Bayesian Learning Using Gaussian Process for Gas Identification

Amine Bermak, *Senior Member, IEEE*, and Sofiane Brahim Belhouari

***Abstract*—In this paper, a novel gas identification approach based on Gaussian process (GP) combined with principal components analysis is proposed. The effectiveness of this approach has been successfully demonstrated on an experimentally obtained dataset. Our aim is the identification of different gases with an array of commercial Taguchi gas sensors (TGS) as well as microelectronic gas sensors. The proposed approach is shown to outperform both K nearest neighbor (KNN) and multilayer perceptron (MLP) classifiers.**

***Index Terms*—Bayesian learning, gas identification, gas sensor array, Gaussian processes (GPs), pattern recognition.**

I. Introduction

The past decade has seen an increased interest for the development of integrated human-like smell sensing capabilities. The problem of classifying and further quantifying chemical substances is very critical for a wide range of applications in the industrial and civil environments [1]. There have already been several reports on the use of different kinds of sensors for detecting poisonous and dangerous gases [2], [3]. Present gas sensors are very sensitive, but their selectivity and response are greatly affected by many pollutants like the relative humidity, temperature variation, and background odors. It is well known that gas sensors suffer from a number of shortcomings such as low selectivity, nonlinearities, and instability. Waiting for the development of a new generation of sensors, strategies and algorithms of pattern recognition must be developed 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. After a preprocessing stage, the resulting 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 [4]. Various kinds of flexible pattern recognition algorithms have been used for classifying chemical sensor data. Most notably, neural networks have been exploited, in particular multilayer perceptrons (MLPs) and radial basis functions (RBFs). Other methods based on the class-conditional density estimation have been used, such as quadratic and K nearest neighbor (KNN) classifiers. These parametric and nonparametric density estimation methods have their merits and limitations.

A new method for regression was inspired by Neal's work on Bayesian learning for neural networks [5]. It is an attractive method for modeling noisy data, based on priors over function using Gaussian processes (GPs). It was shown that many Bayesian regression models based on neural networks converge to GPs in the limit of an infinite network [5]. GP models have been applied to the modeling of noise-free [6] and noisy data [7] as well as to classification problems [8]. Recently, GP has been successfully used for nonstationary time series forecasting with excellent performance [9]. In this paper, we present a gas classification approach based on GP combined with principal components analysis (PCA). The effectiveness of the proposed approach has been successfully applied to an experimentally obtained dataset where the objective is the identification of different gases using commercial Taguchi gas sensors (TGS), as well as microelectronic gas sensor arrays. It is shown that the proposed approach outperforms both KNN and MLP classifiers.

II. Classification Problem

The objective of pattern recognition is to set a decision rule that 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 $C_{\hat{k}}$ [10], i.e.,

$$C_{\hat{k}} = \arg\max_{\{1,\ldots,c\}} [\wp(C_k|\boldsymbol{x})] = \arg\max_{\{1,\ldots,c\}} [\wp(\boldsymbol{x}|C_k)\wp(C_k)] \quad (1)$$

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.

Manuscript received October 10, 2004; revised November 2, 2005. This work was supported by the Research Grant Council of Hong Kong under Competitive Earmarked Research Grant HKUST 6162/04E.

The authors are with the Smart Sensory Integrated Systems Laboratory, Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: eebermak@ust.hk; eebelhou@ust.hk).

Digital Object Identifier 10.1109/TIM.2006.873804

A. KNN

The most widely used method for nonparametric density estimation is the KNN. The KNN rule is a powerful technique that can be used to generate highly nonlinear classification with limited data. To classify a pattern $\boldsymbol{x}$, we find the closest K examples in the dataset and select the predominant class $C_{\hat{k}}$ among those K neighbors. 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.

