A calibrant-free drift compensation method for gas sensor arrays

Pierre Maho, Cyril Herrier, Thierry Livache, Pierre Comon, Simon Barthelme

A calibrant-free drift compensation method for gas sensor arrays

Pierre Maho[a,b], Cyril Herrier[b], Thierry Livache[b], Pierre Comon[a], Simon Barthelmé[a]

[a] *CNRS, GIPSA-Lab, Univ. Grenoble Alpes, F-38000 Grenoble, France*
[b] *Aryballe, 38000 Grenoble, France*

Abstract

Gas sensors lack repeatability over time. They are affected by drift, the result of changes at the sensor level and in the environment. A solution is to design software methods that compensate for the drift. Existing methods are often based on calibration samples acquired at the start of each new measurement session. However, finding a good reference compound is a difficult task and generating calibration samples is time-consuming. We propose a model-based correction method which does not require any calibration sample over time, operating 'blindly'. In this study, we focus on the drift affecting electronic noses. To this end, we built a real data set acquired over 9 months in real-life conditions. By using the proposed method, we show that the drift is partly compensated, thus increasing the reliability of the electronic nose. Besides, we also show that the algorithm can easily adapt if the target compounds are not all sampled during every session.

Keywords: drift, correction method, gas sensors, electronic nose, calibrant-free

1. Introduction

Drift is the result of several changes occurring at the sensor level and in the environment. These changes affect the reproducibility of gas sensors over time (Holmberg et al., 1997). In other words, the response of these sensors towards a compound can change from one measurement session to another.

The causes of drift in gas phase are numerous and depend on the sensing and transduction mechanisms. Several authors have already detailed these causes and we refer the reader to these references for a complete introduction (Holmberg and Artursson, 2004; Hierlemann and Gutierrez-Osuna, 2008). To name a few examples, drift can be caused by variations in temperature and humidity between two sessions, by sensor poisoning, by sensor ageing, by memory effects, etc... It is interesting to note that some components (*e.g.* temperature, humidity, memory effects, ...) of the drift can be avoided when the experiments are carried out in a controlled environment (see for instance the study of Vergara et al. (2012)). However, in the field, all these sources can co-occur, so their effects may not be separable. Consequently, all these causes are often treated as a single process.

In this article, we are interested in a particular type of instruments based on gas sensor arrays, namely electronic noses (eNoses). The technology is broadly inspired by mammalian olfaction (Persaud and Dodd, 1982) and is mainly based on a cross-sensitive chemical sensor array (Albert et al., 2000). An eNose aims at detecting, identifying and quantifying a broad variety of volatile compounds. In practice, this instrument is used in combination with pattern recognition algorithms for the identification part. Concretely, any application often starts by carrying out several measurements with the targeted compounds. We call this first measurement session 'Session 1'. Each gas sensor generates a time series reflecting the interaction between the sensor and the compound. From each time series, we generally extract one or two features. One measurement leads then to a vector (by considering all the gas sensors) which is assumed to be characteristic of the compound and which is called a 'signature'. After Session 1, the database of signatures is used to train a classifier which will then predict the identity of future samples.

eNoses are a good example to illustrate the issue of drift. Despite abundant proofs of concept in academic laboratories, eNoses are not a widespread technology used in everyday life conditions yet (Marco, 2014; Boeker, 2014). According to some authors in the community, drift can be considered as one of the main factors explaining the small number of practical applications of these instruments (Padilla et al., 2010). In the presence of drift between Session 1 and another session, say 'Session 2', the signatures of a given compound will be different. A direct consequence is that the classifier will not generalize from Session 1 to Session 2. In other words, it will confuse one compound for another, rendering the instrument useless.

A 'naive' solution would be to briefly retrain the classifier before using it for Session 2. Obviously, this solution is not desirable at all since each experiment is time-consuming. Therefore, the solution adopted in the literature is generally to design methods that compensate for the drift. All these methods assume what we call the Fundamental Hypothesis: that all the individual drifts of each targeted compound must be related in some way (in the

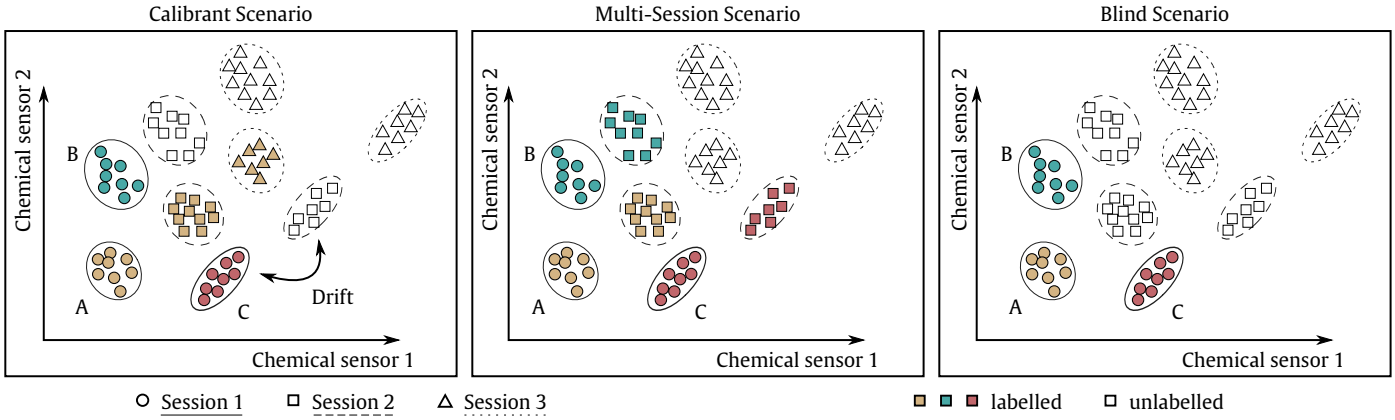


Figure 1: Simplified representation of the different scenarios that we proposed to classify the different state-of-the-art methods. Symbols and class contours stand for the session index. Colored points correspond to labelled samples while white points correspond to unlabelled samples. Left figure: calibration samples are measured during each session (Calibrant Scenario), here A is the calibrant. Middle figure: the training set includes several measurement sessions over time, and we assume that it contains drift information (Multi-Session Scenario). Right figure: the training set includes only Session 1 and there is no other labelled samples (Blind Scenario).

simplest case, equal). In other words, if each compound drifts along a random direction that is different for each targeted compound, then there is nothing one can do and all methods will fail.

Over the past two decades, many different correction methods have been put forward in the literature. These methods differ on the amount of information that is assumed to be available.

A classical approach is to assume the existence of calibration samples of one or several reference compounds during each measurement session. The drift of the reference compounds must be related in some way to the drift of the targeted compounds (Haugen et al., 2000). The calibration samples are then used to estimate the drift directions. We call this scenario: **Calibrant Scenario** (CS). In this article, we discuss 'Principal Component Analysis - Component Correction' (PCA-CC) from Artursson et al. (2000), which is one of the most popular methods in the field (Marco and Gutierrez-Galvez, 2012).

However, acquiring new samples before each new measurement session is time-consuming. In addition, finding reference compounds is not an easy task and requires a lot of upstream work, which must be repeated anytime an application with new target compounds is designed. Consequently, two other approaches have been considered in the field to relax the assumption that calibration samples are available.

The first one assumes that the drift has been captured by training data. In a classification task, a training set is always required and can be acquired during a single measurement session. Here, one assumes that the training set has been acquired across multiple sessions over time. If a long enough time has elapsed between sessions, then the training set must contain drift information. So this group of methods extracts the drift directions directly from the training set and assumes that these directions will no longer evolve for subsequent sessions. We call this scenario: **Multi-Session Scenario** (MSS). Hereafter, we discuss 'Orthogonal Signal Correction' (OSC) from Padilla et al. (2010) and 'Common Principal Component Analysis - Component Correction' (CPCA-CC) from Ziyatdinov et al. (2010).

The second class of methods assume that the only labelled data available comes from the training set, specifically a single measurement session (in which drift is not present). In fact, MSS can be attractive since it does not require any reference compound but it requires a long experimental time. For instance, the results of OSC (Padilla et al., 2010) were obtained with data acquired over 15 days (the number of measurement sessions is not detailed). For the results of CPCA-CC, the authors used at least one month of recordings (Ziyatdinov et al., 2010). This long experimental time has some limitations, so that other authors try another more appealing approach: they simply assume that there is no labelled data for estimating the drift. So the directions of the drift between two different sessions must be estimated 'blindly'. We call this scenario: **Blind Scenario** (BS). The methods dealing with the BS often rely on a preliminary classification step. The classification step identifies the samples and helps therefore to estimate the drift directions. Further in this article, we discuss the method of Di Carlo and Falasconi (2012).

We refer the reader to Marco and Gutierrez-Galvez (2012) or Rudnitskaya (2018) for a more exhaustive review of existing methods.

The three scenarios clearly differ in difficulty, since we do not have the same level of information about the data from one scenario to another. By increasing order of difficulty and by decreasing order of experimental cost, we can rank the scenarios as: Calibrant, Multi-Session and Blind. A simplified representation of the three scenarios is reported in Figure 1. The Blind Scenario is the least

restrictive in practice but is also the least explored in the literature.

