Sensors and Materials, Vol. 25, No. 2 (2013) 109–120
MYU Tokyo

S & M 0906

A Novel Retraining Method of Multiple Self-Organizing Maps for Gas Sensor Drift Compensation

Tao Liu*, Kadri Chaibou and Zhiyong Huang

College of Communication Engineering, Chongqing University, Chongqing, 400030, China

(Received September 21, 2011; accepted March 9, 2012)

Key words: multiple self-organizing maps, drift compensation, retraining method, electronic nose

The drift effect of gas sensors severely affects the performance of an electronic nose, because the primarily built pattern recognition models degrade over time. A multiple self-organizing map (MSOM) network is an adaptive approach to compensate for gas sensor drift by self-retraining during the test phase. However, the conventional local retraining method of multiple self-organizing maps may lose drift information if the retraining is carried out with successive homogeneous samples for a long time. In this paper, we propose a novel global retraining method to keep each retraining vector (RV) fresh over time. Compared with the local retraining approach, the new method updates all the retraining vectors after one of them has been replaced. Experimental results demonstrate that the global retraining method retains the network recognition ability on drift effect, whereas the local retraining and adaptive resonance theory methods show high error rates. Finally, a discussion on the retraining rate is given to optimize the process speed of the MOSM network with the global retraining strategy.

1. Introduction

An electronic nose is a smart device that generally consists of an array of nonspecific gas sensors and a proper pattern recognition module, which mimics the human olfactory system. It has been widely used in several domains, such as food freshness detection,[1] medical diagnosis,[2] and environmental protection.[3] Compared with traditional chemical analysis, the electronic nose is characterized by high speed and low cost. However, the fluctuation of the sensor response caused by sensor drift has still limited the performance of the pattern recognition modules in electronic noses.

*Corresponding author: e-mail: cquliutao@cqu.edu.cn

Depending on its causes, sensor drift can be generally divided into temporary drift and long-term drift. When it follows the vibration of environmental factors, such as temperature and humidity, it is considered as a temporary drift, whereas the long-term drift arises from some inner factors such as sensor aging and poisoning. For the temporary drift, a few methods have been suggested to find appropriate mathematical models describing the relationship between drift and environmental factors.(4) These methods provide guidelines to adjust recognition outputs. In contrast, the long-term drift shows slow and gradual changes in sensor responses over time and has no clear tendency. Therefore, methods to compensate for the long-term drift are more complex than those used in the temporary drift counteraction. In our study, we concentrate on the solutions to cope with the long-term drift. Thus, the drift in the following parts refers to the long-term drift.

Although many efforts have been made to reduce the negative effect from the gas sensor drift, it still constitutes a major cause of the degradation of electronic noses' selectivity and sensitivity. According to the literature, the component correction model and adaptive model are the two main types of drift compensation methodology attracting researchers.

The component correction model is an active approach to remove drift as a component of output signals of a gas sensor array. The key point of this methodology is finding a proper algorithm to eliminate the drift component. The first scheme finds the drift signal through comparison with a reference gas immediately.(5) Unfortunately, it is difficult to maintain uniform testing conditions for both reference gas and sample gas in practical applications. Thus, the evaluated drift elements calculated from the reference gas and sample gas may be different. Moreover, the second idea is collecting drift trace from the former testing process. The common methods used for extracting drift signals include Principal Component Analysis (PCA),(6–8) Independent Component Analysis (ICA),(9) and Partial Least-Squares (PLS).(10) Aside from these methods, the Orthogonal Signal Correction (OSC) is superior to drift correction as shown in a recent study by Padilla *et al.*(11) Generally speaking, all the component correction methods aim to find one preferable direction to eliminate drift. However, the drift direction varies with time and gas type in the mapping space,(6) and consecutive groups of samples are needed for predicting each drift direction. Thus, a great deal of calculation is performed during the testing process. As a result, time-consuming calibrations are not suitable for on-line drift compensation, even if the component correction model is still effective for drift rejection.