B. MLPs

The posterior densities can be estimated via discriminant analysis, where the boundaries are directly learnt from data. This supervised classification is based on the learning of the input pattern-class label mapping. The posterior probability is estimated by $\sigma(y(\boldsymbol{x}))$, where $\sigma(y)$ is the activation function. Usually, we use the logistic function $\sigma(y) = 1/1 + \exp(-y)$ for the two-class problem and the softmax activation function for the multiclass classification. One way to generalize the discriminant function to permit a much larger range of possible decision boundaries is to transform the input vector $\boldsymbol{x}$ using a set of H predefined nonlinear basis functions $\phi_j(\boldsymbol{x})$ and represent the output as a linear combination of these functions.

$$y_k(\boldsymbol{x}) = \sum_{j=1}^{H} \omega_{kj} \phi_j(\boldsymbol{x}) + \omega_{k0}. \tag{2}$$

The MLP is probably the most widely used architecture for practical applications of neural networks [11]. For a high-dimensional input space, there will be a large number of basis functions, each one with the associated parameters, and one risks overfitting the training data. Network complexity must be controlled by determining the optimal number (in terms of small generalization error) of hidden units H. The goal is to find a network that gives the best prediction on a new dataset. The simplest approach is to train several MLPs using the same training dataset. The performance of the networks is then compared by evaluating the error function using an independent validation set. Network complexity can also be controlled by using regularization approaches such as weight decay [10]. The problem with regularization is that it is difficult to determine the optimum value for the regularization parameter. Additional details on complexity control can be found in [12].

C. Bayesian Learning

One approach to the complexity control problem is to consider the Bayesian framework explained in [11] and [13]. In this framework, we start with the priors on the parameters, which express our knowledge before the data is observed. Once we observe the data, Bayes' rule can be used to update our beliefs, and we obtain the posterior probability density. In pattern recognition applications, Bayesian methods offer a number of important practical benefits [11], [14].

- By taking into account the parameter uncertainty, overfitting problem is solved.
- Regularization can be given a natural interpretation in the Bayesian framework.
- The relative importance of different input variables can be determined using automatic relevance determination.

However, the Bayesian analysis is difficult because a simple prior over parameters implies a complex prior distribution over functions. With neural networks, approximation or sampling methods are needed to perform the Bayesian inference and to evaluate some integrals. Rather than expressing our prior knowledge in terms of a prior for the parameters, we can instead integrate over the parameters to obtain a prior distribution for the model outputs in any set of cases. This operation is most easily carried out if all the distributions are Gaussian. Fortunately, GPs are flexible enough to represent a wide variety of interesting model structures, many of which would have a large number of parameters if formulated in more classical fashion. The GP method can be extended to classification problems by defining a GP over y: the input to the sigmoid function.

III. GP CLASSIFIERS

A GP is a collection of random variables, any finite set of which have a joint Gaussian distribution [8]. For a finite collection of inputs, $\boldsymbol{x} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}]^T$, we consider a set of random variables $\boldsymbol{y} = [y^{(1)}, \ldots, y^{(n)}]^T$ to represent the corresponding function values. A GP is used to define the joint distribution

$$\wp(\boldsymbol{y}|\boldsymbol{x}) \sim \exp\left(-\frac{1}{2}\boldsymbol{y}^T \Sigma^{-1} \boldsymbol{y}\right) \tag{3}$$

where the matrix Σ is given by the covariance function $\Sigma_{pq} = \mathrm{cov}(y^{(p)}, y^{(q)}) = C(\boldsymbol{x}^{(p)}, \boldsymbol{x}^{(q)})$.

A. Parameterizing the Covariance Function

There are many possible choices of prior covariance functions. From a modeling point of view, we wish to specify prior covariance that contains our prior beliefs about the structure of the function we are modeling. Formally, we are required to specify a function that will generate a non-negative definite covariance matrix for any set of input points. GP procedure can handle interesting models by simply using a covariance function with an exponential term [7], i.e.,

$$C\left(\boldsymbol{x}^{(p)}, \boldsymbol{x}^{(q)}\right) = v_0 \exp\left\{-\frac{1}{2}\sum_{l=1}^{d} \rho_l \left(x_l^{(p)} - x_l^{(q)}\right)^2\right\} + v_1 \tag{4}$$

where $\boldsymbol{\theta} = (\rho_1, \ldots, \rho_d, v_0, v_1)$ plays the role of hyperparameters. It expresses the idea that cases with nearby inputs will have highly correlated outputs. One way to construct a variety of covariance functions is by adding and multiplying together other covariance functions.

