Gas sensor drift mitigation using classifier ensembles

Conference Paper · January 2011

DOI: 10.1145/2003653.2003655

6 authors, including:

Tuba Ayhan
KU Leuven
21 PUBLICATIONS **341** CITATIONS
SEE PROFILE

Ramón Huerta
Capital Group
181 PUBLICATIONS **6,513** CITATIONS
SEE PROFILE

M.L. Homer
NASA
68 PUBLICATIONS **1,125** CITATIONS
SEE PROFILE

Gas Sensor Drift Mitigation using Classifier Ensembles

Alexander Vergara
BioCircuits Institute
University of California, San Diego
La Jolla, CA 92093, USA
vergara@ucsd.edu

Tuba Ayhan
Department of Electronics and Communication Engineering
Technical University of Istanbul
Maslak, TR-34469, Istanbul, Turkey
ayhant@itu.edu.tr

Shankar Vembu
BioCircuits Institute
University of California, San Diego
La Jolla, CA 92093, USA
svembu@ucsd.edu

Ramón Huerta
BioCircuits Institute
University of California, San Diego
La Jolla, CA 92093, USA
rhuerta@ucsd.edu

Margaret Ryan
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive,
Pasadena, CA 91109, USA
maryan@uchicago.edu

Margie Homer
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive,
Pasadena, CA 91109, USA
margie.l.homer@jpl.nasa.gov

ABSTRACT

Sensor drift remains to be one of the challenging problems in chemical sensing. To address this problem we collected an extensive data set for six different volatile organic compounds over a period of three years under tightly-controlled operating conditions using an array of 16 metal-oxide sensors. We then adopted a machine learning approach namely an ensemble of classifiers to cope with sensor drift. For this particular application we chose support vector machine as our base classifier in the ensemble but, in principle, any other classifier can be used. Experiments clearly indicate the presence of drift in the sensors during the period of three years and that it degrades the performance of classifiers. However, the ensemble method that uses a weighted combination of classifiers trained at different points of time is able to cope well with sensor drift.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Concept learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; I.5.1 [**Pattern Recognition**]: Models—*Deterministic*

General Terms

Algorithms, Design, Experimentation

Keywords

sensor drift, chemical sensing, time series classification, ensemble methods, support vector machines

SensorKDD'11, August 21, 2011, San Diego, CA, USA.

1. INTRODUCTION

The gradual and unpredictable change of sensor responses or the so-called *sensor drift* has long been recognized as one of the major challenges to develop electronic noses [9, 5, 25]. Random changes in the sensor responses deteriorates the performance of classifiers used for recognition of gases thereby increasing the maintenance costs of artificial noses during real-time operation. There are several sources of sensor drift. In [11], the sensor drift is categorized into two main types: *real drift* due to aging and poisoning, and *second-order drift* due to recording protocols, hysteresis, humidity and temperature changes. The most systematic analysis of long term drift was done in [19, 20] with data sets extending for long periods of time in real operating conditions. Among the many conclusions drawn from that project, the following was emphasized: metal-oxide based (Figaro type [3]) gas sensors remain the best option for long term applications in continuous monitoring systems. The authors also emphasized the need for drift compensation using a reference/calibration gas and concluded that sensor replacement is unavoidable.

The standard approaches used in the chemical sensing community to cope with sensor drift are univariate methods where drift compensation is performed for each sensor individually and multivariate methods where the drift is corrected for the entire sensor array. The reader is referred to [11] and references therein for a survey on drift compensation methods. Among the multivariate drift compensation methods, (unsupervised) component correction techniques are popular [5, 10, 24]. These techniques basically rely on finding linear transformations that normalize the sensor responses across time so that a classifier can be directly applied to the resulting stationary data. For instance, the component correction method of [5] applies the following transformation to the measurement/data matrix $X \leftarrow X - (X \cdot c)c^\top$ where c is the principal component vector(s) of the measurements computed using a reference gas that hopefully

approximates the drift direction. The main drawback of these techniques is that they assume the drift direction to be linear in the feature space and therefore a linear transformation of the data suffices to correct it. While it is entirely plausible that kernelized versions of component analysis such as kernel principal component analysis [21] can be applied to account for non-linearities in the drift direction, these techniques have not been investigated much in the chemical sensing community. Also, with the exception of [24], these techniques require a reference gas that is used to approximate the drift direction; in other words, they assume that the drift direction in the reference gas explains the drift in all the other gases.

In this paper, we take a completely different approach to solve the problem where we do not make any of the above assumptions. We use a supervised machine learning method, namely, an ensemble of classifiers to cope with sensor drift. To the best of our knowledge, such a machine learning approach that automatically detects and copes with sensor drift has not been applied in the chemical sensing community although it has been shown to yield promising results on problems with drifting concepts in machine learning and data mining [23, 12, 13, 14]. It is important to note that the ensemble method used in this paper *complements* existing component correction methods mentioned above; since component correction methods are essentially a pre-processing technique, we can indeed use the ensemble method on the pre-processed data too.

In the following sections, we will first describe the experimental setup, the data set, and the feature extraction methods (Section 2). We will then describe the drift compensation algorithm (Section 3). Finally, we will present our experimental findings (Section 4) and conclude (Section 5).

2. DATA AQUISITION

We applied our drift compensation method on an extensive data set[1] recorded by a metal-oxide gas sensor array. In this section, we describe the experimental setup, the recording protocol, and the signal processing algorithms used for feature extraction.