The adaptive model can passively eliminate the drift effect. Frequently used methods to perform calibrations are on-line PCA,(12) adaptive resonance theory (ART),(13,14) and self-organizing maps (SOMs).(15,16) Recently, a new solution based on an evolutionary optimization has been demonstrated to be compatible with different classifiers.(17) The adaptive methods, in general, adjust the recognition models to suit the gas sensor signals with drift effect over time. Among these methods, the SOM network is a popular tool to classify the samples involving drift effect due to its single structure and simple computation. Yet, in the ordinary SOM network, neuron weights (codebook vectors) preserve prior knowledge of different gases through the training process. Once a calibration process starts, the preserved information for a certain type in codebook vectors may be erased by other types of sample. In other words, overtraining is barely avoided if the SOM neurons are arranged in one plane. Therefore, the multiple self-organizing map (MSOM) methodology was proposed to overcome the limitation.(18) Furthermore, the calibration rate of MSOM has been discussed to improve the network performance.(19)

According to current MSOM retraining steps in ref. (19), the network trains each map with its preserved retraining vector (RV). It means that the RVs should be the latest vectors to reflect the current samples with the drift effect. Unfortunately, this precondition cannot be held if the same type of gas is input ceaselessly. In this paper, we provide a novel retraining method to keep the RVs fresh for training all the codebook vectors. Compared with the conventional local retraining process and the ART network, the RV iterations for drift learning are considered to be performed to adapt the drift variations in time. Experimental results show the superiority of the new retraining method. Furthermore, the retraining frequency of this method is discussed to optimize the network performance. The rest of the paper is organized as follows. In § 2, we describe the methodology used in this study. The experimental setup, results, and analysis are presented in § 3, and in § 4, some conclusions are drawn.

2. Methods

2.1 *Self-organizing maps*

The SOM network has been proposed by Kohonen.(20) It is used as an unsupervised clustering tool that maps multidimensional data to two-dimensional space for visualization. Neurons with different codebook vectors are located as a node of planar grid. Once the input vector x comes, the Euclidean distances between the input vector and all the codebook vectors, which have the same dimensions as the input vector, will be calculated. The neuron that has the shortest Euclidean distance is considered as the winner that represents a certain category. Therefore, the input vectors can be categorized according to the rule: a closer location means a higher probability to be of the same type. The learning steps of SOM can be summarized as follows:

(1) Initialize a neighborhood $h(0)$ and random codebook vectors.

(2) Select the winning neuron as follows:

$$||x - w_{\mathrm{c}}|| = \min||x - w||, \tag{1}$$

where w is the weight of an arbitrary neuron, and w_{c} is the codebook vector of the winning neuron.

(3) Renew the weights as follows:

$$w = w + a{\cdot}h(t)(x - w), \tag{2}$$

where a ($a \in (0,1)$) is the learning rate that controls the similarity degree between the input vector and renewed codebook vectors. The function $h(t)$ defines the neighborhood that is monotonically decreasing during the training process.

(4) Stop when the training data is used up; otherwise, return to step (2).

2.2 *Multiple self-organizing maps*

A MSOM network includes several maps as many as the sample types.(21) Each map has a unique serial number. Once a testing sample comes, the winning neuron is determined according to the Euclidean distances between the input vector and all the codebook vectors in MSOM. The serial number of the winning map that implies a certain category is considered to be the output of the network. In other words, the network outputs are never changed if the winning neurons belong to the same map.

A supervised training is carried out after the weights have been initialized randomly. Each plane is trained by one type of sample. Thus, the training dataset is divided into several subsets on the basis of the target outputs from the training dataset. The detailed training strategy for each map is performed as described in § 2.1.

2.3 *Local retraining method*

A self-retraining process is added to complete the drift counteraction in the testing process.(18) The retraining process uses an unsupervised approach to reduce the cumulative uncertainty of gas sensors. It takes place at a fixed rate or flexible frequency.(19) Furthermore, the maximum quantization error for each map should be found in a former supervised training phase as follows:

$$e_i = \max\|x_i - w_i\|, \tag{3}$$

where e_i is the maximum quantization error for the ith map, x_i is the sample of the ith subset, and w_i is the weight of the neuron from the ith map.