B. Classifying With GP

For the two-class problem, we use the logistic function to produce an output that can be interpreted as $\pi(\boldsymbol{x}) = \sigma(y(\boldsymbol{x}))$, the probability of the input $\boldsymbol{x}$ belonging to class 1. The probability and the activation corresponding to input $\boldsymbol{x}_i$ will be noted by π_i and y_i, respectively. We specify a GP prior over the activation function y that we have a probabilistic model as a discriminant classifier.

Making classification for a test input $\boldsymbol{x}^*$ given a set of training data $D = \{\boldsymbol{x}^{(i)}, t^{(i)} | i = 1, \ldots, n\}$, with a fixed set of hyperparameters $\boldsymbol{\theta}$, requires the calculation of the integral defined as

$$\hat{\pi}^* = \int \pi^* \wp(\pi^* | \boldsymbol{t}, \boldsymbol{\theta}) d\pi^* \tag{5}$$

where $\pi^* = \pi(\boldsymbol{x}^*)$. To find the posterior distribution of π^*, we use the distribution $\wp(y^* | \boldsymbol{t}, \boldsymbol{\theta})$, which is defined by

$$\begin{aligned}\wp(y^* | \boldsymbol{t}, \boldsymbol{\theta}) &= \int \wp(y^*, \boldsymbol{y} | \boldsymbol{t}, \boldsymbol{\theta}) d\boldsymbol{y} \\ &= \frac{1}{\wp(\boldsymbol{t}|\boldsymbol{\theta})} \int \wp(y^*, \boldsymbol{y} | \boldsymbol{\theta}) \wp(\boldsymbol{t} | \boldsymbol{y}) d\boldsymbol{y} \end{aligned} \tag{6}$$

and apply the appropriate Jacobian to transform the distribution. Because we are using a logistic sigmoid output function, the appropriate noise model is no longer Gaussian but Bernoulli.

$$\wp(\boldsymbol{t}|\boldsymbol{y}) = \Pi_{i=1}^{n} \pi_i^{t_i} (1 - \pi_i)^{1 - t_i} \tag{7}$$

which means that (6) is no longer analytically tractable. Faced with this problem, there are two ways to evaluate the integral, namely 1) to use an analytic approximation or 2) to use Monte Carlo methods. We consider an analytic approximation based on Laplace's method. We approximate the function $\wp(y^*, \boldsymbol{y} | \boldsymbol{t}, \boldsymbol{\theta})$ by a Gaussian distribution at the maximum of this function with inverse covariance matrix given by $-\nabla\nabla \log \wp(y^*, \boldsymbol{y} | \boldsymbol{t}, \boldsymbol{\theta})$. Finding a maximum can be carried out using the Newton–Raphson algorithm, which then allows the approximation distribution of y^* to be calculated. Details of the maximization procedure can be found in [15].

C. Training a GP

Let us assume that a form of covariance function has been chosen, but depends on undetermined hyperparameters $\boldsymbol{\theta}$. We would like to learn these hyperparameters from the training data. In a maximum likelihood framework, we adjust the hyperparameters to maximize the log likelihood of the hyperparameters $\log \wp(\boldsymbol{t}|\boldsymbol{\theta})$. For the classification problem, we have to approximate this function by using Laplace's method. Let $\varphi = \log \wp(\boldsymbol{t}|\boldsymbol{y}) + \log \wp(\boldsymbol{y})$. Using $\wp(t_i | y_i) = t_i y_i - \log(1 + \exp(y_i))$, we obtain [15]

$$\begin{aligned}\varphi = \boldsymbol{t}^T \boldsymbol{y} - \sum_{i=1}^{n} \log\left(1 + \exp(y_i)\right) \\ - \frac{1}{2} \boldsymbol{y}^T Q^{-1} \boldsymbol{y} - \frac{1}{2} \log \det(Q) - \frac{n}{2} \log 2\pi \end{aligned} \tag{8}$$