2.1 The experimental setup

We used an array of 16 screen-printed commercially available metal-oxide semiconductor gas sensors manufactured and commercialized by Figaro Inc. [3]. The sensors include an independently-controlled RuO_2 (Ruthenium Oxide) electrical heating line and a metal oxide semiconductor film as a sensor material printed onto the measuring electrodes (noble metal). The sensor element obtained is mounted onto an alumina substrate and then connected by lead wires to the pins of the sensor package. The resulting array has four types of sensors tagged as TGS2600, TGS2602, TGS2610, TGS2620. We placed the resulting array (four for each type, 16 sensors in total) into a 60ml volume test chamber where the odorants of interest in gaseous form are injected for trials. To generate the required data set, we connected the test chamber to a computer-controlled continuous flow system, which provides versatility for conveying the chemical compounds of interest at the desired concentrations to the sensing chamber with high accuracy in a highly reproducible way while keeping the total flow constant. In particular, our system utilizes three mass flow controllers (MFCs) (provided by Bronkhorst High-Tech B.V. [2]), each of them with different maximum flow levels. These devices connect to different pressurized gas cylinders diluted in dry air, which contain either the carrier gas or the odorants to be measured. To maintain the moisture level at 10% relative humidity (measured at $25^oC \pm 1^oC$) during the entire measurement process, we utilize synthetic dry air as background for all measurements, provided by Airgas Inc. [1]. Then, the analytes under analysis (ammonia, acetaldehyde, acetone, ethylene, ethanol, toluene) are added to this background in random order. The total flow rate across the sensing chamber was set to 200 ml/min and kept constant for the whole measurement process. The response of the gas sensor array was measured when the operating temperature of sensors was fixed at 450^oC. Finally, to ensure that reproducible response patterns were acquired during each measurement, the sensors were pre-heated for several days prior to the experiment.

The sensor response is read-out in the form of the resistance across the active layer of each sensor; hence each measurement produced a 16-channel time series sequence. The data acquisition board collected the data from the gas sensors and controlled the analog voltage signal to every sensor heater. This voltage is used to control and vary the sensor heater's operating temperature using the LabVIEW [4] environment running on a PC platform. The experimental setup is illustrated in the diagram shown in Figure 1.

2.2 The data set

Our data set consists of a six-gas/analyte problem delivered at different concentrations. The odor identity and concentration values in parts-per-million by volume (ppmv) are listed in Table 1. Each of the possible pairs of gas type and concentration was sampled in no particular order. The resulting data set consists of $13,910$ recordings (time series sequences) collected over a period of 36 months. The exact distribution of the number of measurements per month in shown in Table 2. Notice that some of the months shown in the table does not contain any measurements for one or more gases. The reason for this is that other analytes and complex mixtures that are not considered in this study were collected during that time using the same sensor array and experimental apparatus. Note that this increases the complexity of the problem due to interefence from these external analytes that may potentially affect the sensor life.

As Table 2 indicates, the last batch of recordings, which contains 3600 measurements from the same analytes, was collected 5 months later and after the sensors were powered off during this period. This is particularly important not only because it allows us to validate the suggested method on a set of measurements collected 5 months later, but also because it is during this 5-month period that the sensors were subject to severe contamination, since external interferents could easily get attached to the sensing layer due to the lack of the operating temperature.

To generate the data set, we followed a measurement procedure consisting of the following steps. First, a constant flow of zero-grade dry air was circulated through the sensing chamber while the gas sensor array was kept at a stable operating temperature (450^oC). This was done to measure the baseline steady-state sensor response (i.e., the sensor response in presence of no chemical analytes). Next, the desired concentration of the odorant was injected by the con-

[1]The data set is available upon request.

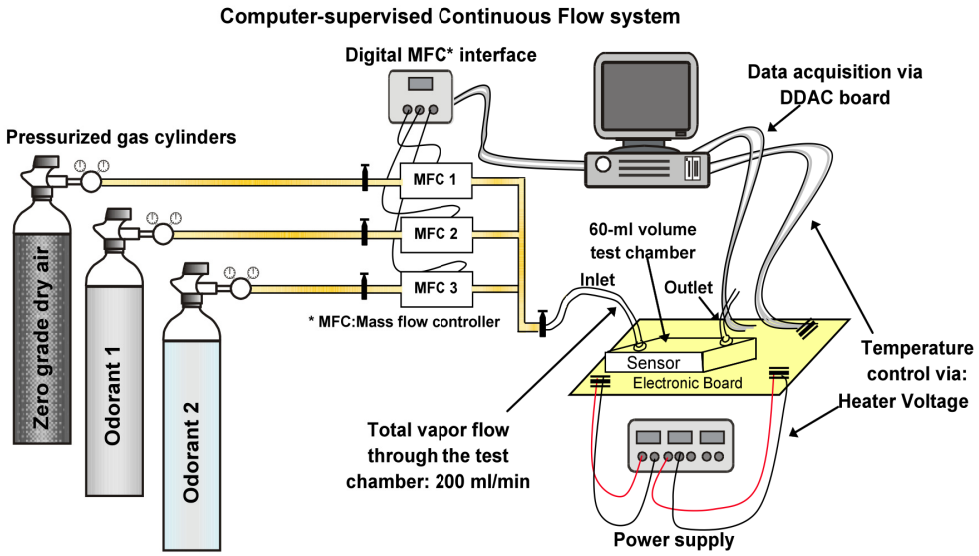

Figure 1: Experimental setup used for data acquisition. The sensor responses are recorded in the presence of the analyte in gaseous form diluted at different concentrations in dry air. The system operates under a fully computerized environment with minimal human intervention. Therefore, no changes in the flow or flow dynamics are reflected in the sensor response. Moreover, since the system is continuously supplying gas to the sensing chamber (either clean dry air or a chemical component), the amount of gas molecules in the sensing chamber is homogeneously distributed.