During the testing phase, the distance of each unknown testing vector to the weight of the winning neuron should be computed. We consider that the winner belongs to the ith map. Then, the testing vector replaces the old RV of the ith map if the distance is smaller than e_i. Otherwise, the testing vector is dropped for the RV refresh process. Once the retraining process has been triggered, the RVs train their related maps using SOM training steps.

2.4 *Global retraining method*

The local retraining utilizes the preserved local information to track the newest odor observations with drift effect. However, the RV in the local retraining method may become useless if it has no updating for a long period of time. In other words, each RV should be refreshed at a proper rate to keep up with drift variations. Therefore, we develop a global retraining method to ensure that all the RVs are fresh. This method is carried out as follows:

(1) Input an observation and obtain its output.

(2) Calculate the drift effect vector Δd as follows:

$$\Delta d = x - x_{\mathrm{p}}^{i}, \tag{4}$$

where x_{p}^{i} denotes the RV of the ith map, x is the current observation that belongs to the ith map, and the global drift parameter Δd represents the drift variation at present.

(3) Iterations for all the RVs are implemented as follows:

$$\begin{aligned} x_{\mathrm{p}}^{i} &= x, \\ x_{\mathrm{p}}^{i} &= x_{\mathrm{p}}^{i} + b \cdot \Delta d, \quad i \neq j \end{aligned} \tag{5}$$

where b is the drift rate in the interval (0 1). The value of this factor depends on the fluctuating speed of the drift component. After these iterations, each RV is renewed to prevent losing the drift tracking ability.

(4) Adjust the weights of MSOM as a local retraining method once the retraining process has begun.

2.5 *Learning vector quantization refining*

After retraining, the refined process is needed to clear the boundaries between similar categories. The learning vector quantization (LVQ) scheme with the reward-punishment rule is used in the MSOM network. The neuron weights are refined according to the following steps:

(1) Find the winning neuron weight w_{c} and the second winning (nearest) neuron weight w_{s} for each RV x_{p}.

(2) If both the winner and second winner belong to the same map, the weight iterations perform as follows:

$$\begin{aligned} w_{\mathrm{c}} &= w_{\mathrm{c}} + \lambda \cdot (x_{\mathrm{p}} - w_{\mathrm{c}}), \\ w_{\mathrm{s}} &= w_{\mathrm{s}} + \lambda \cdot (x_{\mathrm{p}} - w_{\mathrm{s}}). \end{aligned} \tag{6}$$

(3) If the winner and second winner are found in different maps, an error rate of the network arises. Thus, to avoid unclear class boundaries, the weights update in distinguishing manners:

$$\begin{aligned} w_{\mathrm{c}} &= w_{\mathrm{c}} + \lambda \cdot (x_{\mathrm{p}} - w_{\mathrm{c}}), \\ w_{\mathrm{s}} &= w_{\mathrm{s}} + \lambda \cdot (x_{\mathrm{p}} - w_{\mathrm{s}}). \end{aligned} \tag{7}$$

where λ is the learning rate. To ensure the convergence, the initial value of the learning rate should be a small number and decrease monotonically.

2.6 *Adaptive resonance theory*

The main ART(13,14) structures include ART1, ART2, and ARTMAP. Among them, ART2 is suitable for on-line detections and analog inputs. A general ART2 network has two layers: comparison field (F_1) and category representation field (F_2), and performs its function as follows:

(1) Once an i-dimensional input vector comes, F_1 performs iterations to obtain a stable pattern using the following equations:

$$z_i = w_i + a \cdot u_i, \tag{8}$$

$$q_i = \frac{z_i}{e + \|z\|}, \tag{9}$$

$$v_i = f(q_i) + b \cdot f(s_i),\ f(x) = \begin{cases} 0,\ 0 \le x < \theta \\ x,\ x \ge \theta \end{cases}, \tag{10}$$

$$u_i = \frac{v_i}{e + \|v\|}, \tag{11}$$

$$s_i = \frac{p_i}{e + \|p\|}, \tag{12}$$

$$p_i = \begin{cases} u_i, j \ne j^* \\ u_i + d \cdot z_{ji}, j = j^* \end{cases}, \tag{13}$$

$$r_i = \frac{c \cdot p_i + u_i}{e + \|c \cdot p\| + \|u\|}, \tag{14}$$

where j is the number of neurons in F_2, and winning neuron j^* is chosen according to the inner product between p and the neuron weight w_{ij} (bottom to top) of F_2 by using a winner-take-all strategy. z_{ji}, which is called long-term memory (LTM), is the neuron weight (top to bottom) of F_2. e is a fixed tiny value, while a, b, c, and d are constant parameters that should be set before computations.

(2) Define ρ as a vigilance parameter. If $\|r\| > \rho$, the network selects j^* as output value, then go to step (4). Otherwise, disable the wining neuron and go to step (1) to find a new winner, then go to step (4). If all the F_2 neurons have been disabled, then go to step (3).

(3) Add a new neuron in F_2 and set it as a winner.

(4) Adjust z_{ji} and w_{ij} of the winning neuron as follows:

$$\frac{d}{dt} w_{ij} = d(1 - d) \cdot \left(\frac{u_j}{1-d} - w_{ij}\right),$$
$$\frac{d}{dt} z_{ji} = d(1 - d) \cdot \left(\frac{u_j}{1-d} - z_{ji}\right), \tag{15}$$

3. Results and Discussion

3.1 *Experimental setup*

The dataset used in this section is obtained from several experiments using the experimental platform shown in Fig. 1. We set two pumps here to control the gas injection and ejection: the standard gas samples enter the platform via pump 1 while pump 2 releases the gases from the chamber (the inner surface of which is plated with Teflon). The gas sensor array is composed of four metal oxide (MOX) gas sensors,

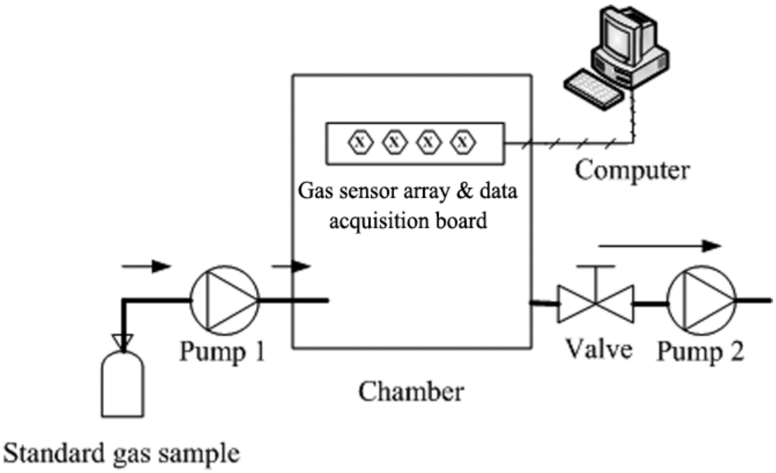

Fig. 1. Experimental platform.

namely, GSBT11, TGS2602, TGS2620, and TGS2201. Among these gas sensors, TGS2201 has dual-sensing elements, TGS2201A and TGS2201B, for gasoline and diesel exhausts, respectively; thus, five output signals are obtained from the gas sensor array. Moreover, the sensors are fixed on a printed circuit board (PCB) that contains an analog signal processing circuit and an ADC chip (TLC2543) controlled with a microprocessor (89C52). Twelve-bit digital signals are transferred to the upper monitor at a constant sampling rate (1 Hz). Additionally, the upper monitor is a computer wherein the transferred data are saved in ASCII text files for further processing.