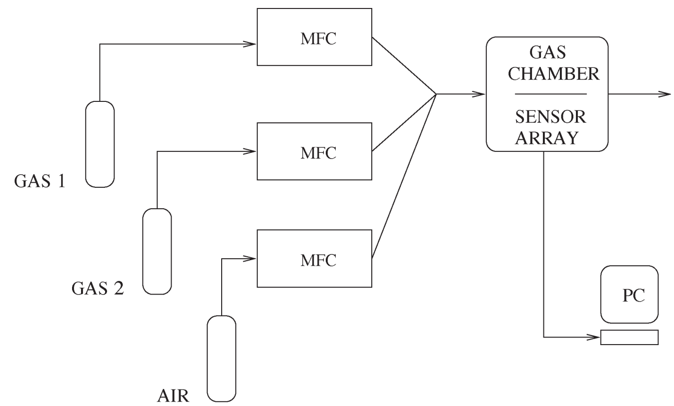

Fig. 1. Scheme of the experimental setup. MFC stands for mass flow controller.

where $Q_{pq} = C(\boldsymbol{x}^{(p)}, \boldsymbol{x}^{(q)})$. By using Laplace's approximation about the maximum $\tilde{\boldsymbol{y}}$, we find that

$$\log \wp(\boldsymbol{t}|\boldsymbol{\theta}) = \varphi(\tilde{\boldsymbol{y}}) - \frac{1}{2} \log \det(Q^{-1} + W) - \frac{n}{2} \log 2\pi. \tag{9}$$

We initialize the hyperparameters to random values (in a reasonable range) and then use an iterative method, for example, conjugate gradient, to search for optimal values of the hyperparameters. We found that this approach is sometimes susceptible to local minima. To overcome this drawback, we randomly selected a number of starting positions within the hyperparameters space.

The extension of the method to multiple classes is essentially straightforward, although considerably more complex. A softmax activation function for each class C_k $(k = 1, \ldots, c)$ with a one-of-c coding scheme is used [15]. The probability of the input $\boldsymbol{x}$ belonging to class C_k is given by

$$\pi_k(\boldsymbol{x}) = \frac{\exp(y_k)}{\sum_{k'} \exp(y'_k)} \tag{10}$$

which ensures that the GP outputs lay in the range [0, 1] and summed to 1. As for a two-class case, one can take the new log likelihood for similarly represented targets t_k^i $(i = 1, \ldots, n)$ and assume that GP prior operates in the activation space.

IV. Experimental Results

We evaluated the performance of the proposed GP classifier on five datasets of gases collected from both a commercial TGS and a microhotplate (MHP) microelectronic gas sensors arrays.

A. Data Description

Measurements have been done with an experimental equipment consisting of gas pumps, mass flow controllers, a sensor chamber, and a computer used for data acquisition and the experiment control. Fig. 1 shows the experimental setup used in our work. In a gas chamber, we placed a sensor array based either on the commercial TGS or MHP microelectronic gas sensors. Vapors were injected into the gas chamber at a flow rate determined by the mass flow controllers. Measurement

TABLE I
GASES AND THEIR CONCENTRATION RANGES

Gas	Concentration range (ppm)
C_2H_6O	1000-5000
$C_4H_{10}O$	800-3000
CO	25-200
CH_4	500-4000
H_2	500-2000

TABLE II
DATASET CHARACTERISTICS

Dataset	Type of gas	Number of sensors	Number of patterns
1	$(C_2H_6O, C_4H_{10}O)$	5	124
2	$(CH_4, CO-CH_4)$	8	120
3	(H_2, H_2-CO)	8	54
4	$(CO, CO-CH_4)$	8	128
5	$(CO, CH_4, CO-CH_4)$	8	168

procedure consists of two steps. The first step consists of injecting the tested gas during the 10-min period, whereas 40 min is allocated to a cleaning stage with dry air. Data are collected at a sampling period of 3 s. We have performed five experiments with different gases.