In this article, we propose a new correction method for the Blind Scenario based on a drift model: each class drifts along a common direction weighted by a coefficient depending on the class. This method is called 'Expectation-Maximisation Component Correction' (EMC^2). In the next section, we develop the theory and the proposed algorithm. We also present a way to initialise EMC^2. The method is first tested on a simulated data set and then on a real data set that we built. This data set has been generated over 9 months, through 15 different sessions and shows a clear drift. The ultimate goal of this paper is to be able to correctly classify sessions 2 to 15 using only Session 1 for the training of the classifier (and without knowing any labels in sessions 2 to 15).

2. Theory

We assume that we have acquired N_1 samples (in total) of R different compounds during Session 1, all labelled. During Session 2, $Q \leq R$ compounds among the R target compounds are acquired and the N_2 samples are unlabelled. The instrument boards P chemical sensors.

A scalar is noted a, a vector $\boldsymbol{a}$ and a matrix $\boldsymbol{A}$.

2.1. Drift model

Let $g(\tilde{\boldsymbol{x}}|r, t, c_r(t))$ be the probability density of the signature $\tilde{\boldsymbol{x}} \in \mathbb{R}^P$ of the compound r, measured at time t, at concentration $c_r(t)$. In this article we are interested in compounds classification, so variations in concentration could cause problems for identification. If a classifier learns to differentiate A from B in a given range of concentration, it is unlikely that it will generalize out of the range since the signatures will not be the same (they depend on the concentration). The signature is often normalised to remove or reduce variations due to changes in concentration. In this article, we normalise each signature by dividing by its L_2-norm. We assume that $\tilde{\boldsymbol{x}}$ has been normalised and we note $\boldsymbol{x}$ the concentration-free version of $\tilde{\boldsymbol{x}}$. We note $f(\boldsymbol{x}|r, t)$ the probability density of the normalised signature $\boldsymbol{x}$, and we define as class r all the normalised signatures of the compound r.

Under the Fundamental Hypothesis, the drift of all compounds is assumed to be related. Here, we assume that the distribution of each compound r is simply shifted by a common vector $\boldsymbol{d}(t) \in \mathbb{R}^P$ weighted by a coefficient $\alpha_r(t) \in \mathbb{R}$ (depending on the compound). Mathematically, it means that there exists a density $f_0(.|r)$ such that $f(\boldsymbol{x}|r, t) = f_0(\boldsymbol{x} + \alpha_r(t)\boldsymbol{d}(t)|r)$. We assume that $\boldsymbol{d}(t = 0) = 0$, which means that the training set (Session 1) defines the starting position.

We can define the following drift model for a point $\boldsymbol{x}(t)$ at time t:

$$\boldsymbol{x}(t) \sim f_0(\boldsymbol{x} + \alpha_r(t)\boldsymbol{d}(t)|r) \tag{1}$$

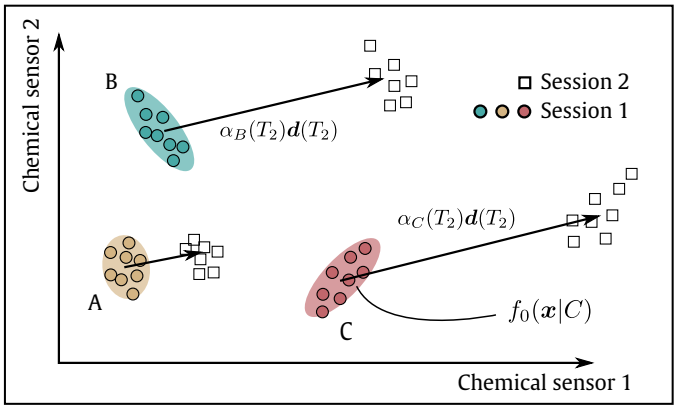


Figure 2: The drift model that we assume. Each class is translated along a common direction weighted by a coefficient depending on the class.

Finally, we assume that the drift is a slow-varying process, which is a commonly used assumption (Hierlemann and Gutierrez-Osuna, 2008). This means that for a short time window Δ_t, we have $\boldsymbol{d}(t+\Delta_t) \approx \boldsymbol{d}(t)$ and $\alpha(t+\Delta_t) \approx \alpha(t)$. So, if Session 2 starts at $t = T_2$ and lasts only for few hours, we assume that each distribution r has been shifted by $\alpha_r(T_2)\boldsymbol{d}(T_2)$. In other words, noting $\boldsymbol{x}_1$ a sample from Session 1 (training session) and $\boldsymbol{x}_2$ a sample from Session 2 (testing session):

$$\boldsymbol{x}_2 \sim f_0(\boldsymbol{x}_1 + \alpha_r(T_2)\boldsymbol{d}(T_2)|r) \tag{2}$$

A simplified representation of the drift model is reported in Figure 2.

In the case where Session 2 lasts longer than a few hours, it is possible that the slow-varying assumption will be violated. However, a straightforward solution is to divide Session 2 into smaller sessions (overlapping or not).

We can notice that the proposed model could be improved by taking into account that L_2-normalized signatures belong to the unit P-sphere. In this case, the transfer caused by the drift could be more properly modeled as a circular shift. Here, we implicitly assume that all the signatures are close enough together in the high dimension space to suppose that they belong (approximately) to an Euclidean space of dimension $P - 1$.

The goal of the next section is to estimate the R coefficients $\alpha_r(T_2)$ and the direction $\boldsymbol{d}(T_2) \in \mathbb{R}^P$, which characterize the drift between Session 1 and Session 2. This task would have been trivial if the labels $\boldsymbol{\ell}(T_2) \in \mathbb{R}^{N_2}$ of Session 2 were known, but they need also to be estimated during the compensation process (since the weight α_r depends on the compound). This estimation of the labels makes the task more challenging.

To lighten the notation, we remove the time index T_2 of Session 2. So, $\alpha_r(T_2)$ is noted α_r, $\boldsymbol{d}(T_2)$ is noted $\boldsymbol{d}$, etc...

2.2. Expectation-Maximisation Component Correction

Before going into details, we need an additional assumption: a necessary condition for identifiability. A model is said to be identifiable if the parameters of the model can be theoretically estimated uniquely. In our case, the model assumes that each class is shifted along a common direction whose amplitude depends on the class. If there is a single class during Session 2, then we can estimate R different translations moving the class in Session 2 to each class in Session 1, thus correcting the drift. However, which one of the R translations is the true one? It would be difficult to answer this question[1]. This is a matter of identifiability, and to be able to correctly estimate the parameters, we assume that we are not in this particular case: in Session 2, we have sampled at least two different (unknown) classes, so $2 \leq Q \leq R$.

Expectation-Maximisation (EM) algorithm is an iterative method for finding maximum-likelihood solutions of statistical models that have "latent variables" (Dempster et al., 1977; Bishop, 2006). Latent variables are variables that cannot be directly observed. In our case, the latent variables are the labels $\boldsymbol{\ell} \in \mathbb{R}^{N_2}$ of the data $\boldsymbol{X} \in \mathbb{R}^{N_2 \times P}$ of Session 2 and the parameters are the drift direction $\boldsymbol{d} \in \mathbb{R}^P$ and the weights $\boldsymbol{\alpha} \in \mathbb{R}^R$. The main difficulty is that we cannot estimate $\boldsymbol{\ell}$ without the knowledge of $\boldsymbol{d}$ and $\boldsymbol{\alpha}$, and we cannot estimate $\boldsymbol{d}$ and $\boldsymbol{\alpha}$ without the knowledge of $\boldsymbol{\ell}$. To solve this issue, an EM iteration proceeds in two steps: the Expectation step and the Maximisation step. The Expectation step will estimate the latent variables by considering that the parameters are known and the Maximisation step will do the reverse. These two steps are then alternated until convergence.

To derive the algorithm, we assume that each compound r is drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_r, \boldsymbol{S}_r)$, with $\boldsymbol{\mu}_r \in \mathbb{R}^P$ the class centroid and $\boldsymbol{S}_r \in \mathbb{R}^{P \times P}$ the class covariance matrix. $\boldsymbol{\mu}_r$ and $\boldsymbol{S}_r$ are estimated from the data $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times P}$ of Session 1. We define as π_r ($\boldsymbol{\pi}$ the corresponding vector of size R) the probability that a sample of Session 2 belongs to the class r.

During the Expectation step, $\boldsymbol{d}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$ have been already estimated so we can estimate the *a posteriori* probability that a sample $\boldsymbol{x}_n \in \mathbb{R}^P$ of Session 2 belongs to class r:

$$p(\ell_n = r|\boldsymbol{x}_n, \boldsymbol{d}, \boldsymbol{\alpha}, \boldsymbol{\pi}) = \frac{\pi_r \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_r + \alpha_r \boldsymbol{d}, \boldsymbol{S}_r)}{\sum_{k=1}^{R} \pi_k \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k + \alpha_k \boldsymbol{d}, \boldsymbol{S}_k)} \quad (3)$$

Equation (3) is classical in EM algorithms for mixture estimation and the reader is referred to Bishop (2006) for details.