Table 2: Data set details. Each row corresponds to samples collected during a period of one month for six gases.

Month ID	Number of examples						
	Ammonia	Acetaldehyde	Acetone	Ethylene	Ethanol	Toluene	Total
month1	76	0	0	88	84	0	248
month2	7	30	70	10	6	74	197
month3	0	0	7	140	70	0	217
month4	0	4	0	170	82	5	261
month8	0	0	0	20	0	0	20
month9	0	0	0	4	11	0	15
month10	100	105	525	0	1	0	731
month11	0	0	0	146	360	0	506
month12	0	192	0	334	0	0	526
month13	216	48	275	10	5	0	554
month14	0	18	0	43	52	0	113
month15	12	12	12	0	12	0	48
month16	20	46	63	40	28	0	197
month17	0	0	0	20	0	0	20
month18	0	0	0	3	0	0	3
month19	110	29	140	100	264	9	652
month20	0	0	466	451	250	458	1625
month21	360	744	630	662	649	568	3613
month22	25	15	123	0	0	0	163
month23	15	18	20	30	30	18	131
month24	0	25	28	0	0	1	54
month30	100	50	50	55	61	100	416
month36	600	600	600	600	600	600	3600

Table 1: Analytes and concentrations in the data set.

Analytes	Concentrations in ppmv
Ammonia	50, 60, 70, 75, 80, 90, 100, 110, 120, 125, 130 140, 150, 160, 170, 175, 180, 190, 200, 210 220, 225, 230, 240, 250, 260, 270, 275, 280 290, 300, 350, 400, 450, 500, 600, 700, 750 800, 900, 950, 1000
Acetaldehyde	5, 10, 13, 20, 25, 30, 35, 40, 45, 50, 60, 70 75, 80, 90, 100, 120, 125, 130, 140, 150, 160 170, 175, 180, 190, 200, 210, 220, 225, 230 240, 250, 275, 300, 500
Acetone	12, 25, 38, 50, 60, 62, 70, 75, 80, 88, 90, 100 110, 120, 125, 130, 140, 150, 170, 175, 180 190, 200, 210, 220, 225, 230, 240, 250, 260 270, 275, 280, 290, 300, 350, 400, 450, 500 1000
Ethylene	10, 20, 25, 30, 35, 40, 50, 60, 70, 75, 90, 100, 110, 120, 125, 130, 140, 150, 160, 170, 175, 180, 190, 200, 210, 220, 225, 230, 240, 250, 275, 300
Ethanol	10, 20, 25, 30, 40, 50, 60, 70, 75, 80, 90, 100, 110, 120, 125, 130, 140, 150, 160, 170, 175, 180, 190, 200, 210, 220, 225, 230, 240, 250, 275, 500, 600
Toluene	10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100

tinuous flow system into the sensing chamber. Finally, in the third step (cleaning phase) the vapor was vacuumed away from the sensor array and the test chamber was cleaned with dry air before the concentration phase of a new measurement (for at least 10 minutes). The acquisition of these measurements took at least 300 seconds to complete, divided into 100 seconds for the gas injection phase and at least 200 seconds for the recovery (cleaning phase). For processing purposes, we considered the whole sensor response after subtracting the baseline from each record. The sampling rate was set to 100 Hz. Finally, the measurement process herein described was replicated for subsequent measurements.

2.3 Data processing and feature extraction

The Figaro metal-oxide gas sensors are known to have a slow response to a chemical analyte. This response, under tightly-controlled operating conditions (i.e., constant air flow and fixed operating temperature), typically involves a monotonically saturating smooth change in the conductance/resistance across its sensing layer due to the adsorption/desorption reactions of the chemical analyte occurring at the micro-porous surface of the sensor. The amount and speed of these reactions depends on (i) the analyte identity, (ii) the analyte concentration, (iii) the active layer (i.e., sensor type), and (iv) the surface temperature (i.e., the sensors' operating temperature). Since the last two factors are fixed throughout the entire measurement procedure of this analysis, the sensor-analyte identity/concentration interaction process becomes the only factor that, as a pair, shapes the response profile, and thus, that defines the identity of the chemical analyte of interest [22]. Accordingly, features reflecting all the sensing dynamics at the sensor surface are of special interest in our drift compensation analysis.

Feature extraction plays an important role in every chemosensory application [17]. It is defined as a transformation mapping the sensor response to a space of lower dimension preserving the most meaningful portion of information contained in the original sensor signal. In this work, we consider two distinct types of features that exploit all dynamic processes occurring at the sensor surface, including the ones that will reflect its adsorption, desorption, and steady-state (or final) responses of the sensor element. On one hand, we utilize the steady-state feature, which is the "gold-standard" for chemo-sensory feature extraction [15]. It is defined as the difference of the maximal resistance change and the baseline,

$$\Delta R = \max_k r[k] - \min_k r[k], \tag{1}$$

and its normalized version expressed by the ratio of the maximal resistance and the baseline values,

$$||\Delta R|| = \frac{\max_k r[k] - \min_k r[k]}{\min_k r[k]}, \tag{2}$$

where $r[k]$ is the time profile of sensor resistance, k is the discrete time indexing the recording interval $[0, T]$ when the chemical vapor is present in the test chamber.