- First experiment: A sensor array composed of five commercial tin oxide gas sensors, TGS Figaro (TGS 2600, 2602, 2610, 2611, and 2620), have been used to discriminate between ethanol (C_2H_6O) and butanol ($C_4H_{10}O$). Different concentrations were used for both vapors (see Table I). We have recorded the steady-state outputs of the sensor array.
- Second experiment: For the detection of CO in the presence of CH_4, we have used an array of eight microelectronic gas sensors. This multisensor based on SnO_2 thin film was integrated using a microelectronic structure referred to as the MHP [16]. Different concentrations were used for each gas (see Table I). The steady-state values of the sensor array were recorded while periodically injecting different gases.
- Third experiment: We used the same experimental procedure as the second one to detect CO in the presence of hydrogen H_2.
- Fourth experiment: We built the fourth dataset to separate the signatures of carbon monoxide and the mixture of carbon monoxide and methane. The experimental setup is similar to the one used in the previous two experiments.
- Fifth experiment: Methane, carbon monoxide, or their mixture vapors were injected into the gas chamber at a flow rate determined and accurately controlled by the mass flow controllers. Different concentrations were used for CO ranging from 25 to 200 ppm and for CH_4 ranging from 500 to 4000 ppm. The CO–CH_4 mixture concentration was also injected into the gas chamber with different combination of both gases' concentrations. The MHP microelectronic gas sensor array was again used in this experiment.

Concentration ranges used for the experiments are reported in Table I. Inasmuch as our goal is the qualitative classification of patterns, a normalization procedure is used to reduce the influence of concentrations and nonlinearities. Each input pattern is divided by its Euclidean norm. The characteristics of the datasets are summarized in Table II. It is important to notice the small amount of data available, which is typically the case in the gas sensor experiments as it is very costly and time-consuming to obtain a large, reliable, and representative set of examples.

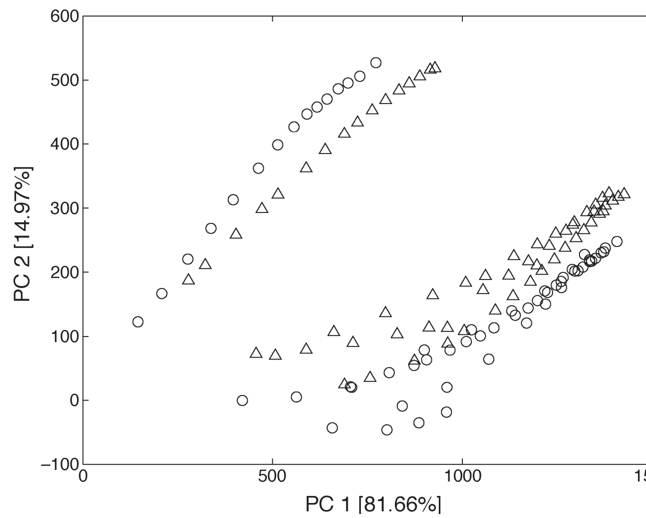

Fig. 2. PCA results for the first dataset. Measurement type: ethanol (circles) and butanol (triangles).

B. Identification Results

The obtained datasets were used to estimate the performance of the GP classifier and compared with both MLP and KNN. The most important multivariate tool for exploratory analysis is the PCA. PCA was therefore used as a preprocessing stage for redundancy removing and feature reduction before applying a given classifier. By applying this technique, principal components and eigenvalues are calculated. Inasmuch as the two first components allow us to take into account more than 95% of the data variance, we limit our representation to these components. Figs. 2–6 present the first two PCA scores for each of the datasets so that we can judge the relative complexity of the different datasets. We note that the features can be separated into different distributions corresponding to each measurement gas type. However, the decision boundaries are not well defined due to some overlapping, especially the third and the fifth datasets, having the most complex PCA scatter plots (Figs. 4–6).