During the Maximisation step, we estimate each parameter by maximising the likelihood. The likelihood is the probability of observing $\boldsymbol{X}$ given the parameters $\boldsymbol{d}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. For the estimation, we simply write the derivatives with respect to each parameter and set them to zero. The derivations are given in the Appendix. There is an obvious indetermination in our model: a scale ambiguity between $\boldsymbol{\alpha}$ and $\boldsymbol{d}$. To remove this ambiguity, we simply normalise $\boldsymbol{d}$ after its estimation.

After convergence (typically, less than 20 iterations), we have estimated the parameters $\boldsymbol{d}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\pi}$. The drift of each class r is assumed to be along $\boldsymbol{d}$ but the amplitude α_r of this shift depends on the class. Consequently, we need to estimate the label ℓ_n of the sample $\boldsymbol{x}_n$ to correct its drift. To that end, we take the class which maximises the *a posteriori* probability:

$$\hat{\ell}_n = \arg\max_r p(\ell_n = r|\boldsymbol{x}_n, \hat{\boldsymbol{d}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\pi}}) \quad (4)$$

Once we have estimated the label ℓ_n of the sample $\boldsymbol{x}_n$, we

Algorithm 1: Expectation-Maximisation Component Correction (EMC^2)

Data: $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times P}, \boldsymbol{\ell}_1 \in \mathbb{R}^{N_1}, \boldsymbol{X} \in \mathbb{R}^{N_2 \times P}$

Initialise $\hat{\boldsymbol{d}}^{(0)}, \hat{\boldsymbol{\alpha}}^{(0)}, \hat{\boldsymbol{\pi}}^{(0)}$

while $\hat{\boldsymbol{d}}^{(i)}$ *unchanged* **do**

`// Expectation step`

$\forall n \in [\![1, N_2]\!], \forall r \in [\![1, R]\!]$,

$$\hat{\rho}_{nr}^{(i+1)} = \frac{\hat{\pi}_r^{(i)} \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_r + \hat{\alpha}_r^{(i)} \hat{\boldsymbol{d}}^{(i)}, \boldsymbol{S}_r)}{\sum_k \hat{\pi}_k^{(i)} \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k + \hat{\alpha}_k^{(i)} \hat{\boldsymbol{d}}^{(i)}, \boldsymbol{S}_k)}$$

`// Maximisation step`

$$\forall r \in [\![1, R]\!], \quad \hat{\pi}_r^{(i+1)} = \frac{\sum_n \hat{\rho}_{nr}^{(i+1)}}{N_2}$$

$$\hat{\boldsymbol{d}}^{(i+1)} = \Lambda^{-1} \sum_{n,k} \hat{\rho}_{nk}^{(i+1)} \hat{\alpha}_k^{(i)} \boldsymbol{S}_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)$$

$$\text{with } \Lambda = \sum_k (\hat{\alpha}_k^{(i)})^2 \boldsymbol{S}_k^{-1} \sum_n \hat{\rho}_{nk}^{(i+1)}$$

$$\hat{\boldsymbol{d}}^{(i+1)} = \frac{\hat{\boldsymbol{d}}^{(i+1)}}{\|\hat{\boldsymbol{d}}^{(i+1)}\|_2}$$

$\forall r \in [\![1, R]\!]$,

if $\hat{\pi}_r^{(i+1)} \geq \frac{1}{N_2}$ **then**

$$\hat{\alpha}_r^{(i+1)} = \frac{\sum_n \hat{\rho}_{nr}^{(i+1)} \hat{\boldsymbol{d}}^{(i+1)^T} \boldsymbol{S}_r^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_r)}{\hat{\boldsymbol{d}}^{(i+1)^T} \boldsymbol{S}_r^{-1} \hat{\boldsymbol{d}}^{(i+1)} \sum_n \hat{\rho}_{nr}^{(i+1)}}$$

else

end

end

`// Drift correction`

$\forall n \in [\![1, N_2]\!]$, $\hat{\ell}_n = \arg\max_r p(\ell_n = r|\boldsymbol{x}_n, \hat{\boldsymbol{d}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\pi}})$

$\forall n \in [\![1, N_2]\!]$, $\boldsymbol{x}_{n,\text{cor}} = \boldsymbol{x}_n - \hat{\alpha}_{\hat{\ell}_n} \hat{\boldsymbol{d}}$

$\rho_{nr} = p(\ell_n = r|\boldsymbol{x}_n, \boldsymbol{d}, \boldsymbol{\alpha}, \boldsymbol{\pi})$

[1] We could use the second order moments assuming that they are different for each class, but this could be easily violated in practice.

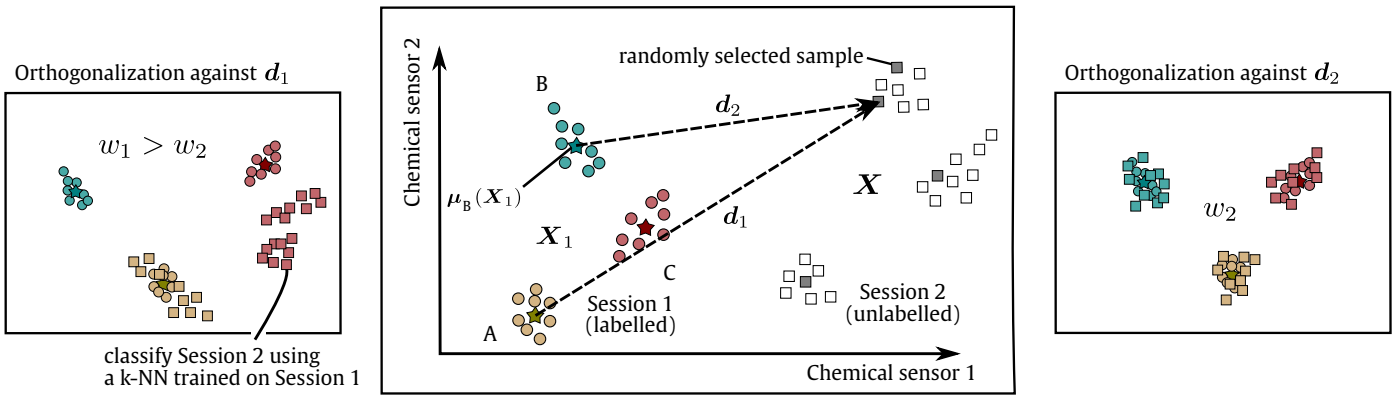


Figure 3: Graphical representation of the initialisation procedure for EMC^2 ($A = 4, R = 3$). We first select random directions from the centroids of $\boldsymbol{X}_1$ to some random samples of $\boldsymbol{X}$. Then, for each candidate direction, we orthogonalise $\boldsymbol{X}_1$ and $\boldsymbol{X}$ against the direction. In the subspace, we classify $\boldsymbol{X}_\perp$ with a k-NN trained on $\boldsymbol{X}_{1,\perp}$. Finally we compute a score w as the trace of the within-class covariance matrix (Eq. (6)) in the subspace taking both sessions. Bad directions are assumed to have a greater value w. Here, $\boldsymbol{d}_2$ is a better initial direction than $\boldsymbol{d}_1$.

can correct it:

$$\boldsymbol{x}_{n,\text{cor}} = \boldsymbol{x}_n - \hat{\alpha}_{\hat{\ell}_n} \hat{\boldsymbol{d}} \tag{5}$$

It is interesting to note that our EM-based algorithm removes one component from $\boldsymbol{x}_n$ which is $\hat{\alpha}_{\hat{\ell}_n} \hat{\boldsymbol{d}}$. It is reminiscent of some methods from the literature called Component Correction (*e.g.* PCA-CC from Artursson et al. (2000)). Therefore we call this method: Expectation-Maximisation Component Correction, abbreviated EMC^2. The algorithm is reported in Algorithm 1.

2.3. Initialisation

The results of EM algorithms depend strongly on the starting point (McLachlan and Krishnan, 2007). Poor initial parameters $\hat{\boldsymbol{d}}^{(0)}, \hat{\boldsymbol{\alpha}}^{(0)}, \hat{\boldsymbol{\pi}}^{(0)}$ will likely lead to a bad local maximum and as a result to a poor drift correction. Therefore, we propose a heuristic to correctly initialise EMC^2.

The main idea is that a simple orthogonalisation against $\boldsymbol{d}$ will superimpose $\boldsymbol{X}_1$ and $\boldsymbol{X}$ and information about $\boldsymbol{d}$ must be contained in some pairwise directions between some samples in $\boldsymbol{X}_1$ (here, we use the centroids $\boldsymbol{\mu}_r(\boldsymbol{X}_1)$ since we know the labels $\boldsymbol{\ell}_1$) and some samples in $\boldsymbol{X}$.