On the other hand, we also extract transient features, namely the exponential moving average (ema_α), which is an aggregate of a number of standard features that we use to evaluate effectively the transient portion of the sensor response during the entire measurement procedure under controlled flow conditions,

$$y[k] = (1-\alpha)y[k-1] + \alpha\left(r[k] - r[k-1]\right), \tag{3}$$

with $k = 1, 2, \ldots, T$, $y[0] = 0$, and the scalar $0 < \alpha < 1$ is the smoothing parameter. ema_α is defined for a given discrete-time signal $r[\cdot]$ by the maximum gradient value of the transform. For different values of α, the magnitude of this peak and its exact location (in time) varies. In particular, we set three different values for α ($\alpha = 0.1$, $\alpha = 0.01$, and $\alpha = 0.001$) as already used in [16] starting from the recorded rising portion, and three additional features with the same α values in the decaying portion of the sensor response. Figure 2 shows the typical signal response of a chemical sensor in presence of 30 *ppmv* of Acetaldehyde and its ema_α representation for the three different α values. Then, by applying this transformation to each one of the 16 channels in the recorded time-series, we map the sensor array response to a 128-dimensional feature vector, which resulted from the combination of the 8 features described above $\times$ 16 sensors (see Table 3). The readers are referred to [16] for a more detailed discussion on these features.

3. DRIFT COMPENSATION METHOD

We use an ensemble of classifiers [23, 13, 14] to detect and cope with sensor drift. Consider a binary classification problem with a set of features x as inputs and a class label (a gas/analyte in our problem) y as output. At every time step t, we receive a batch of examples $S_t = \{(x_1, y_1), \ldots, (x_{m_t}, y_{m_t})\}$ of size m_t. We train a classifier $f_t(x)$, for example a support vector machine (SVM) [7], using the current batch of examples. The final classifier $h_{t+1}(x)$ at time step $(t+1)$ is a weighted combination of classifiers, i.e, $h_{t+1}(x) = \sum_{i=1}^{t} \beta_i f_i(x)$,