The inputs to each classifier are the projections of the data on the first two principal components. The hyperparameters of the GP classifier were adapted to the training data using conjugate gradient search algorithm. All parameters for MLP and KNN were chosen according to the validation stage. Inasmuch as the datasets we used were small, generalization performances were estimated by using the N-fold cross validation approach. Table III reports the identification accuracy of the GP classifier in comparison to the one of KNN and the MLP classifiers. For the five datasets, the results using the GPs are better than both

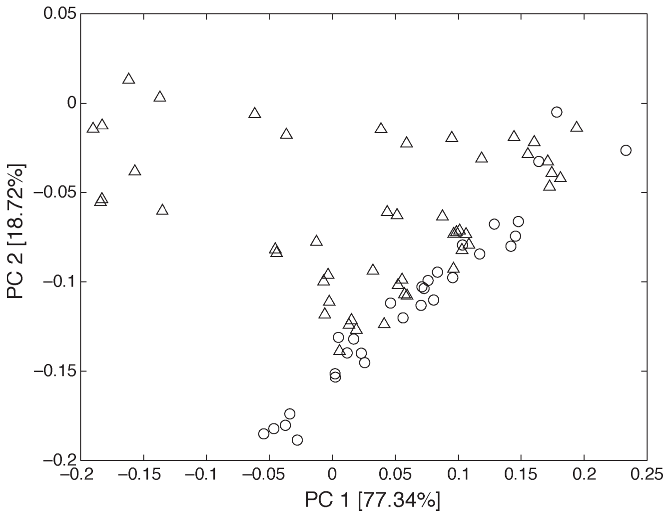

Fig. 3. PCA results for the second dataset. Measurement type: CH_4 (circles) and mixture CO–CH_4 (triangles).

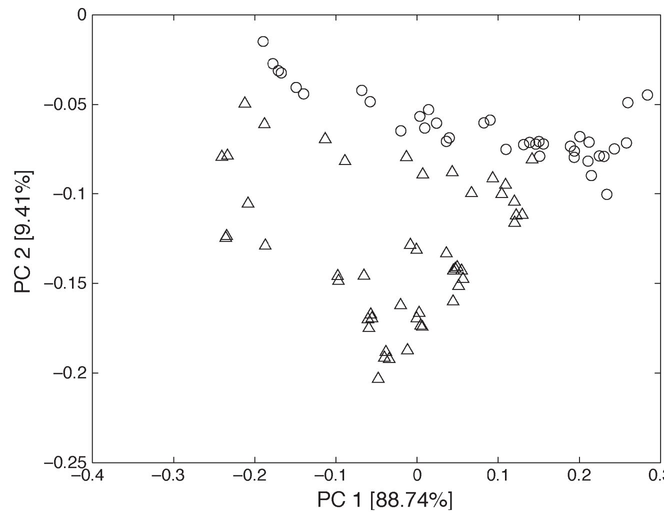

Fig. 5. PCA results for the fourth dataset. Measurement type: CO (circles) and mixture CO–CH_4 (triangles).