We start by selecting random directions going from each centroid $\boldsymbol{\mu}_r(\boldsymbol{X}_1)$ of $\boldsymbol{X}_1$ to A random samples of $\boldsymbol{X}$. For each random direction, we project $\boldsymbol{X}$ and $\boldsymbol{X}_1$ in the subspace orthogonal to the candidate direction. These versions of $\boldsymbol{X}$ and $\boldsymbol{X}_1$ are called $\boldsymbol{X}_\perp$ and $\boldsymbol{X}_{1,\perp}$. A good direction would then superimpose $\boldsymbol{X}_\perp$ and $\boldsymbol{X}_{1,\perp}$, meaning that we would have only R different clusters. A bad direction would likely create additional clusters in the subspace. To check if $\boldsymbol{X}_\perp$ and $\boldsymbol{X}_{1,\perp}$ are superimposed or not, a good figure of merit is the within-class covariance matrix. Note $\check{\boldsymbol{X}}_\perp$ the combination of the two datasets $\boldsymbol{X}_\perp$ and $\boldsymbol{X}_{1,\perp}$:

$$\boldsymbol{S}_w = \sum_{r=1}^{R} \sum_{n \in r} (\check{\boldsymbol{x}}_{n,\perp} - \boldsymbol{\mu}_r(\check{\boldsymbol{X}}_\perp))(\check{\boldsymbol{x}}_{n,\perp} - \boldsymbol{\mu}_r(\check{\boldsymbol{X}}_\perp))^{\mathrm{T}} \tag{6}$$

Algorithm 2: Heuristic for initialising EMC^2

Data: $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times P}, \boldsymbol{\ell}_1 \in \mathbb{R}^{N_1}, \boldsymbol{X} \in \mathbb{R}^{N_2 \times P}$
Input: $A \in \mathbb{N}, k \in \mathbb{N}$
Compute the centroids $\boldsymbol{\mu}_r(\boldsymbol{X}_1)$ from $\boldsymbol{X}_1$
Select randomly A samples $\boldsymbol{X}_A \in \mathbb{R}^{A \times P}$ in $\boldsymbol{X}$
Compute all the $B = A \times R$ possible directions with unit-norm between each sample in $\boldsymbol{X}_A$ and each centroid $\boldsymbol{\mu}_r(\boldsymbol{X}_1)$
for i *in* $1 : B$ **do**
  $\boldsymbol{X}_{1,\perp} = \boldsymbol{X}_1(\boldsymbol{I}_P - \boldsymbol{d}_i \boldsymbol{d}_i^{\mathrm{T}})$
  $\boldsymbol{X}_\perp = \boldsymbol{X}(\boldsymbol{I}_P - \boldsymbol{d}_i \boldsymbol{d}_i^{\mathrm{T}})$
  $\boldsymbol{\ell}_i = k\text{-NN}(\boldsymbol{X}_{1,\perp}, \boldsymbol{\ell}_1, \boldsymbol{X}_\perp, k)$
  Merge $\boldsymbol{X}_{1,\perp}$ and $\boldsymbol{X}_\perp$ as $\check{\boldsymbol{X}}_\perp$, $\boldsymbol{\ell}_1$ and $\boldsymbol{\ell}_i$ as $\check{\boldsymbol{\ell}}$
  Compute the score w_i with $(\check{\boldsymbol{X}}_\perp, \check{\boldsymbol{\ell}})$ as the trace of Eq. (6)
end
$I = \arg\min_i w_i$
$\boldsymbol{\ell} = \boldsymbol{\ell}_I, \quad \boldsymbol{d}^{(0)} = \boldsymbol{d}_I,$
$\alpha_r^{(0)} = (\boldsymbol{\mu}_r(\boldsymbol{X}) - \boldsymbol{\mu}_r(\boldsymbol{X}_1))^{\mathrm{T}} \boldsymbol{d}^{(0)}, \quad \pi_r^{(0)} = \frac{\#\{\boldsymbol{\ell}_I = r\}}{N}$

with $\boldsymbol{\mu}_r(\check{\boldsymbol{X}}_\perp)$ the centroid of the class r in the orthogonal subspace, computed from $\check{\boldsymbol{X}}_\perp$.

$\boldsymbol{S}_w$ evaluates how far from each other the samples belonging to a same class are. The greater the within-class covariance is, the more spread-out the classes are. To have only a single number, we take $w = \mathrm{Tr}(\boldsymbol{S}_w)$. The lower w is, the better $\boldsymbol{X}_\perp$ and $\boldsymbol{X}_{1,\perp}$ are superimposed.

However, the within-class covariance matrix needs labels and we do not have any label for $\boldsymbol{X}_\perp$ (the Session 2 projected in the orthogonal subspace). Therefore, the final step is a classification step. For that, we simply apply a k-NN in the orthogonal subspace of the candidate direction to estimate the labels $\boldsymbol{\ell}$ of $\boldsymbol{X}_\perp$ based on $(\boldsymbol{X}_{1,\perp}, \boldsymbol{\ell}_1)$. The initialisation procedure is described in Algorithm 2 and a graphical representation is reported in Figure 3.

As a remark, we assume here that there actually is a significant drift in the data. If there is no drift, applying a drift correction may lead to worse results (this is true for every method of drift correction). In our case, the initialization will lead to a random direction and this random direction can unfortunately be correlated with a discriminative direction. The risk is then that two clusters may be merged together after correction. To overcome this issue, a drift detection method needs to be developed to decide to apply or not a drift correction method.

2.4. Code availability

`R` code for Algorithms 1 and 2 as well as the other methods used in this paper are available at: `https://github.com/mahopie/eNoseDrift.git`. Simulation code is also provided.

3. Data sets

3.1. Artificial data set

For the generation of the artificial data set, we reuse and complicate the simulation settings proposed by Di Carlo et al. (2011).

A simulation starts with the generation of a training set $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times P}$ with no drift and with $R = 6$ compounds, $N_1 = 60$ and $P = 4$. Each class is Gaussian with parameters $\boldsymbol{\mu}_r \sim \mathcal{N}(\boldsymbol{0}_P, \frac{\beta^2}{2P}\boldsymbol{I}_P)$ and $\boldsymbol{S}_r = \boldsymbol{I}_P$. Then, we generate data with drift according to the model in Equation (1) and only $Q \leq R$ compounds are simulated (for the same training set, we test from $Q = R$ to $Q = 2$).

Our model only implies that the drift is additive, slow-varying and along a direction $\boldsymbol{d}(t)$ independent of the compound (only the weight factor is). So, $\boldsymbol{d}(t)$ is not constrained to be constant or linear over time, and can be rather complex. To introduce a non-linearity in the drift over time, we simulate the direction $\boldsymbol{d}(t)$ with an arbitrary non-linear function: $\boldsymbol{d}(t) = \frac{t}{h}\boldsymbol{d}_0 + \frac{t}{h}\sin(\frac{t}{h}\boldsymbol{d}_0)$, sin is applied component-wise, $\boldsymbol{d}_0 \sim \mathcal{U}(0, 1)$ is a random direction (normalised) and h is a hyperparameter ($h = 30$). The weights $\alpha_r \sim \mathcal{U}(0.5, 3)$ are also random and the class probabilities π_r are set to $\frac{1}{Q}$. So, $\forall r \in [\![1, Q]\!]$, $\forall t \in [\![0, t_{\max}]\!]$:

$$\boldsymbol{x}_r(t) \sim \mathcal{N}(\boldsymbol{\mu}_r + \alpha_r \frac{t}{h}(\boldsymbol{d}_0 + \sin(\frac{t}{h}\boldsymbol{d}_0)), \boldsymbol{I}_P) \qquad (7)$$

$t_{\max}$ is the duration of the experiment and is set to 251.

The simulations are random so the generated classes could be hardly separable. In this case, we may not evaluate only the drift correction but also the initial discrimination between classes. To prevent this issue from occuring, we discard simulations for which any pairwise distance between centroids is lower than $\frac{\beta}{2}$ (Di Carlo et al., 2011). The simulations (starting from the generation of Session 1) are repeated 100 times.

3.2. Real data set

A part of the data set presented in this paper has been already introduced in Maho et al. (2019).

Electronic nose. The optoelectronic nose used in this study is the commercial version of the one described in Brenet et al. (2018). The instrument is provided by the company Aryballe. More details about the working principle of the instrument are given in Brenet et al. (2018) or in Maho et al. (2020b). In this article, the instrument uses 26 different chemical sensors which are replicated 2 or 3 times on the surface, leading to an array of 63 gas sensors. This electronic nose is rather new compared to other existing eNoses (e.g. metal-oxide based sensors) but it has already shown noteworthy results in laboratory conditions for the discrimination of compounds with similar molecular structures (down to only one carbon atom different in Brenet et al. (2018) and even enantiomers in Maho et al. (2020b)).

Experimental setup. We built a robotic setup to study drift in realistic conditions. Three VOC sources of pure compounds (Geranyl acetate, Citral and β-Pinene) are disseminated in an open environment. This environment is scanned by a robot which carries the eNose. By following a predefined path, the robot goes over the VOC sources and the eNose detects them. The path is repeated multiple times to obtain a substantial data set at each measurement session. The setup and the response (after the baseline subtraction) of the instrument after one lap are reported in Figure 4. Temperature and humidity are not controlled during the experiments (temperature variations belong to [20.4°C, 34.5°C] and humidity variations to [14.2%, 40.3%]) and compounds concentration is neither controlled, nor measured.

