Sensors and Actuators B 225 (2016) 288–297

Contents lists available at ScienceDirect

Sensors and Actuators B: Chemical

journal homepage: www.elsevier.com/locate/snb

Calibration transfer and drift compensation of e-noses via coupled task learning

Ke Yan [a], David Zhang [b,*]

[a] Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
[b] Biometric Research Centre, Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 2 July 2015
Received in revised form 16 October 2015
Accepted 12 November 2015

Keywords:
Electronic nose
Calibration transfer
Drift
Transfer learning
Multi-task learning
Transfer sample

ABSTRACT

The problems of instrumental variation and sensor drift are receiving increasing concerns in the field of electronic noses. Because the two problems can be uniformly viewed as a variation of the data distribution in the feature space, they can be handled by algorithms such as transfer learning. In this paper, we propose a novel algorithm framework called transfer sample-based coupled task learning (TCTL). It is based on transfer learning and multi-task learning. Given labeled samples in the source domain (i.e. from the master device or without drift) and a small set of transfer samples as inputs, TCTL simultaneously learns a prediction model for data in the source domain and one for data in the target domain (i.e. from the slave device or with drift). The transfer samples are incorporated into a regularization term of the objective function. TCTL is an extensible framework that can apply to various classification and regression models. When combined with the standardization error-based model improvement (SEMI) strategy, its accuracy can be further enhanced. Experiments on a multi-device dataset and a popular long-term drift dataset show that the proposed algorithms achieve better accuracy compared with typical existing methods with much fewer auxiliary samples needed, which proves their efficacy and usability in real-life applications.

1. Introduction

As an important part of an electronic nose (e-nose) system, a prediction model is learned from a group of training samples and used to recognize odors. Ideally, the model is expected to perform well under a variety of conditions. However, there are several factors that may decrease a model's universality. For example, owing to the variations in the manufacture of gas sensors and e-noses, two e-noses of the same model can respond differently to the same gas sample. Because of the aging and poisoning of gas sensors and the change in operating conditions, the sensors' sensitivity characteristics change over time in a complex manner [1–3]. These factors make the distribution of test samples different from that of training samples, which will degrade the performance of prediction models in real-life applications.

A number of methods have been developed to cope with this problem. Some of them were originally proposed for spectroscopic data [4–6], but they can also be applied to e-noses. Methods to address instrumental variation are often called calibration transfers. A widely used idea is to transform the data from the slave device (with which test samples are collected) to match the master device (with which training samples are collected) [1,4,5,7–10]. We will refer to this method as variable standardization hereinafter. A set of transfer samples are measured on both the master and the slave devices. Then, regression algorithms are applied to fit the master variables with those of the slave.

When the problem concerns sensor drift over time, variable standardization is also a viable method. Multiplicative drift correction (MDC) [11,12] can be regarded as a simplified version. Another category of popular algorithms is component correction (CC). CC based on principal component analysis (CC-PCA) [11], which was proposed by Artursson et al., finds the direction of drift by applying PCA to transfer samples then removes the component on the direction from the data. Orthogonal signal correction (OSC) [6,13] is a CC-like method. It finds the undesired component by calculating the subspace that is orthogonal to the target variable. One drawback of CC-like methods is that when the drift is complex, it may be difficult to accurately separate the directions of useful information and drift [12].

Recently, Vergara et al. [3] applied an ensemble strategy to enhance the robustness of classifiers and address sensor drift. It showed improved accuracy on an e-nose dataset collected over a period of three years but required many labeled drifted samples

* Corresponding author.
E-mail addresses: yank10@mails.tsinghua.edu.cn (K. Yan), csdzhang@comp.polyu.edu.hk (D. Zhang).

http://dx.doi.org/10.1016/j.snb.2015.11.058
0925-4005/

to train multiple classifiers. Meanwhile, semi-supervised learning approaches [14,15] mainly rely on unlabeled drifted samples for drift counteraction, which are easier to collect. Zhang et al. [16] extended an extreme learning machine for domain adaptation. The algorithm has a similar principle to that of the Tikhonov regularization method in [17]. It can achieve high accuracy on the drift dataset in [3] if a sufficient number of labeled and unlabeled drifted samples is provided.

