For the identification of combustion gases we have used an array of eight micro-electronic gas sensors. This multi-sensor based on SnO_2 thin film was integrated using a micro-electronic structure referred to as the micro-hotplate (MHP) (Yan et al., 1998). The experiment equipment consists, basically, of the sensor chamber, mass flow controllers, gas pumps and a data acquisition board. Fig. 2 shows this experimental setup.

The measurement procedure consists of two steps. The first step consists of injecting the tested gas during 10 min period, while 40 min are allocated to a cleaning stage with dry air. Data are collected at a sampling period of 3 s. Gases used in the experiment are methane, carbon monoxide, hydrogen, and two binary mixtures of methane and carbon monoxide as well as hydrogen and carbon monoxide respectively. Different concentrations are used for each gas. Concentration ranges used for the experiment are reported in Table 1.

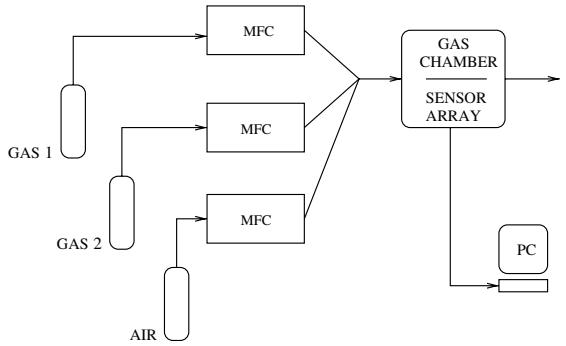

Fig. 2. Scheme of the experimental setup. MFC stands for Mass Flow Controler.

The sensors' outputs are raw voltage measurements in the form of exponential-like curves. Fig. 3 shows the typical steady state response for the sensor array exposed to methane, carbon monoxide and their mixture.

The steady state values of the array sensor were recorded while periodically injecting different gases. Fig. 4 shows the polar representation of the normalized steady state responses for the sensor array exposed to different gases, which gives a good indication of the different response signatures. Although some differences between the patterns can be seen, the shapes appear to be similar and most of the sensors present a lack of selectivity.

A gas data set of 220 patterns (each pattern consists of eight sensor responses) was created to train the different density classifiers and to evaluate their identification performance.

Table 1
Gases and their concentration ranges

Gas	Concentration range (ppm)
CO	25–200
CH_4	500–4000
CO & CH_4	25–200 & 500–4000
H_2	500–2000
CO & H_2	25–200 & 500–2000

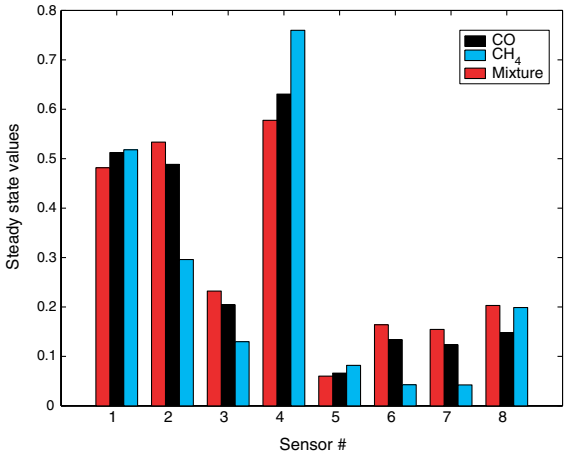

Fig. 3. Histograms showing the response patterns of the eight gas sensors exposed to CH_4, CO and their mixture.

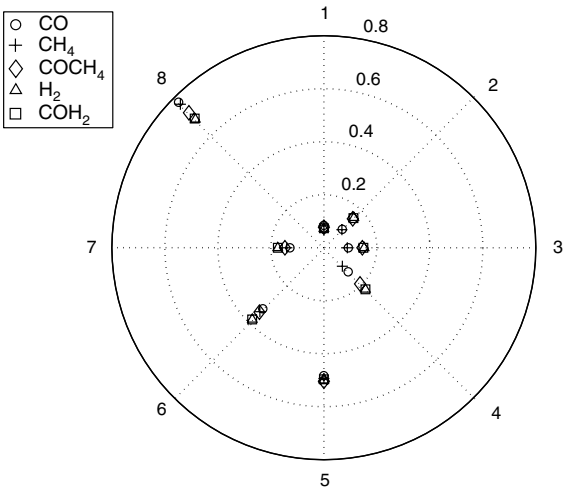

Fig. 4. Polar representation of the array response exposed to different gases. Measurement type, CO (circles), CH_4 (plus signs), mixture CO–CH_4 (diamonds), H_2 (triangles) and mixture CO–H_2 (squares).