To have a better idea of the experiments, we encourage the reader to watch the video in footnote[2]. In this video, a larger setup is used but the idea remains the same.

Data set. The responses of the gas sensors look like peaks due to the very short time injection ($\sim$ 1 sec). Each peak corresponds to one experiment with one of the 3 compounds. A segmentation procedure is applied to extract these peaks and is detailed in another paper (see Maho et al. (2020a)). Note that the resulting time series are uncommonly short, compared to the standard for e-Noses, where measurements typically take a few minutes. Each peak is integrated for each chemical sensor. Therefore, each peak gives a signature $\tilde{\boldsymbol{x}}$. The replicas are averaged to reduce dimensionality, so $P = 26$ and $\tilde{\boldsymbol{x}} \in \mathbb{R}^{26}$. A session usually lasts for few hours so some compounds run out at the end of the experiment and injected concentration decreases. To get rid of the variations in concentration, we normalise each signature individually by dividing it by its ℓ_2-norm. The normalised signature is noted $\boldsymbol{x}$.

One lap gives one signature for each one of the $R = 3$ compounds, so after a certain number of laps we have a

[2]Link to the explanatory video: `https://drive.google.com/file/d/1AddmCYFwQcHtG1aTksn8zyC-0e7CDzbU/view`

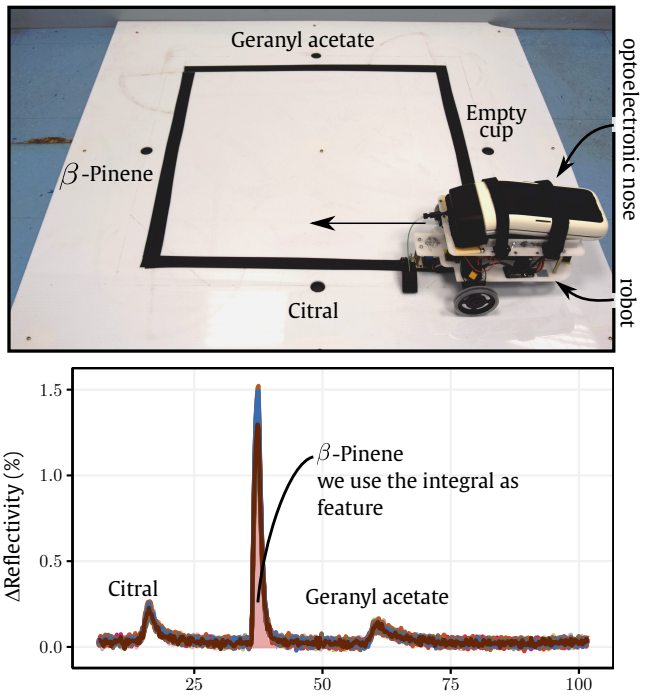


Figure 4: Top: robotic setup. Bottom: response of the instrument after one lap. Each peak corresponds to the passage of the robot over a gas source.

	Date	N_i	R	T (°C)	RH (%)	SVM (%)
1	06-07-2018	110	3	NA	NA	98 ± 1
2	14-09-2018	112	3	NA	NA	94 ± 2
3	21-09-2018	93	3	[31.6, 34.5]	[27.0, 36.0]	90 ± 1
4	04-10-2018	165	3	[25.1, 29.6]	[30.1, 39.0]	93 ± 1
5	16-10-2018	71	3	[28.0, 28.9]	[34.6, 37.5]	92 ± 3
6	09-11-2018	142	3	[23.3, 28.2]	[31.4, 37.5]	81 ± 1
7	23-11-2018	132	12	[20.4, 27.6]	[26.6, 40.0]	89 ± 1
8	04-12-2018	148	3	[23.5, 27.1]	[34.7, 40.3]	91 ± 1
9	14-12-2018	173	3	[24.4, 27.8]	[18.6, 23.7]	91 ± 1
10	20-12-2018	161	12	[26.2, 27.5]	[24.1, 29.3]	76 ± 1
11	09-01-2019	75	12	[25.1, 27.5]	[20.3, 24.7]	71 ± 3
12	11-01-2019	71	3	[25.6, 27.3]	[14.2, 16]	89 ± 2
13	06-02-2019	164	3	[24.2, 27.5]	[17.2, 21.3]	91 ± 1
14	13-02-2019	73	12	[24.4, 27.0]	[17.6, 23.3]	87 ± 1
15	21-03-2019	256	3	[24.4, 27.2]	[18.1, 22.7]	94 ± 0
	9 months	1934		[20.4, 34.5]	[14.2, 40.3]	[71, 98]

Table 1: Information about the data sets studied in this section. *NA* values are due to the absence of the sensors at the recording date. R column corresponds to the number of compounds used in the experiment, if $R = 12$ then we extract from the session the 3 target compounds. RH stands for Relative Humidity. SVM column corresponds to the cross-validated classification rate (repeated 10 times) when we train and test on the same session (mean ± standard deviation).

data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times P}$. $\boldsymbol{X}$ corresponds to only one measurement session lasting for few hours. Since we perform several measurement sessions over time, we note $\boldsymbol{X}_i \in \mathbb{R}^{N_i \times P}$ the data acquired during Session i.

In total, we have acquired 15 different sessions over 9 months, corresponding to a total of ∼2000 samples. The measurement sessions differ on several experimental parameters: the amount of liquid solution left in the gas sources, the environmental conditions (temperature, humidity), the running order, the robot speed and the number of laps. Some of these experimental parameters and the recording dates are reported in Table 1.

To add another difficulty, all the sessions have not been generated with the same setup. Most of the sessions have been generated with a setup containing only the 3 target compounds (see Figure 4). However, some of the sessions have been generated with a similar, but larger, setup containing 12 different compounds in 24 different cups, including the studied compounds[2]. For these sessions, we extracted from the data the 3 compounds of interest. In Table 1, the column R indicates the number of compounds actually present in the session.

In Table 1, we also report some classification results (SVM column). They correspond to the ability of the eNose to tell the 3 compounds apart when the training and the testing are performed in the same session: this corresponds to the 'naive' solution explained in Introduction and we call them intra-session scores. They have been obtained after a dimensionality reduction with PCA using 5 components and after a 5-fold cross validation using a linear SVM. They highlight that the performance of the eNose regarding these 3 compounds is stable over time. However, if one wants to use only Session 1 to predict the future sessions, we will see later that a drift between the sessions appeared.

The data set is available upon request.

4. Results and discussion

4.1. Artificial data set

In Figure 5a (left), we present a random realization of the performed simulations. In order to cope with the non-linearity that we introduced, we segment $\boldsymbol{X}$ as a series of non-overlapping windows of size $\Delta_t = 9$. For one simulation, it corresponds to a total of 28 time windows. Each time window is considered as a measurement session of $9 \times Q$ samples (Session i) which needs to be drift corrected. We recall that Q, the number of compounds in Session i, varies from 2 to $R = 6$.

We compare the proposed method to 2 other methods: PCA-CC from Artursson et al. (2000) working under the Calibrant Scenario and the method of Di Carlo and Falasconi (2012) working under the Blind Scenario. The methods considering the Multi-Session Scenario are not used here since the first sessions will not contain enough information about the future directions ($\boldsymbol{d}(t)$ is rather complex).

For PCA-CC, the calibration samples are accumulated over time (for Session i we use the calibration samples from Sessions $1 \rightarrow i$) and the number of components to remove is set to the minimal number of components required to explain at least 80% of the variance of $\boldsymbol{X}_{\text{cal}}$ (the matrix containing the calibration samples). For the classification, the corrected samples are classified with a linear

SVM learnt from Session 1. Compound 1 is used as the reference.

EMC^2 and Di Carlo's method are applied session-by-session. These two algorithms return automatically labels. For Di Carlo's methods we use linear SVM and Mahalanobis distance (see Di Carlo and Falasconi (2012)). We initialise EMC^2 by using Algorithm 2. In Figure 5a (right), we report the result using EMC^2 for a random realization.

In Figure 5b, we report the classification results. The column labelled "Raw" corresponds to the results with no drift correction.

EMC^2 clearly outperforms PCA-CC while the latter requires much more information (calibration samples). We notice that both methods are stable regarding the number of compounds present Q, whatever the performance. Parameter Q mostly influences the variance of the classification rate: the lower Q is, the more a misclassification is influencing the score. The performance of PCA-CC is surprisingly quite poor. We explain that by the small number of chemical sensors that we simulated ($P = 4$) in addition to the complexity of the drift. Even if Artursson et al. (2000) claimed that PCA-CC can compensate for a non-linear drift by removing more and more components, we noticed that this method tends to remove discriminative directions. In fact, due to the small number of chemical sensors simulated here, it is more likely that the Principal directions of $\boldsymbol{X}_{\text{cal}}$ (the matrix containing the calibration samples) carry information about class discrimination. To support this remark, we made other simulations with the same settings, but we increased the dimension P of the space to 6. In this case, the average score of PCA-CC reached 82 % (averaged over Q, not reported in Figure 5b).