From the perspective of pattern recognition, instrumental variation and sensor drift over time are essentially a variation of the data distribution in the feature space. Following the terms in transfer learning [18], we can refer to the data from the master device or without drift as the data from the source domain, and the data from the slave device or with drift as the data from the target domain, because the goal is to transfer the knowledge from the former one (source) to the latter one (target). In e-nose applications, we usually have enough labeled source data to train a prediction model. However, the labeled target data are scarce or hard to acquire, making it difficult to train a new model for the target domain. In this paper, we propose a novel algorithm framework called transfer sample-based coupled task learning (TCTL) to learn it. TCTL belongs to a special case of multi-task learning (MTL) with two tasks. MTL is a type of inductive transfer learning method [18] that aims at learning multiple models simultaneously [19–21]. When the tasks are related, information can be shared across models during the learning process to improve their accuracies [19]. In the proposed algorithm, labeled source data and a set of transfer samples are used to learn the source and target models jointly. The transfer samples collected in both the source and the target domains are leveraged for knowledge transfer. Coefficients of the two models are connected through regularization terms in the objective function of TCTL.

TCTL is an algorithm framework that can be combined with various prediction algorithms. Two popular classification/regression algorithms, namely, logistic regression and ridge regression, are demonstrated under the framework of this paper. They are evaluated on two e-nose datasets. One is a multi-device dataset collected with three e-noses [10], and the other is the long-term drift dataset in [3]. Experimental results show that the proposed methods can improve the prediction accuracy of target e-nose data and outperform typical existing algorithms with much fewer auxiliary samples needed. With the help of the standardization error-based model improvement (SEMI) strategy [10], the accuracies can be further enhanced.

The rest of this paper is organized as follows. Section 2 describes the TCTL framework in detail and shows how logistic regression and ridge regression should be extended under the framework. Section 3 introduces the experimental configurations, including the two datasets used in the paper and the related data analysis procedure. Section 4 presents the results of the experiments carried on target e-nose data and provides some discussion. Section 5 concludes the paper.

2. Transfer sample based coupled task learning (TCTL)

The objective of TCTL is to learn a prediction model for target e-nose data (from the slave device or with drift) on the basis of labeled source data (from the master device or without drift) and a set of transfer samples. In this section, we use non-bold capital letters for matrices, bold lowercase ones for vectors, and non-bold lowercase ones for scalars. Suppose $X_S \in \mathbf{R}^{n\times p}$ is the matrix of source data with each row as a feature vector; n is the number of labeled source samples; p is the number of variables; $\boldsymbol{y}_S \in \mathbf{R}^n$ is the label vector of X_S; $T_S, T_T \in \mathbf{R}^{n_t\times p}$ are the matrices of the source and target transfer samples, respectively; n_t is the number of transfer samples; and $\boldsymbol{\beta}_S$, $\boldsymbol{\beta}_T \in \mathbf{R}^p$ are the source and target prediction models to be estimated, respectively. In TCTL, the two models are learned by solving

$$\min_{\boldsymbol{\beta}_S,\boldsymbol{\beta}_T} L(X_S,\boldsymbol{y}_S,\boldsymbol{\beta}_S) + \lambda_1\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2 + \lambda_2\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2. \quad (1)$$

In Eq. (1), the first term is the loss function, which represents the training error of the labeled source data. The second term links $\boldsymbol{\beta}_S$ and $\boldsymbol{\beta}_T$ with the transfer samples and requires that each source transfer sample projected by the source model be close to its corresponding target transfer sample projected by the target model. The third term requires the two models to be similar under the assumption that the difference between source and target distributions is small. To reduce the difference before applying TCTL, one can preprocess the source and target data separately with standard normal variate (SNV) [1]; i.e., each variable is centered and scaled by the mean and standard deviation calculated from the transfer samples. $\lambda_1, \lambda_2 > 0$ are regularization parameters that control the weights of the terms.

There are two essential regularization terms in Eq. (1), the transfer sample term $\lambda_1\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2$ and the similarity term $\lambda_2\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2$. If the transfer sample term is absent, identical $\boldsymbol{\beta}_S$ and $\boldsymbol{\beta}_T$ will be learned, which only fit the source distribution and cannot reflect the difference between source and target distributions. If the similarity term is absent, the control over $\boldsymbol{\beta}_T$ will be too weak. Because the number of transfer samples is often small, there will be infinite solutions to $\boldsymbol{\beta}_T$ that can minimize the transfer sample term and make it zero. The similarity term also occurs in analogical forms in some other MTL-based algorithms [18,21].