4.2. Classification accuracy

In order to evaluate the classification performance of each density model and compare their performances, we first consider the whole dataset. The mixture model classifier is built by considering each of the classes in turn, and estimating the corresponding class-conditional densities $\wp(\boldsymbol{x}|C_k)$ from the data. The parameters of each mixture model were adapted to the training data in the maximum likelihood framework using EM algorithm. Since the dataset we used was small, generalization performances were estimated by using the 10-fold cross validation approach. Table 2 reports the classification performance of the mixture models in comparison to the one of the KNN ($K = 3$) classifier. The GMM classifier appears to be the best model in terms of classification accuracy.

A second study was made on reduced datasets using different projection techniques. The inputs to each classifier are the projections of the data using PCA, LDA or Neuroscale. The obtained results are reported in Table 3.

PCA projection improves the classification accuracy for all density model classifiers. We have noticed that the classifiers performance depends on the dimension of the reduced feature vector. For PCA, the classification performance was optimized as a function of the number of principal components. The best performance is achieved using GMM with a success rate of 94.5% obtained for five principal components.

One way to build a classifier is to develop a discriminant function that estimates the posterior probabilities directly, where the boundaries are learnt from data. MLP and RBF are the two most popular types of neural network architectures used for electronic nose applications (Gutierrez-Osuna,

Table 2
Classification accuracy of various classifiers on the whole dataset

Classifier	Accuracy (%)
KNN	88.6
GMM	**90.9**
GTM	88.2
PPCA	88.2

Table 3
Identification accuracy (%) for density models, using PCA, Neuroscale and LDA projections

Classifier	PCA	Neuroscale	LDA
KNN	91.4	**91.3**	83.2
GMM	**94.5**	90.9	**90.5**
GTM	90	86.8	83.2
PPCA	89	85.5	81.4

Table 4
Identification accuracy (%) for discriminant functions, using PCA, Neuroscale and LDA projections

Classifier	PCA	Neuroscale	LDA
RBF	83	76.8	80
MLP	**90**	**88.2**	**89.4**
GLM	86.4	86.4	81.8

2002). We have also compared our density models with RBF, MLP and GLM (Generalized Linear Model). For discriminant functions a softmax output activation function is used to ensure that the outputs lay in the range [0,1] and summed to one. All parameters (for example, the number of hidden units) were chosen according to the validation stage. Table 4 reports the classification performance of the trained classifiers. This table shows that the best results are given by the PCA projection. The most accurate discriminant function is the MLP. GMM is shown to outperform MLP.

5. Concluding remarks

In this paper we presented a gas identification method based on class-conditional density estimation. We conducted a comparative experiment to decide which density models were most suitable for the problem of classifying combustion gases. In this study, GMM is shown to outperform GTM, PPCA mixture and KNN for the gas sensors dataset collected from the integrated gas sensor array. Experiments showed that PCA projection enhances the classification quality. A significant dependence of the classifiers performance on the number of principal components was found. The classification performance was therefore optimized as a function of the number of components.

Comparison with various neural networks architectures (MLP, RBF and GLM) revealed that GMM presents the best classification performance (over 94%) among all tested classifiers.

Acknowledgments

The work described in this paper was supported by a competitive earmarked research grant (HKUST 6162/04E) from the research grant council of Hong Kong.

References

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Bishop, C.M., Svensen, M., William, C.K.I., 1996. GTM: The Generative topographic mapping. Neural Comput. 10 (1), 215–235.

Gutierrez-Osuna, R., 2002. Pattern analysis for machine olfaction: a review. IEEE Sens. J. 2 (3), 189–202.

Lowe, D., Tipping, M.E., 1996. Feed-forward neural networks and topographic mappings for exploratory data analysis. Neural Comput. Appl. 4, 83–95.

McLachalan, G.J., Basford, K.E., 1988. Mixture Models: Inference and Applications to clustering. Marcel Dekker, New York.

Tipping, M.E., Bishop, C.M., 1999. Probabilistic principal component analysis. J. Roy. Statist. Soc. B 61, 611–622.

Titterington, D.M., Smith, A.F.M., Makov, U.E., 1985. Statistical Analysis of Finite Mixture Distributions. John Wiley, New York.

Vasconcelos, N., Lippman, A., 2000. Feature representations for image retrieval: beyond the color histogram. In: Proc. IEEE ICME Conf., vol. 2, 899–902.

Yan, G., Sheng, L., Tang, Z., Wu, J., Chan, P.C.H., Sin, J.K.O., 1998. A low power CMOS compatible integrated gas sensor using maskless tin oxide sputtering. Sens. Actuators B 49, 81–87.

Zhang, Y.M., Alder, T.R., 1994. Using Gaussian mixture modeling in speech recognition. In: Proc. ICASSP Conf., vol. 1, pp. 613–616.