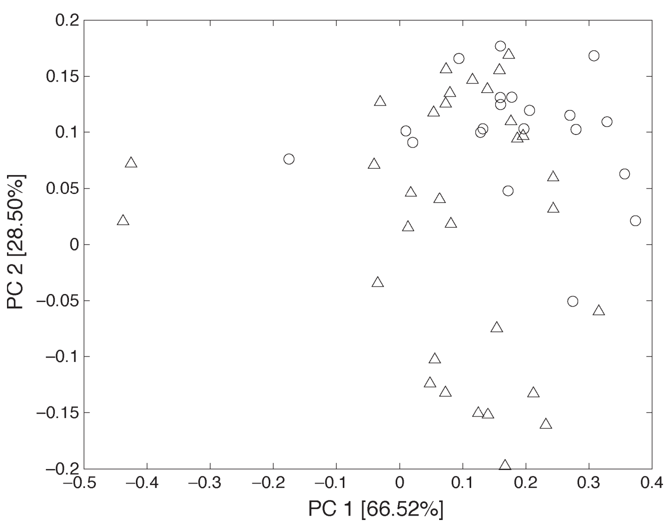

Fig. 4. PCA results for the third dataset. Measurement type: H_2 (circles) and mixture CO–H_2 (triangles).

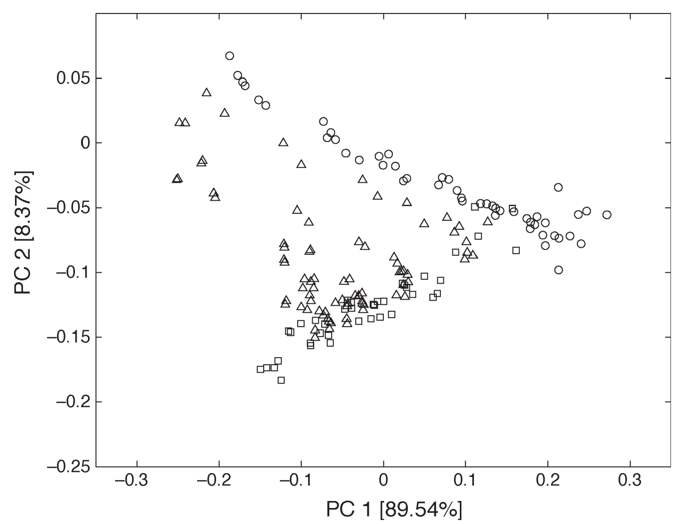

Fig. 6. PCA results for the fifth dataset. Measurement type: CO (circles), CH_4 (squares), and mixture CO–CH_4 (triangles).

the KNN and MLP classifiers. These results can be explained by the fact that, with Bayesian learning, GP classifier succeeds in building a statistical model of the process that generates the data and, hence, exhibits better generalization performance than KNN and MLP. The GP classifier is also applied to the general case of more than two classes as shown in the fifth experiment. The performances are still consistently higher as compared with KNN and MLP.

TABLE III
GAS IDENTIFICATION RESULTS (%)

Classifier	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
GP	**90.5**	**98**	**75**	**100**	**90**
KNN	85	93.3	57.7	95.4	89.9
MLP	88.3	95.8	67.3	99.2	88.3

V. CONCLUSION

In this paper, we proposed a gas identification method based on GP models combined with PCA. We have shown that an excellent classification rate can be achieved on both TGS and microelectronic gas sensor data. Using GP classifiers, we obtained results that are superior to both KNN and MLP for gas identification applications. In addition, it was shown that the GP classifier can be applied to the general case of identifying more than two gases. However, a problem found in GP-based methods is the computational cost because they require computations (trace, determinants, and linear solutions) involving $n \times n$ matrices, where n is the number of training examples. This represents a drawback when dealing with large datasets. It is of interest to investigate the possibility of speeding up Bayesian computations by using specific numerical methods.

ACKNOWLEDGMENT

The authors would like to thank Prof. P. Chan and Dr. D. Martinez for support and help in obtaining the datasets.