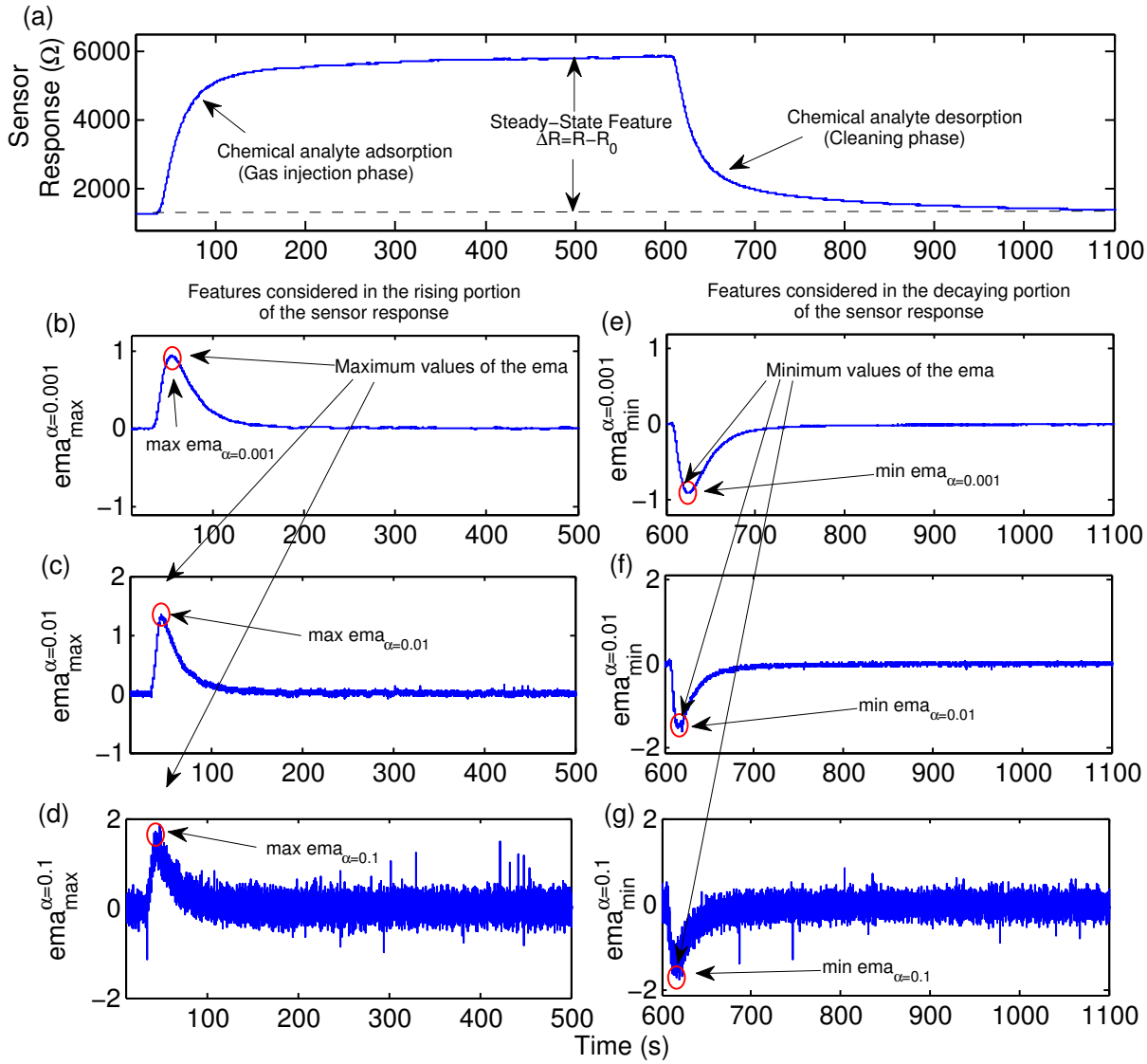

Figure 2: (a) Typical response of a metal-oxide based chemical sensor to 30 $ppmv$ of Acetaldehyde. (b), (c), (d) Exponential moving average of the rising portion of the sensor response (gas injection) for $\alpha = 0.1$, $\alpha = 0.01$, and $\alpha = 0.001$. The maximum values of the plots, (i.e., $max_k ema_\alpha r[k]$), represent the features extracted of the chemical sensor in response to the chemical analyte. (e), (f), (g) Exponential moving average of the decaying portion of the sensor response (cleaning phase) for $\alpha = 0.1$, $\alpha = 0.01$, and $\alpha = 0.001$. The minimum values of the plots, (i.e., $min_k ema_\alpha r[k]$), are the features we extracted to represent the signal sensor during its cleaning phase. Features are marked with red circles in the plot.

Table 3: Features extracted from the time series data.

Steady-State features	Transient features	
	rising portion	decaying portion
ΔR	$MAXema_{\alpha=0.001}$	$MINema_{\alpha=0.001}$
$\|\|\Delta R\|\|$	$MAXema_{\alpha=0.01}$	$MINema_{\alpha=0.01}$
	$MAXema_{\alpha=0.1}$	$MINema_{\alpha=0.1}$

where $\{\beta_1, \ldots, \beta_t\}$ is the set of classifier weights. Under the assumption that the distribution of examples in the current batch S_t has not changed much from those in the previous batch S_{t-1}, we use the examples in batch S_t to estimate the weights $\{\beta_1, \ldots, \beta_t\}$. There are several ways to estimate these weights. A simple and intuitive way is to assign weights to classifiers according to their prediction performance on batch S_t. Alternatively, we could solve an optimization problem such as

$$\underset{\beta_1,\ldots,\beta_t}{\operatorname{argmin}} \sum_{i=1}^{m_t} \left(\sum_{j=1}^{t} \beta_j f_j(x_i) - y_i \right)^2 . \quad (4)$$

It is also possible to minimize a different loss function than the squared loss used above. For example, if the base classifiers are SVMs, we can minimize the hinge loss $L(f(x), y) = \max(0, 1 - yf(x))$. Minimizing the squared loss as shown above has the advantage of computing the solution $\{\beta_1, \ldots, \beta_t\}$ in closed form. The algorithm in its most general form is given below.

Algorithm 1 Algorithm to cope with concept drift

1: **for** $t = 1, \ldots, T$ **do**
2: Receive $S_t = \{(x_1, y_1), \ldots, (x_{m_t}, y_{m_t})\}$
3: Train a classifier (SVM) on S_t
4: Estimate the weights $\{\beta_1, \ldots, \beta_t\}$ using one of the techniques described in the text
5: **end for**
6: Output final classifier: $\{\beta_1, \ldots, \beta_T\}$ and $\{f_1, \ldots, f_T\}$

For multiple classes, we can use one-vs-one or one-vs-all strategy [18] to train a classifier for every pair of classes or for every class respectively. Let L be the number of classes. We can estimate a set of weights $\{\beta_1^c, \ldots, \beta_t^c\}$ for every class $c \in \{1, \ldots, \binom{L}{2}\}$ or $c \in \{1, \ldots, L-1\}$ by solving (4). Alternatively, if we want to assign weights according to the prediction performance of classifiers on the most recent batch, we can simply estimate a single set of weights $\{\beta_1, \ldots, \beta_t\}$ using the multi-class classifier prediction accuracies on every batch. In this case, predictions are made according to a weighted majority voting, i.e.,

$$h_{t+1}(x) = \underset{y \in \{1,\ldots,L\}}{\operatorname{argmax}} \sum_{t: f_t(x) = y} \beta_t \ . \tag{5}$$

4. EXPERIMENTAL RESULTS

In all our experiments, we trained multi-class SVMs (one-vs-one strategy) with RBF kernel using the publicly available LibSVM software [6]. The kernel bandwidth parameter γ and the SVM C parameter were chosen using 10-fold cross validation by performing a grid search in the range $[2^{-10}, 2^{-9}, \ldots, 2^4, 2^5]$ and $[2^{-5}, 2^{-4}, \ldots, 2^9, 2^{10}]$ respectively.

We first established the fact that the sensors are drifting and that the drift is degrading the performance of classifiers. We trained a multi-class classifier on data collected during the first two months and tested it on data from the remaining months. Details on the number of measurements collected during each month for a period of three years is given in Table 2. The classifier's performance measured in terms of prediction accuracy is shown in Figure 3 (left). We see that the performance gradually degrades with time; this serves as a clear indicator of sensor drift and the way it has affected the accuracy of the classifier. On month 18 (ID: month18, in Table 2), we have only 3 measurements from class 2 and therefore the performance suddenly rises to 100%. We consider this to be an exception.