Both carbon monoxide (CO) and formaldehyde (CHOH) samples are tested in different concentrations during a period of one month. Fifty tests were arranged in a specific time sequence, as shown in Table 1. Each test has 240 observations from baseline to steady state for a certain gas. Among these data, we have selected 50 consecutive baseline samples and 100 consecutive gas samples. Totally, 7500 samples were acquired for drift compensation analysis. The samples of the earliest tests for CO and CHOH are used as a training dataset (300 samples). The remaining 7200 observations are used as testing samples to evaluate the drift compensation methods. The responses of the gas sensor array are shown in Fig. 2, where a strong drift influence exists in terms of the tendency of the response values.

The MSOM network here is composed of three maps corresponding to the different gas states in the training dataset: the first map is related with the baseline state, the second one represents the CO samples, and the third one corresponds to the CHOH samples. Each planar map has a uniform structure of 4 × 4 size. The learning rate and neighborhood of the network decrease exponentially, and the other parameters are summarized in Table 2. Furthermore, constant parameters of the ART2 network are determined according to Table 3.

Table 1
Dataset arrangement.

Testing sequence	Sample type	Number of samples
1–10 (first phase)	Carbon monoxide	1500
11–25 (second phase)	Formaldehyde	2250
26–50 (third phase)	Carbon monoxide	3750

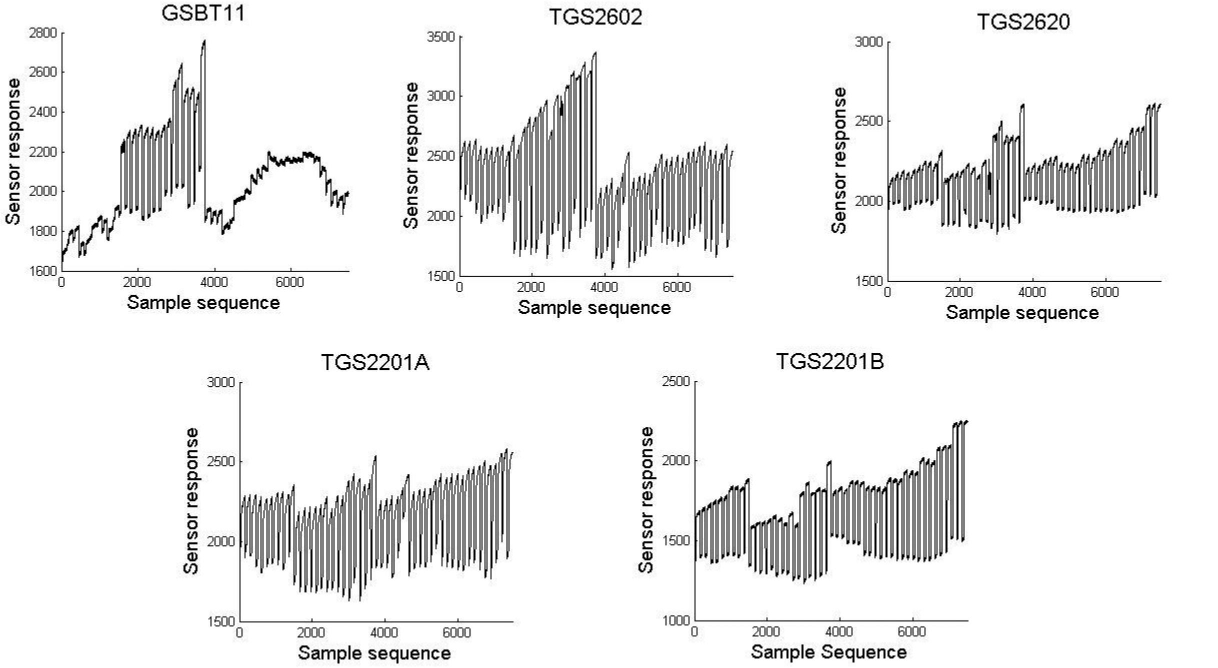

Fig. 2. Responses of the gas sensor array.

Table 2
MSOM parameters.

Parameter	Learning rate	Neighbor radius	Training epoch	Retraining epoch	Retraining frequency	Refining learning rate	Drift rate
Value	0.1	6	20	1	1	0.1	0.4

Table 3
ART2 parameters.