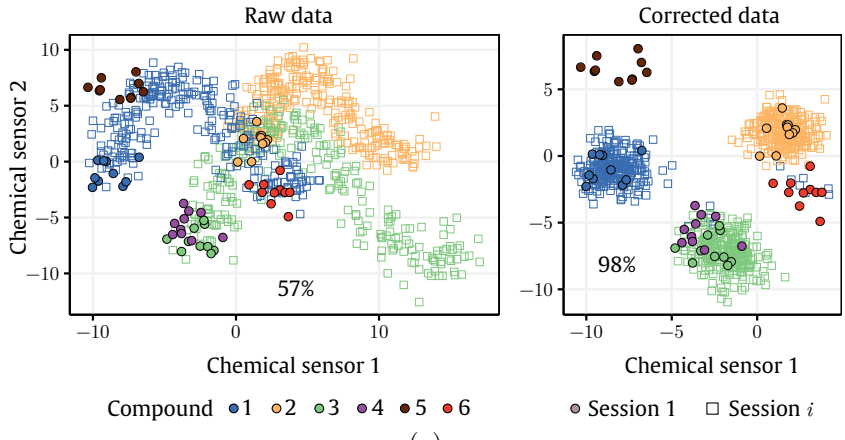
EMC^2 also outperforms the method of Di Carlo et al. It is interesting to note that the performance of the method of Di Carlo et al. is influenced by Q: the lower Q is, the better. We explain that by the fact that their method makes use of the correction factor of Session i as initialisation for Session $i + 1$. So, this method will tend to accumulate errors and errors will likely increase with Q. In fact, the classification rate tends to considerably decrease over time (starting from 100% for Session 2 to 70% for Session 28). EMC^2 does not suffer from this drawback since each session (or time window) is treated separately. So, if a session is badly corrected, it will not influence the next session.

4.2. Real data set

We recall that each signature $\boldsymbol{x} \in \mathbb{R}^{26}$ is obtained after averaging the replicas and after normalising it by its ℓ_2-norm. To reduce dimensionality, we use PCA and we project the data of each Session i onto the 5 first Principal directions of the data of Session 1 (and only Session 1). In other words, we use only the knowledge of Session 1 to reduce the number of dimensions. For PCA-CC, CPCA-CC and OSC, the correction is applied before reducing dimensionality (it leads to better results). For EMC^2 and Di Carlo's method, we first reduce dimensionality since some matrices of dimension $P \times P$ need to be estimated from the data (covariance matrices for EMC^2 and correction factors for Di Carlo's method).

Results for $Q = 3$. In Figure 6a (left), we report the projection of the entire data set onto the 2 first Principal directions of the data from Session 1. From this representation, the drift between Session 1 and the subsequent sessions is clear. If we try to predict Sessions 2 to 15 by training on Session 1 then the score reaches the chance level (33%) after Session 3 (purple line in Figure 6b).

To evaluate the correction methods, we train a linear SVM using only the data from Session 1 and we predict the corrected data from Sessions 2 to 15 (after dimensionality reduction down to 5). EMC^2 and Di Carlo's method directly return the labels ℓ_i of Session i so we use them for computing the scores. However, we can also use any

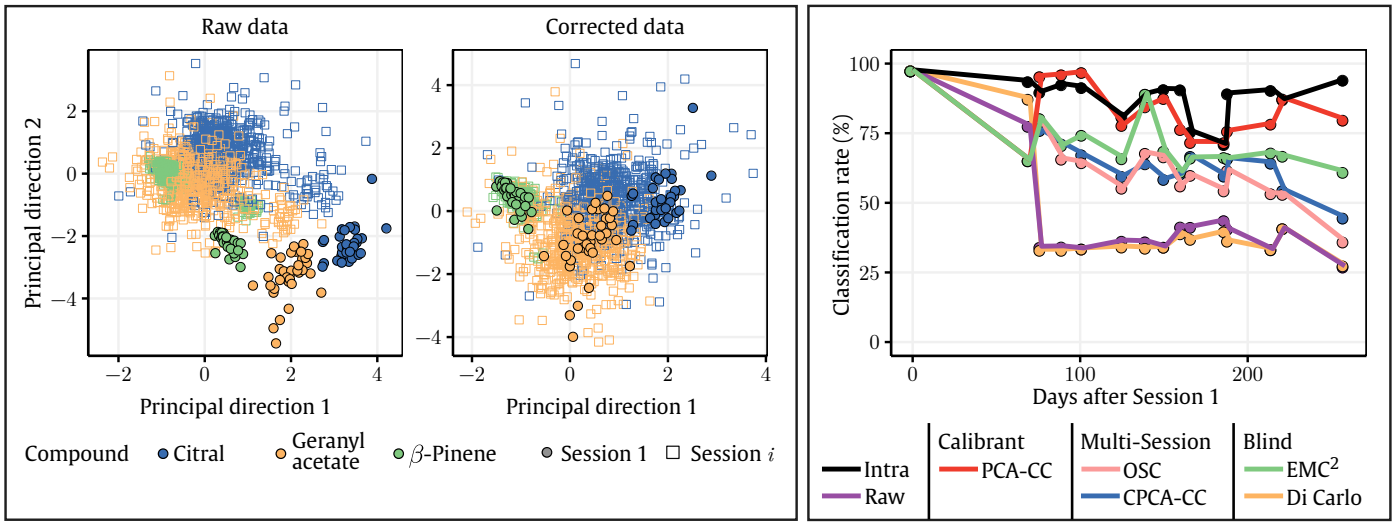


(a)

Q	Raw	PCA-CC	Di Carlo et al.	EMC^2
6	64±10	62±16	80±15	98±2
5	65±13	60±18	85±12	98±2
4	65±15	61±21	89±14	97±2
3	65±17	60±23	93±10	97±3
2	64±21	59±27	95±11	89±9

(b)

Figure 5: (a) A random realization ($Q = 3$). The indicated scores correspond to classification rates by training on Session 1 (the ten first points). Left figure corresponds to raw data (without any correction) and right figure corresponds to data corrected using EMC^2. (b) Classification results (%) ± standard deviation, over the 100 simulations. Raw column corresponds to the data with no drift correction.



(a) Projection of the data onto the 2 first Principal directions of Session 1

(b) Classification rates over time

Figure 6: (a) Left figure corresponds to the uncorrected data: circles stand for Session 1 and squares for Sessions 2 to 15. All the sessions are projected onto the 2 first Principal directions (PCA) of the data from Session 1. Right figure corresponds to the data after applying EMC^2 session-by-session. (b) Classification rates over time for all the methods considered in this paper. Intra is when we train and test on Session i. Raw method corresponds to the uncorrected data (training on Session 1 and testing on Session i).

Method	Scenario	Score (%)
PCA-CC	CS	82.1
EMC^2	BS	69.5
CPCA-CC	MSS	62.8
OSC	MSS	59.8
Raw	-	39.9
Di Carlo	BS	39
Intra-session	-	87.9

Table 2: Ranked averaged scores over sessions. Intra-session method is when we train and test on Session i (5-fold cross-validation procedure). Raw method corresponds to the uncorrected data (training on Session 1 and testing on Session i).

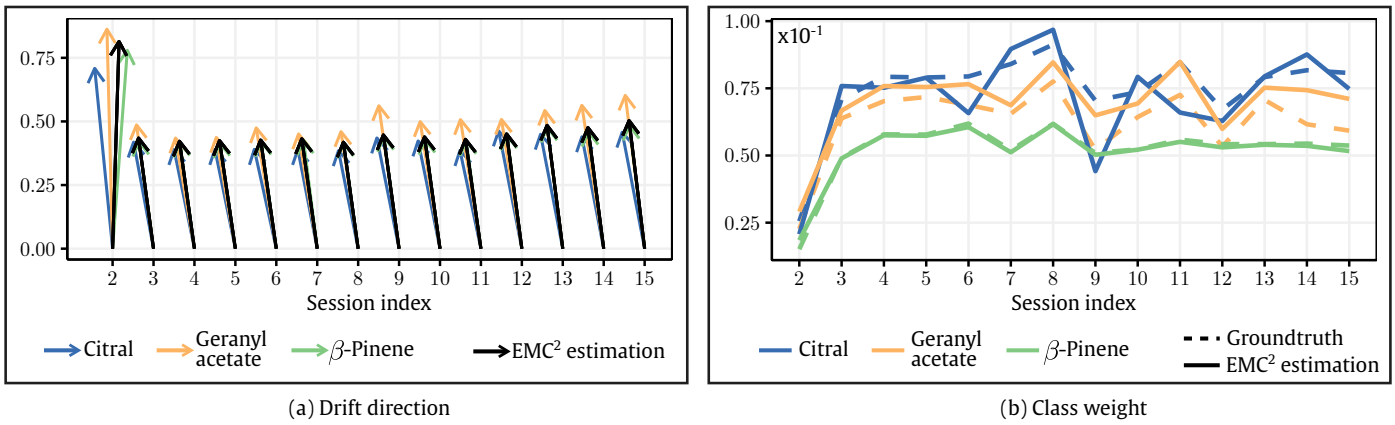
other classifier after correcting the data using EMC^2. For all methods, we remove only one component for the comparison to be fair.