We then considered three settings as described below:

Setting 1 For every month, we trained a multi-class classifier with data from *only* the previous month and tested it on the current month.

Setting 2 For every month, we trained an ensemble of multi-class classifiers using Algorithm 1.

Setting 3 Same as Setting 2 but with uniform weights on the individual (base) classifiers.

In order to be able to train classifiers under these settings, we need sufficient number of examples in each class and month.[2]. We therefore combined measurements from 36 months to form 10 batches in such a way that the number of measurements were as uniformly distributed as possible. Details on the number of measurements in each batch is given in Table 4. Figure 3 (right) shows the performance of an SVM (black line) trained on batch 1 and tested on batches 2 through 10. Note that this is the same SVM model used in Figure 3 (left) but tested on data from batches instead of months. Once again, we see that the performance of the classifier is degrading with time due to drift. We found similar behavior when we trained several SVMs on batches 2 through 5 and tested them on successive batches. These results are again shown in Figure 3 (right). The complete set of results, i.e., the accuracy of classifiers trained on batches 1 through 9 and tested on successive batches, is given in Table 5. The first five rows in this table correspond to the results plotted in Figure 3 (right). These classifiers were eventually used in our ensemble method with appropriate weights. Figure 4 shows the performance of classifiers under settings 1, 2 and 3 described above. As expected, classifiers trained under setting 1, i.e., using the most recent batch of examples, perform better than the classifier trained with data from only batch 1. We believe the drop in accuracy when testing on batches 5 and 6 using classifiers trained on batches 4 and 5 respectively is an artefact of the data set. Notice that batch 5 has no examples in class *Toluene* whereas batch 6 has 467 examples. Indeed, we found that the accuracy on batch 6 was increased by 10% when we removed examples from class *Toluene*. Still the accuracy is significantly low when compared to batches 4, 6 and 7. A possible explanation is that the distribution of data in batch 5 is very different from the data in batches 4, 6 and 7 thereby resulting in a drop in classifier performance when experimented under Settings 1 and 2. Interestingly, the classifiers trained under Setting 1 were able to cope with drift to some extent and we believe they are a natural and strong baseline for any drift-correcting machine learning algorithm to compare against. As shown in the figure, the classifier ensembles were able to perform better than these baseline classifiers when tested on most of the batches with significant improvements

Table 4: Data set details. Each row corresponds to months that were combined to form a batch.

Batch ID	Month IDs
batch1	month1, month2
batch2	month3, month4, month8, month9, month10
batch3	month11, month12, month13
batch4	month14, month15
batch5	month16
batch6	month17, month18, month19, month20
batch7	month21
batch8	month22, month23
batch9	month24, month30
batch10	month36

[2]Note that this requirement is not necessary if we train classifiers in a *purely online* fashion [13, 14], but in this paper we focus only on batch training using SVMs.

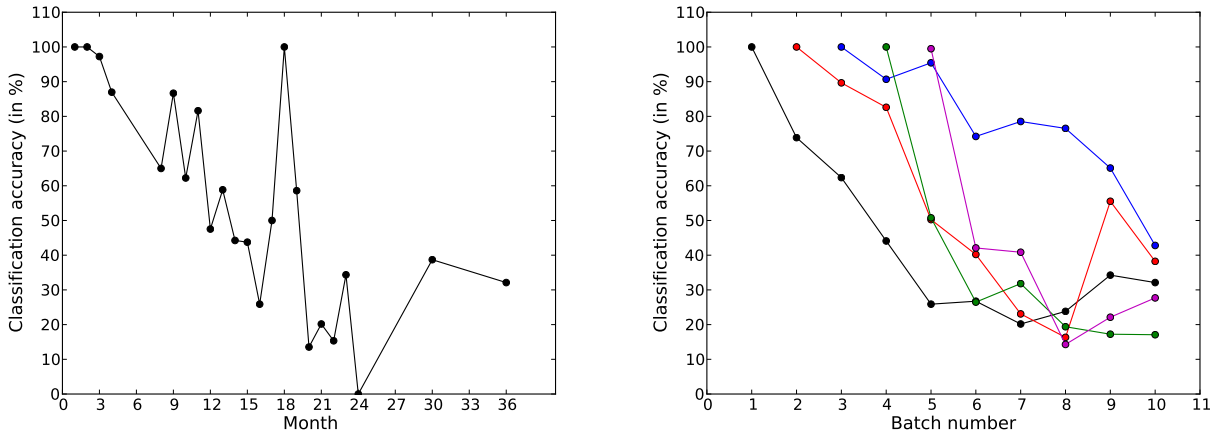

Figure 3: ***Left:*** **Performance of classifiers when trained on months 1 and 2 and tested on months 3 through 30.** ***Right:*** **Performance of classifiers trained on batches 1 through 5 and tested on successive batches.**

Table 5: Performance of classifiers trained on batches 1 through 9 and tested on successive batches.

	Classification accuracy (in %) on batches 2 through 10								
Batch ID	2	3	4	5	6	7	8	9	10
batch1	73.87	62.36	44.10	25.89	26.74	20.18	23.81	34.26	32.11
batch2		89.66	82.61	50.25	40.22	23.08	16.33	55.53	38.22
batch3			90.68	95.43	74.22	78.52	76.53	65.11	42.81
batch4				50.76	26.48	31.81	19.39	17.23	17.08
batch5					42.09	40.85	14.29	22.13	27.69
batch6						86.63	90.48	64.47	55.72
batch7							89.46	75.96	64.17
batch8								60.64	21.39
batch9									37.03

in accuracy on several batches. This demonstrates the effectiveness of ensemble methods to automatically detect and cope with sensor drift. Finally, Figure 5 shows how the classifier weights used in the ensemble change with time.