Parameter	*a*	*b*	*c*	*d*	*ρ*	*θ*
Value	10	10	0.1	0.9	0.9997	0.01

3.2 *Evaluation of methods*

In this section, we evaluate the performance of three self-training methods, including MSOM with the global retraining (MSOM1), MSOM with the local retraining (MSOM2), and ART2, and the dataset that we used has been described in the above section. To restrict the response scale and enhance the similarity of homogeneous samples with different concentrations, the dataset (5-dimensional space) is normalized in the range [0, 1]. We consider that the output values 1, 2, and 3 of the MSOMs denote baseline, CO, and CHOH, respectively. Furthermore, the same output values of the ART2 network denote the same type of sample. To compare the results of the three methods, the true MSOM outputs corresponding to the dataset are given in Fig. 3(a), and Figs. 3(b)–3(d) demonstrate the recognition results of the three methods, and the recognition rates are summarized in Table 4.

We note that the MSOM1 output values strictly equal to the true ones, whereas the MSOM2 results become useless in the third phase of the experiment. The reason is that all the CO samples in the third phase are recognized as baseline by using the MSOM2 network. This issue is caused by the local retraining method itself. According to the method, the RV has a renewed opportunity only when the related samples appeared as input vectors. Therefore, no RV updating has taken place for the second map in the second phase (CHOH testing). As a result, the RV for the CO observations cannot keep up with the current CO samples with the drift effect. In the other version, the RV is out of date for losing the information about drift over a period of time. Thus, the MSOM2 network is defective under the experimental conditions in our study. The global retraining method, on the other hand, renews all the retraining vectors if anyone of them has been replaced. Thus, all the retraining vectors catch up with the drift signals at each retraining time. Figure 3(d) gives the clustering outcomes of the ART2 network. In the first two phases, the network separates the CO samples, the most baseline and CO samples successfully. However, the ART2 network cannot compensate for all the LTMs during retraining; thus, the network recognizes the baseline samples as a new type of gas in the last stage, and this causes the recognition rate of baseline samples to drop to 0% dramatically. Finally, only the MSOM1 network based on the global retraining method recognizes all the 7200 testing observations correctly.

3.3 *Refresh rate optimization*

For the sake of simplicity, the retraining frequency is kept constant (1 Hz) in the above discussion. It means that the retraining process will be triggered at each time the testing sample enters. However, it is time consuming under practical conditions, although we have obtained excellent results. In contrast, a long time interval between two self-retraining processes may deteriorate the network capability. A compromise, as a result, is needed to balance the time cost and network performance. The relationship between network accuracy with the global retraining method and the retraining intervals is shown in Fig. 4.

From Fig. 4, one can see that the error rate is larger with a large retraining interval than with a small retraining interval. We prefer to find a range of retraining intervals with steady and small error rates. Thus, the retraining intervals in both [1 70] and