REFERENCES

[1] D.-S. Lee *et al.*, "SnO_2 gas sensing array for combustible and explosive gas leakage recognition," *IEEE Sensors J.*, vol. 2, no. 3, pp. 140–149, Jun. 2002.
[2] D.-D. Lee, "Hydrocarbon gas sensors," in *Chemical Sensor Technology*, vol. 5. Tokyo, Japan: Kodansha, 1994, pp. 79–99.
[3] J. W. Gardner and P. N. Barelett, "A brief history of electronic noses," *Sens. Actuators B, Chem.*, vol. 18/19, no. 1–3, pp. 211–220, Mar. 1994, Elsevier Sequoia.
[4] R. Gutierrez-Osuna, "Pattern analysis for machine olfaction: A review," *IEEE Sensors J.*, vol. 2, no. 3, pp. 189–202, Jun. 2002.
[5] R. M. Neal, *Bayesian Learning for Neural Networks*. New York: Springer-Verlag, 1996.
[6] ——, "Monte Carlo implementation of Gaussian process models for Bayesian regression and classification," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 9702, 1997.
[7] C. K. I. Williams and C. E. Rasmussen, "Gaussian process for regression," in *Advances in Information Processing Systems*, 8th ed. Cambridge, MA: MIT Press, 1996, pp. 111–116.
[8] D. J. C. MacKay. (1997). "Gaussian processes—A replacement for supervised neural networks," in *Lecture Notes Tutorial NIPS*. [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/gp.pdf
[9] S. Brahim Belhouari and A. Bermak, "Gaussian process for nonstationary time series prediction," *Computat. Stat. Data Anal.*, vol. 47, no. 4, pp. 705–712, Nov. 2004.
[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.
[11] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.
[12] M. Pardo and G. Sberveglieri, "Learning from data: A tutorial with emphasis on modern pattern recognition methods," *IEEE Sensors J.*, vol. 2, no. 3, pp. 203–217, Jun. 2002.
[13] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, May 1992.
[14] I. T. Nabney, *Netlab, Algorithms for Pattern Recognition*. London, U.K.: Springer-Verlag, 2003.
[15] C. K. I. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1342–1351, Dec. 1998.
[16] G. Yan, L. Sheng, Z. Tang, J. Wu, P. C. H. Chan, and J. K. O. Sin, "A low power CMOS compatible integrated gas sensor using maskless tin oxide sputtering," *Sens. Actuators B, Chem.*, vol. 49, no. 1/2, pp. 81–87, Jun. 1998.

Amine Bermak (M'00–SM'04) received the M.Eng. and Ph.D. degrees in electronic engineering from Paul Sabatier University, Toulouse, France, in 1994 and 1998, respectively.

While completing the Ph.D. degree, he was part of the Microsystems and Microstructures Research Group at the French National Research Center LAAS-CNRS, where he developed a three-dimensional very large scale integration (VLSI) chip for artificial neural network classification and detection applications. He joined the Advanced Computer Architecture research group at York University, York, U.K., where he was a Postdoctoral Researcher on VLSI implementation of correlation matrix memory neural networks for vision applications in a project funded by British Aerospace. In November 1998, he joined Edith Cowan University, Perth, Australia, first as a Research Fellow working on smart vision sensors and then as a Lecturer and a Senior Lecturer in the School of Engineering and Mathematics. He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong, where he is also serving as the Associate Director of the Computer Engineering Programme. His research interests are related to VLSI circuits and systems for signal, image processing, sensors, and microsystem applications. He has published extensively on the aforementioned topics in various journals, book chapters, and refereed international conferences.

Dr. Bermak has received many distinguished awards, including the 2004 IEEE Chester Sall Award, the 2004 Bechtel Foundation Engineering Teaching Excellence Award, and the best paper award at the Fifth IEEE International Workshop on System-On-Chip for Real-Time Applications. He is a member of the IEEE CAS Committee on Sensory Systems.

Sofiane Brahim Belhouari received the engineer diploma in electrical engineering from the Polytechnic Institute of Algiers, Algeria, in 1993 and the Ph.D. degree in automatic control and signal processing from the University of Paris XI, Paris, France, in 2000.

After completing the Ph.D. degree, he held a Postdoctoral positions at Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, and Hong Kong University of Science and Technology (HKUST), Kowloon, Hong Kong. His main research interests are in data analysis, statistical signal processing, and pattern recognition.