5. CONCLUSIONS

Drift remains to be a challenging problem that sensor technology has to cope with. The most common solution consists of using pre-processing techniques, including univariate and multivariate linear compensation methods, which basically rely on finding linear transformations that normalize the sensor responses across time, and/or selecting the optimal combination of features that are less prone to drift (i.e., feature selection methods). These methods often require frequent re-calibrations and updates and, in some cases, make assumptions of linear and monotonic distributions of sensor drift across time (i.e., they assume that drift is along a specific direction rather than a random one). The classifier ensemble used in this paper is effective in detecting and coping with sensor drift by using an ensemble of classifiers that track changing conditions. The ensemble method uses a linear combinations of classifiers, in which every base classifier is weighted according to its performance, to obtain better predictive performance as a whole than that could be obtained from any of the individual models acting alone.

The number of measurements chosen from the 36 months to create the 10 functional training *batches* (cf. Table 4) to be trained by the classifier ensemble should by no means be viewed as a unique combination in framing the time-evolved sensor array information. Although it is arguable how many months should be considered in each batch, the number of measurements in these functional training batches should be as uniformly distributed as possible, so that every classifier model can be trained with sufficient number of samples from each class and month. As mentioned before, it is possible to avoid batch learning algorithms that require sufficient number of samples for training and instead design concept drift algorithms that train classifiers in a *purely online* fashion [13, 14]. These algorithms create a new classifier whenever the overall prediction of the ensemble on a newly arrived training sample turns out to be incorrect and there are pruning mechanisms to control the number of classifiers in the ensemble. However, note that in these approaches the base classifiers in the ensemble have to be trained in an online fashion. In our application, we wanted to be in a position to use powerful non-linear classifiers in the ensemble and therefore SVMs trained in batch mode were an obvious choice. If we were to use online algorithms for coping with concept drift, then we would have to train non-linear SVMs in an online setting which, although not impossible, is a non-trivial problem [8]. Exploring the use of online non-linear SVMs as base classifiers in the ensemble is left for future

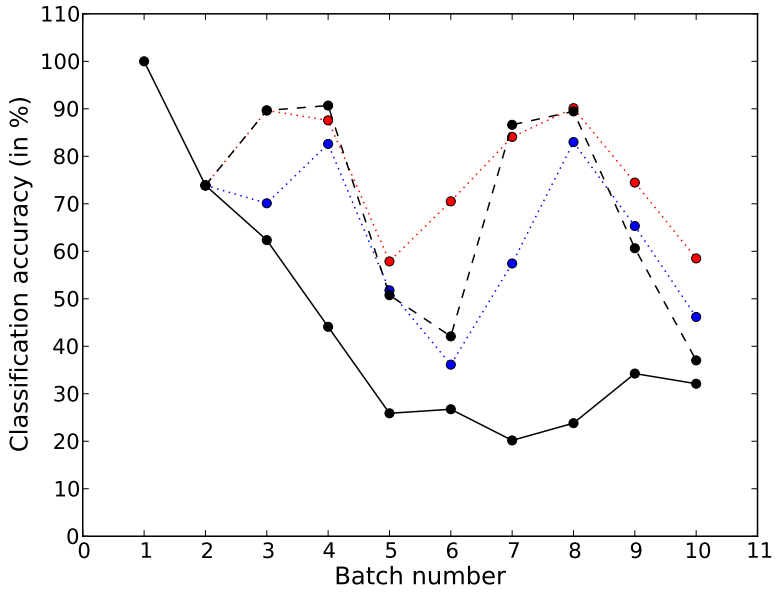

Figure 4: Performance of classifiers under Setting 1 (black, dashed), Setting 2 (red, dotted) and Setting 3 (blue, dotted) described in the text. The black, continuous line corresponds to the setting where a classifier was trained with batch 1 and tested on successive batches.

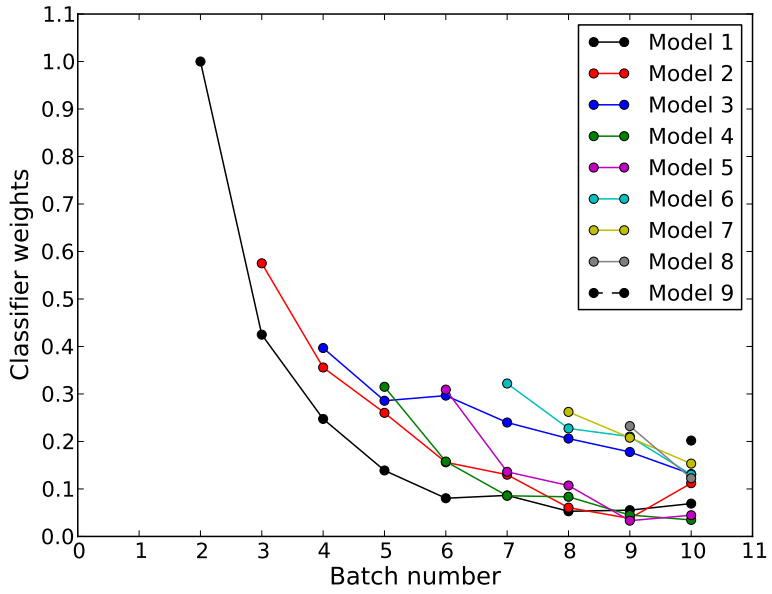

Figure 5: Classifier weights used in the ensembles (model 1 through model 9). At every point on the x-axis (time/batch) the corresponding points in the y-axis are the weights of the individual classifiers used in the ensemble. Notice that these weights sum up to 1 at every point of time/batch (x-axis) and that the contribution of an individual classifier to the ensemble gradually decreases with time.

work.