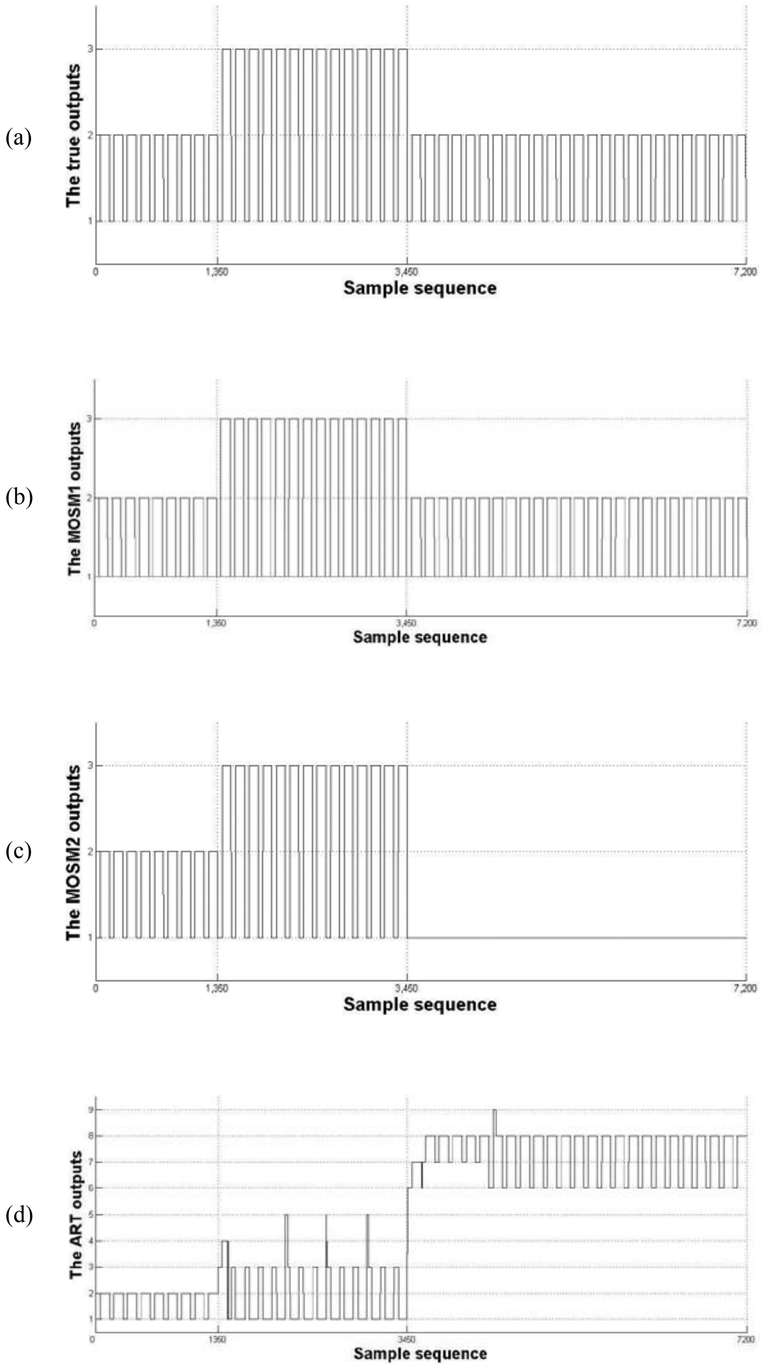

Fig. 3. Network outputs: (a) true outputs, (b) MSOM1 outputs, (c) MSOM2 outputs, and (d) ART2 outputs.

Table 4
Method evaluation results.

Testing sequence		MSOM1 recognition rate	MSOM2 recognition rate	ART2 recognition rate
First phase (CO testing)	Baseline samples	100%	100%	100%
	CO samples	100%	100%	100%
Second phase (CHO testing)	Baseline samples	100%	100%	92.9%
	CHO samples	100%	100%	89.9%
Third phase (CO testing)	Baseline samples	100%	100%	0%
	CO samples	100%	0%	94.8%

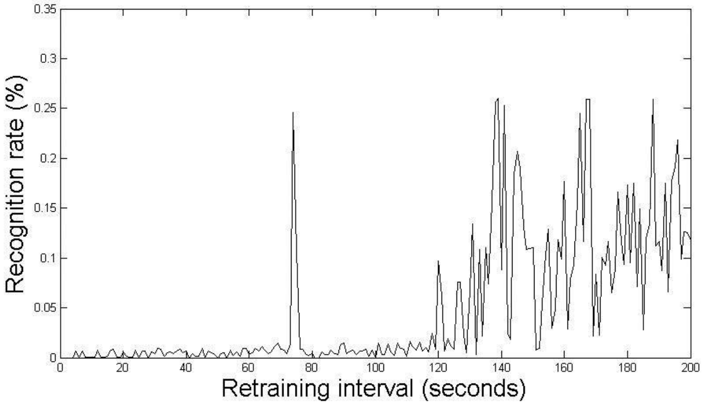

Fig. 4. Relationship between error rates and retraining intervals.