For the methods working under the Multi-Session Scenario, we use only the two first sessions. In fact, this parameter can greatly influence the results and is particularly difficult to tune. In practice, we cannot know early how many sessions we will need to correctly evaluate the drift directions. So we apply here the principle: the least, the better.

In Figure 6a (right), we report the projection of corrected data using EMC^2. In Figure 6b, we report the classification results over time (or over sessions) for all the methods considered in this paper. These results (excluding Session 1) are summarized in Table 2.

Clearly, the method using calibration samples (PCA-CC) outperforms the others. We observed that the accumulation of the calibration samples over time (for processing Session i, we take all the calibration samples from Sessions 1 to i) helps the method to find better directions for the drift. If we no longer accumulate the samples but use only the calibration samples from Sessions 1 and i, then the averaged score goes down to 70% and becomes equivalent to the one obtained by our method (not reported in Table 2). The accumulation will help if the direction is stable over time but can also decrease the performance in case of a direction change (which is hard to predict in practice). In addition, the results that we reported in Table 2 for PCA-CC depend greatly on the choice of the reference compound. We chose Citral because it leads to the best results. However, in practice, we do not know the labels of the subsequent sessions so it is quite hard to tell which compound will be the best reference in the end. In fact, if we take β-Pinene as reference, then the average score goes down to 63% (not reported in Table 2). These two remarks reveal that this kind of methods depends strongly on experimental decisions which are hard to make.

Methods working under the Blind Scenario do not suffer from these decisions which influence the overall performance. However, they can be less robust. Table 2 shows that the proposed method, EMC^2, performs fairly well without requiring any labels from the subsequent sessions. It performs even better than the Multi-Session based methods. The results validate both the proposed algorithm and its heuristic for initialisation. In Figure 7a, we report the drift directions of each compound individually (colored arrows) and the common direction estimated with EMC^2 (black arrow). We can note that the assumption according to which each class drifts along a common direction seems appropriate. In Figure 7b, we report the class weights α_r (dashed: ground-truth, solid: EMC^2 estimation). We ob-



(a) Drift direction

(b) Class weight

Figure 7: Validation of the drift model. (a) Drift directions over sessions of each compound individually (colored arrows) and the common direction estimated by EMC^2. Directions are represented along the 2 first Principal directions, and from Session 1 to Session i. (b) True class weights (dashed lines) over time and estimated weights using EMC^2 (solid lines).

serve that the factors are different for each class and vary over time, as expected by the proposed model. It is noteworthy that the estimation of the weights of β-pinene by EMC^2 fits perfectly with the groundtruth. However, the algorithm performs less well for the weights of Citral and Geranyl acetate. This may be due to the overlap between the two clusters which leads to some misclassifications in our correction procedure. In fact, we observed that most of the misclassifications comes from the Citral cluster which is confused by the classifier with Geranyl acetate. These errors then result in an underestimation of the weights.

The poor performance of CPCA-CC and OSC can be attributed to the small number of training sessions that we have used. The 2 first sessions may not contain enough drift information about the subsequent sessions. In this case, performance can be increased by considering more sessions but it is again a matter of experimental decisions. This remark highlights their main drawback: in practice, nobody knows how many training sessions are required for the future.

The last method used, namely Di Carlo's method, has the same performance than the Raw method, the method which does not apply any correction. In fact, this method fails to adapt when the data shows a rather 'discontinuous' drift (in comparison to the 'continuous' impression of the simulated drift in Figure 5a left). This method would likely perform better if sessions had been closer together in time. However, the timing of the sessions is often not controlled in the field and methods have to cope with large changes.

Results for $Q = 2$. In practice, a training set with the R target compounds will be generated during Session 1. However, it is not realistic to expect that the R compounds will be measured every time. It is more realistic to expect only $Q < R$ compounds. The identity of these Q compounds is obviously unknown, which makes the task of drift correction even more tricky for the methods under the Blind Scenario.

Method	Scenario	Compound out			Mean
		Citral	Geranyl	β-Pinene	
PCA-CC	CS	81	**92**	**75**	82
EMC2	BS	**92**	57	70	73
OSC	MSS	78	59	34	57
CPCA-CC	MSS	72	59	28	53
Di Carlo	BS	56	47	47	50
Raw	-	54	55	6	39
Intra-session	-	90	99	91	93

Table 3: Average scores over sessions when one compound is removed from the subsequent sessions. Results are ranked according to the global average (over sessions and compound out).

In our data set, all the sessions measure the same number $R = 3$ of compounds. However, we can artificially remove from the subsequent sessions one compound out of the R compounds, leading to $Q = 2$ compounds. Only the first session (and the 2 first sessions for methods considering the MSS) keeps $R = 3$ compounds for the training. Drift correction methods are then applied according to the same pipelines. For methods requiring calibration samples, one compound out of the $Q = 2$ compounds is randomly selected as the calibrant for all the sessions. For selecting which compound is removed from all the testing sessions, the solution is simple: in separated experiments, each one of the $R = 3$ compounds is selected in turn.

Table 3 reports the classification results, averaged over sessions (excluding Session 1). The ranking is more or less unchanged compared to Table 2. The method considering the CS still outperforms the others. However, when $Q = 2$, 50% of the information is known by this group of methods (information from calibration samples), so the significance of their performance must be moderated. More surprisingly, the proposed method EMC^2 performs fairly well, meaning that the method is able to adapt to a class disappearance during Sessions 2 to 15. Concerning the meth-

ods CPCA-CC and OSC, their global performance is poor which is mainly due to the data set where β-Pinene is removed. In fact, most of the good classifications of these methods come from this cluster, and by removing it, we remove the good classifications.

5. Conclusion

In this paper, we proposed a correction method to compensate for the drift affecting gas sensor arrays. In the literature, most drift-correction methods to assume the Calibrant Scenario or the Multi-Session Scenario. However, we saw that their results depend on experimental decisions which are hard to make in practice: which reference compound to take? Do we keep the calibration samples of the previous sessions? How many training sessions do we need for correctly estimating the drift?

As an alternative, the methods working under the Blind Scenario are free of these considerations. By using only the labels available during the first measurement session, they are by far the least restrictive in practice. This contribution proposed a new correction technique for this group of methods which has been little explored in the literature. It is based on a drift model and has been validated on artificial and real data sets. The method can easily adapt to abrupt changes in the drift direction which could occur over time and does not require that all the classes are sampled during each new measurement session (*i.e.* some classes may be absent in some sessions).

However, drift is still an open issue and various perspectives of this work can be explored by further studies. First, our method has compensated only partly for the drift so there is still some margin for improving the results. Second, we studied here the drift correction for compounds classification. Another main task is the estimation of the concentration of the volatile compounds. It is still unclear what would give the correction for this task. Finally, the final goal by using gas sensor arrays is to use one instrument to generate a data base for training a classifier, and then generalize to other instruments deployed in the field. In this case, another drift will appear, an inter-instrument drift (see for instance the study of Fonollosa et al. (2016)). In fact, the manufacturing process of gas sensors is rarely perfectly reproducible. This means that a classifier which learns from one instrument may not directly generalize to another instrument. In this article, we focused only on the intra-instrument drift. The solution proposed may generalize to inter-instrument drift.

Appendix

By assuming independent and identically distributed samples, the likelihood is defined as:

$$\mathcal{L}(\boldsymbol{d},\boldsymbol{\alpha},\boldsymbol{\pi}|\boldsymbol{X}) = \prod_{n=1}^{N_2} p(\boldsymbol{x}_n|\boldsymbol{d},\boldsymbol{\alpha},\boldsymbol{\pi}) \tag{8}$$

We usually prefer to work with the log-likelihood $\mathcal{H}$:

$$\mathcal{H}(\boldsymbol{d},\boldsymbol{\alpha},\boldsymbol{\pi}|\boldsymbol{X}) = \sum_{n=1}^{N_2} \ln(\sum_{k=1}^{R} \pi_k \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k + \alpha_k \boldsymbol{d}, \boldsymbol{S}_k)) \tag{9}$$

We note $\mathcal{N}(k) = \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k + \alpha_k \boldsymbol{d}, \boldsymbol{S}_k)$ and $\rho_{nk} = p(\ell_n = k|\boldsymbol{x}_n, \boldsymbol{d}, \boldsymbol{\alpha}, \boldsymbol{\pi})$.

① Estimation of $\boldsymbol{d}$.