An important observation from our sensor drift analysis is that the compensations made by the proposed method remain valid for long periods of time. The 6-month gap between the recordings from the first 30 months and the last recorded month did not invalidate the compensation of drift on the same task, despite the fact that the sensors were exposed to poisoning as they were powered-off during this period of time. Such a generalization capability of our method with time makes it feasible to leverage the proposed

drift-correction method even further.

Finally, the level of generality that the proposed drift counteraction method attains is high from the following perspective. By design, the drift counteraction method using classifier ensembles operates on samples collected on-the-fly in an online fashion which is important for real-time operations. The technique does not make any assumptions about the nature of drift and is also agnostic to the classifier used in the gas identification task. As a consequence, this solution is effective for many other applications where data is collected over an extended period of time with drifting underlying distributions, and it can be readily translated to more realistic gas sensing applications, including but not limited to identification and localization in the wind tunnel, gas distribution mapping, and gas plume tracking using robots.

6. ACKNOWLEDGMENTS

This work has been supported by ONR N00014-07-1-0741 and Jet Propulsion Lab contract 1396686. We thank Travis Wong for reading drafts of this manuscript and suggesting improvements, and the anonymous reviewers for their comments.

7. REFERENCES

[1] Airgas, Inc. `http://www.airgas.com/`.

[2] Bronkhorst High-Tech B.V. `http://www.bronkhorst.com/`.

[3] Figaro USA, Inc. `http://www.figarosensor.com/`.

[4] The LabVIEW Environment. `http://www.ni.com/labview/`.

[5] T. Artusson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, and M. Holmberg. Drift correction for gas sensors using multivariate methods. *Journal of Chemometrics*, 14(5-6):711–723, 2000.

[6] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] O. Dekel, S. Shalev-Shwartz, and Y. Singer. The forgetron: A kernel-based perceptron on a budget. *SIAM Journal of Computing*, 37(5):1342–1372, 2008.

[9] W. Gopel and K.-D. Schierbaum. *Chemical and Biochemical Sensors, Part I*, chapter Definitions and typical example, pages 1–28. VCH: Weinhheim, 1992.

[10] R. Gutierrez-Osuna. Drift reduction for metal-oxide sensor arrays using canonical correlation regression and partial least squares. In *Proceedings of the 7th International Symposium On Olfaction & Electronic Nose*, 2000.

[11] A. Hierlemann and R. Gutierrez-Osuna. Higher-order chemical sensing. *ACS Chemical Reviews*, 108:563–613, 2008.

[12] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. In *Proceedings of the International IEEE Conference on Data Mining*, 2003.

[13] J. Z. Kolter and M. A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, 2005.

[14] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, 8(Dec):2755–2790, 2007.

[15] E. Llobet, J. Brezmes, X. Vilanova, J. E. Sueiras, and X. Correig. Qualitative and quantitative analysis of volatile organic compounds using transient and steady-state responses of a thick-film tin oxide gas sensor array. *Sensors and Actuators B: Chemical*, 41(1-3):13–21, 1997.

[16] M. K. Muezzinoglu, A. Vergara, R. Huerta, N. Rulkov, M. I. Rabinovich, A. Selverston, and H. D. Abarbanel. Acceleration of chemo-sensory information processing using transient features. *Sensors and Actuators B: Chemical*, 137(2):507–512, 2009.

[17] A. Pardo, S. Marco, and J. Samitier. Nonlinear inverse dynamic models of gas sensing systems based on chemical sensor arrays for quantitative measurements. *IEEE Transactions on Instrumentation and Measurement*, 47(3):644–651, June 1998.

[18] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, December 2004.

[19] A.-C. Romain, P. André, and J. Nicolas. Three years experiment with the same tin oxide sensor arrays for the identification of malodorous sources in the environment. *Sensors and Actuators B: Chemical*, 84(2-3):271–277, 2002.

[20] A.-C. Romain and J. Nicolas. Long term stability of metal oxide-based gas sensors for e-nose environmental applications: An overview. *Sensors and Actuators B: Chemical*, 146(2):502–506, 2010.

[21] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[22] A. Vergara, M. K. Muezzinoglu, N. Rulkov, and R. Huerta. Information-theoretic optimization of chemical sensors. *Sensors and Actuators B: Chemical*, 148(1):298–306, 2010.

[23] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[24] A. Ziyatdinov, S. Marco, , A. Chaudry, K. Persaud, P. Caminal, and A. Perera. Drift compensation of gas sensor array data by common principal component analysis. *Sensors and Actuators B: Chemical*, 146(2):460–465, 2010.

[25] M. Zuppa, C. Distante, P. Siciliano, and K. C. Persaud. Drift counteraction with multiple self-organising maps for an electronic nose. *Sensors and Actuators B: Chemical*, 98(2-3):305–317, 2004.