[80 110] are suited to keep the error rates below 2%, whereas other retraining intervals do not suit our requirement. Furthermore, the optimized retraining interval should be as large as possible to avoid low time efficiency. Therefore, the retraining interval of about 110, which approximates to 0.01 Hz, is the most suitable choice for our present network with the global retraining method.

4. Conclusions

In this study, a new global retraining method has been proposed to keep the right RVs related to the drift effect. This method renews the preserved RVs with different strategies: the latest observation replaces the corresponding RV directly, whereas the estimated drift vector updates the other RVs. The global retraining method shows its superiority in experimental data analysis for long-time detection involving the drift effect compared with the normal local retraining and ART2 approaches. Additionally, the possible retraining rates have been discussed to optimize the network performance. Further studies will focus on the automatic parameter optimization under on-line working conditions.

Acknowledgements

This work was supported by the Natural Science Foundation Project of CQ CSTC (2012jjA40005), the Natural Science Foundation Project of CQ CSTC (2010BB2242), 2010, P. R. China, and the Fundamental Research Funds for Central Universities (No. CDJRC10160010).

References

1 N. El Barbri, J. Mirhisse, R. Ionescu, X. Correig, B. Bouchikhi and E. Llobet: Sens. Actuators, B **141** (2009) 538.
2 K. Yu, Y. Wang, J. Yu and P. Wang: Sens. Lett. **9** (2011) 876.
3 D. V. Saverio, P. Marco and M. Luca: Sens. Actuators, B **143** (2009) 182.
4 F. Hossein-Babaei and V. Ghafarinia: Sens. Actuators, B **143** (2010) 641.
5 J. Haugen, O. Tomic and K. Kvaal: Anal. Chim. Acta **407** (2000) 23.
6 A. Ziyatdinov, S. Marcoe, A. Chaudry, K. Persaud, P. Caminal and A. Perera: Sens. Actuators, B **146** (2010) 460.
7 O. Tomic, T. Eklöv, K. Kvaal and J. Haugen: Anal. Chim. Acta **512** (2004) 199.
8 T. Artursson, T. Eklov, I. Lundstrom, P. Martensson, M. Sjostrom and M. Holmberg: J. Chemom. **14** (2000) 711.
9 M. Kermit and O. Tomic: IEEE Sens. J. **3** (2003) 218.
10 R. Gutierrez-Osuna: Electronic Noses and Olfaction 2000 (Institute of Physics Publishing, Brighton, 2000) p. 147.
11 M. Padilla, A. Perera, I. Montoliu, A. Chaudry, K. Persaud and S. Marco: Chemom. Intell. Lab. Syst. **100** (2010) 28.
12 M. Paniagua, E. Llobet, J. Brermes, X. Vilanova, X. Correig and E. L. Hines: Electron. Lett. **39** (2003) 40.
13 C. Distante, P. Siciliano and L. Vasanelli: Sens. Actuators, B **69** (2000) 248.
14 D. Vlachos, D. Fragoulis and J. Avaritsiotis: Sens. Actuators, B **45** (1997) 223.
15 S. Marco, A. Ortega, A. Pardo and J. Samitier: IEEE Trans. Instrum. Meas. **47** (1998) 316.
16 C. Di Natale, A. Macagnano, A. D'Amico and F. Davide: Meas. Sci. Technol. **8** (1997) 1236.
17 S. Di Carlo, M. Falasconi, E. Sanchez, A. Scionti, G. Squillero and A. Tonda: Pattern Recognit. Lett. **32** (2011) 1594.
18 C. Distante, P. Siciliano and K. C. Persaud: Pattern Anal. Appl. **5** (2002) 306.
19 M. Zuppa, C. Distante, P. Siciliano and K. C. Persaud: Sens. Actuators, B **98** (2004) 305.
20 T. Kohonen: Neurocomputing **21** (1998) 1.
21 E. Cervera and A. P. del Pobil: Neurocomputing **16** (1997) 309.