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{d}} = \sum_{n=1}^{N_2}\sum_{k=1}^{R} \underbrace{\frac{\hat{\pi}_k \mathcal{N}(k)}{\sum_{i=1}^{R} \hat{\pi}_i \mathcal{N}(i)}}_{\rho_{nk}} \times \underbrace{\frac{\partial}{\partial \boldsymbol{d}}(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k - \hat{\alpha}_k \boldsymbol{d})^{\mathrm{T}} \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k - \hat{\alpha}_k \boldsymbol{d}))}_{\hat{\alpha}_k \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) - \hat{\alpha}_k^2 \boldsymbol{S}_k^{-1}\boldsymbol{d}}$$

So, $\frac{\partial \mathcal{H}}{\partial \boldsymbol{d}} = 0$ if and only if

$$\hat{\boldsymbol{d}} = \Lambda^{-1} \sum_{n,k} \rho_{nk} \hat{\alpha}_k \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \tag{10}$$

with $\Lambda = \sum_k \hat{\alpha}_k^2 \boldsymbol{S}_k^{-1} \sum_n \rho_{nk}$

② Estimation of $\boldsymbol{\alpha}$.

$$\frac{\partial \mathcal{H}}{\partial \alpha_k} = \sum_{n=1}^{N_2} \underbrace{\frac{\hat{\pi}_k \mathcal{N}(k)}{\sum_{i=1}^{R} \hat{\pi}_i \mathcal{N}(i)}}_{\rho_{nk}} \times \underbrace{\frac{\partial}{\partial \alpha_k}(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k - \alpha_k \hat{\boldsymbol{d}})^{\mathrm{T}} \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k - \alpha_k \hat{\boldsymbol{d}}))}_{\hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) - \alpha_k \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}^{-1} \hat{\boldsymbol{d}}}$$

So, $\frac{\partial \mathcal{H}}{\partial \alpha_k} = 0$ if and only if

$$\sum_n \rho_{nk} \alpha_k \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}_k^{-1} \hat{\boldsymbol{d}} = \sum_n \rho_{nk} \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)$$

If $\sum_n \rho_{nk} \neq 0$ then,

$$\forall k \in [\![1, R]\!], \quad \hat{\alpha}_k = \frac{\sum_n \rho_{nk} \hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)}{\hat{\boldsymbol{d}}^{\mathrm{T}} \boldsymbol{S}_k^{-1} \hat{\boldsymbol{d}} \sum_n \rho_{nk}} \tag{11}$$

③ Estimation of $\boldsymbol{\pi}$. Since vector $\boldsymbol{\pi}$ contains probabilities, $\sum_{k=1}^{R} \pi_k = 1$. To incorporate this constraint in the optimisation, we build the Lagrangian $\Gamma(\boldsymbol{\pi}, \lambda)$:

$$\Gamma(\boldsymbol{\pi}, \lambda) = \mathcal{H}(\boldsymbol{d},\boldsymbol{\alpha},\boldsymbol{\pi}|\boldsymbol{X}) + \lambda(\sum_{k=1}^{R} \pi_k - 1)$$

$$\frac{\partial \Gamma}{\partial \pi_k} = \sum_{n=1}^{N_2} \frac{\mathcal{N}(k)}{\sum_{i=1}^{R} \pi_i \mathcal{N}(i)} + \lambda \tag{12}$$

So, $\frac{\partial \Gamma}{\partial \pi_k} = 0$ if and only if

$$\begin{aligned}
& \sum_{n=1}^{N_2} \frac{\mathcal{N}(k)}{\sum_{i=1}^{R} \pi_i \mathcal{N}(i)} + \lambda = 0 \\
\overset{\times \pi_k}{\iff} \quad & \sum_{n=1}^{N_2} \pi_k \frac{\mathcal{N}(k)}{\sum_{i=1}^{R} \pi_i \mathcal{N}(i)} + \pi_k \lambda = 0 \\
\overset{\text{Sum over k}}{\iff} \quad & \sum_{n=1}^{N_2} \sum_{k=1}^{R} \pi_k \frac{\mathcal{N}(k)}{\sum_{i=1}^{R} \pi_i \mathcal{N}(i)} + \sum_{k=1}^{R} \pi_k \lambda = 0 \\
\iff \quad & \lambda = -N_2
\end{aligned}$$

To conclude, inject $\lambda = -N_2$ in Equation (12) and multiply both sides by π_k:

$$\forall k \in [\![1, R]\!], \quad \hat{\pi}_k = \frac{\sum_n \rho_{nk}}{N_2} \tag{13}$$

Equation (13) is classical in EM algorithms for mixture estimation (see Bishop (2006)) while Equations (10) and (11) are derived from the model that we proposed.

Acknowledgments

This work was supported by the FUI-WISE AAP21 Minalogic project, funded by BPI France. This work has been partially supported by MIAI at Univ. Grenoble Alpes (ANR-19-P3IA-0003), chair on detection of pollutants and by an ANR grant to Simon Barthelmé (GenGP, ANR-16-CE23-0008).

References

Albert, K.J., Lewis, N.S., Schauer, C.L., Sotzing, G.A., Stitzel, S.E., Vaid, T.P., Walt, D.R., 2000. Cross-Reactive Chemical Sensor Arrays. Chemical Reviews 100, 2595–2626.

Artursson, T., Eklöv, T., Lundström, I., Mårtensson, P., Sjöström, M., Holmberg, M., 2000. Drift correction for gas sensors using multivariate methods. Journal of Chemometrics 14, 711–723.

Bishop, C., 2006. Pattern Recognition and Machine Learning. Information Science and Statistics, Springer-Verlag.

Boeker, P., 2014. On 'Electronic Nose' methodology. Sensors and Actuators B: Chemical 204, 2–17.

Brenet, S., John-Herpin, A., Gallat, F.X., Musnier, B., Buhot, A., Herrier, C., Rousselle, T., Livache, T., Hou, Y., 2018. Highly-Selective Optoelectronic Nose Based on Surface Plasmon Resonance Imaging for Sensing Volatile Organic Compounds. Analytical Chemistry .

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1–38.

Di Carlo, S., Falasconi, M., 2012. Drift correction methods for gas chemical sensors in artificial olfaction systems: techniques and challenges, in: Advances in Chemical Sensors. InTech.

Di Carlo, S., Falasconi, M., Sanchez, E., Scionti, A., Squillero, G., Tonda, A., 2011. Increasing pattern recognition accuracy for chemical sensing by evolutionary based drift compensation. Pattern Recognition Letters 32, 1594–1603.

Fonollosa, J., Fernández, L., Gutiérrez-Gálvez, A., Huerta, R., Marco, S., 2016. Calibration transfer and drift counteraction in chemical sensor arrays using Direct Standardization. Sensors and Actuators B: Chemical 236, 1044–1053.

Haugen, J.E., Tomic, O., Kvaal, K., 2000. A calibration method for handling the temporal drift of solid state gas-sensors. Analytica Chimica Acta 407, 23–39.

Hierlemann, A., Gutierrez-Osuna, R., 2008. Higher-Order Chemical Sensing. Chemical Reviews 108, 563–613.

Holmberg, M., Artursson, T., 2004. Drift Compensation, Standards, and Calibration Methods, in: Pearce, T.C., Schiffman, S.S., Nagle, H.T., Gardner, J.W. (Eds.), Handbook of Machine Olfaction. John Wiley & Sons, Ltd, pp. 325–346.

Holmberg, M., Davide, F.A.M., Di Natale, C., D'Amico, A., Winquist, F., Lundström, I., 1997. Drift counteraction in odour recognition applications: lifelong calibration method. Sensors and Actuators B: Chemical 42, 185–194.

Maho, P., Dolcinotti, C.L., Livache, T., Herrier, C., Andreev, A., Comon, P., Barthelmé, S., 2019. Olfactive robot for gas discrimination over several months using a new optoelectronic nose, in: 2019 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN).

Maho, P., Herrier, C., Livache, T., Comon, P., Barthelme, S., 2020a. Real-time gas recognition and gas unmixing in a robot application. URL: `https://hal.archives-ouvertes.fr/hal-02448737`. to appear.

Maho, P., Herrier, C., Livache, T., Rolland, G., Comon, P., Barthelmé, S., 2020b. Reliable chiral recognition with an optoelectronic nose. Biosensors and Bioelectronics 159, 112183.

Marco, S., 2014. The need for external validation in machine olfaction: emphasis on health-related applications. Analytical and Bioanalytical Chemistry 406, 3941–3956.

Marco, S., Gutierrez-Galvez, A., 2012. Signal and data processing for machine olfaction and chemical sensing: a review. IEEE Sensors Journal 12, 3189–3214.

McLachlan, G.J., Krishnan, T., 2007. The EM algorithm and extensions. volume 382. John Wiley & Sons.

Padilla, M., Perera, A., Montoliu, I., Chaudry, A., Persaud, K., Marco, S., 2010. Drift compensation of gas sensor array data by Orthogonal Signal Correction. Chemometrics and Intelligent Laboratory Systems 100, 28–35.

Persaud, K., Dodd, G., 1982. Analysis of discrimination mechanisms in the mammalian olfactory system using a model nose. Nature 299, 352–355.

Rudnitskaya, A., 2018. Calibration Update and Drift Correction for Electronic Noses and Tongues. Frontiers in Chemistry 6.

Vergara, A., Vembu, S., Ayhan, T., Ryan, M.A., Homer, M.L., Huerta, R., 2012. Chemical gas sensor drift compensation using classifier ensembles. Sensors and Actuators B: Chemical 166-167, 320–329.

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