The similarity term is closely related to conventional variable standardization methods. In variable standardization, the target samples are transformed by a standardization model, which is obtained by fitting each source variable with one or more target variables based on the transfer samples. Then the standardized target samples can be predicted by source prediction models. The standardization model can be viewed as a matrix $M \in \mathbf{R}^{p\times p}$ (in linear cases), which makes T_TM close to T_S [5]. In the prediction step, each sample is projected by $\boldsymbol{\beta}_S$. Thus, the goal is actually making T_TM close to T_S in the projected direction—in other words, minimizing $\|T_S\boldsymbol{\beta}_S - T_TM\boldsymbol{\beta}_S\|_2^2$. We can find that $\boldsymbol{\beta}_T$ in TCTL actually corresponds to a combination of M and $\boldsymbol{\beta}_S$. By merging M into $\boldsymbol{\beta}_T$, we can omit the step of variable standardization and directly obtain the target prediction model. In variable standardization methods, M is often supposed to have special forms to avoid overfitting. For example, univariate algorithms [8,9] require M to be diagonal. Piecewise direct standardization (PDS) [5,10] requires M to be banded. However, when the variation in distribution is complex, the constraints on M may lower its ability to fully model the mapping between source and target variables. Conversely, direct standardization (DS) [5,7] allows M to be a full matrix. However, it is prone to overfitting when the number of transfer samples is small, because all elements in M must be estimated. TCTL allows M to be full while reducing the number of coefficients to be estimated by merging M into $\boldsymbol{\beta}_T$ and thus may have better performance when the variation in distribution is complex as well as being less prone to overfitting.

The regularized multi-task learning (RMTL) method [21] also applies an MTL framework. However, instead of aligning a set of transfer samples, RMTL requires a set of labeled target samples to be well predicted by the target model. In other words, it changes the transfer sample term in TCTL into a loss function of target samples, which is similar to the principle of DAELM [16] and the method in [17]. In real-life applications, labeled target samples are sometimes hard to acquire. For example, if a breath analysis system based on an e-nose [22] is produced in batch, it is impractical to collect breath samples from patients with each new device for calibration transfer. Conversely, transfer samples are widely used for both calibration transfer [1,5,8] and correction of sensor drift over time

[2,11,12]. Transfer samples often comprise standard gases, which are reproducible. The correspondence relationship in two groups of transfer samples brings mapping information between source and target variables. Thus, fewer transfer samples than labeled target samples may be needed for comparable transfer performance.

Logistic regression and ridge regression are popular algorithms for classification and regression, respectively. As examples for the TCTL framework, their formulations and solutions under TCTL are briefly described below.

2.1. Logistic regression-based classification

A detailed introduction to logistic regression (LR) can be found in [23]. We denote $\boldsymbol{x}^{(i)}$ as the ith training sample and $y^{(i)}$ as its label. $y^{(i)} = 1$ if $\boldsymbol{x}^{(i)}$ belongs to the positive class and is 0 otherwise. $X = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}]^{\mathrm{T}}$ and $\boldsymbol{y} = [y^{(1)}, \ldots, y^{(n)}]^{\mathrm{T}}$. LR models are usually fitted by maximum likelihood estimation, which seeks to maximize the log-likelihood function:

$$\ell_{LR}(X, \boldsymbol{y}, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} \log h_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)}) + (1 - y^{(i)}) \log(1 - h_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)}))\right), \tag{2}$$

where $h_{\boldsymbol{\beta}}$ is the sigmoid decision function:

$$h_{\boldsymbol{\beta}}(\boldsymbol{x}) = \mathrm{sigmoid}(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{x})}. \tag{3}$$

A test sample $\boldsymbol{x}$ is classified into the positive class if $h_{\boldsymbol{\beta}}(\boldsymbol{x}) \geq 0.5$. Because $h_{\boldsymbol{\beta}}(\boldsymbol{x})$ is between 0 and 1, it can be viewed as the probability of the test sample belonging to the positive class, which is also the characteristic of LR. The LR model under the TCTL framework can be formulated as minimizing the following objective function:

$$J_{LR}(\boldsymbol{\beta}_S, \boldsymbol{\beta}_T) = -\ell_{LR}(X_S, \boldsymbol{y}_S, \boldsymbol{\beta}_S) + \frac{\lambda_1}{2n_t}\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2, \tag{4}$$

whose gradient is given by:

$$\frac{\partial J_{LR}}{\partial \boldsymbol{\beta}_S} = \frac{1}{n}\sum_{i=1}^{n}((h_{\boldsymbol{\beta}}(\boldsymbol{x}^{(i)}) - y^{(i)})\boldsymbol{x}^{(i)}) + \frac{\lambda_1}{n_t}T_S^{\mathrm{T}}(T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T) + \lambda_2(\boldsymbol{\beta}_S - \boldsymbol{\beta}_T); \tag{5}$$

$$\frac{\partial J_{LR}}{\partial \boldsymbol{\beta}_T} = -\frac{\lambda_1}{n_t}T_T^{\mathrm{T}}(T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T) - \lambda_2(\boldsymbol{\beta}_S - \boldsymbol{\beta}_T). \tag{6}$$

Then, the problem can be solved using numerical optimization methods such as conjugate gradient. The model above is designed for binary-class cases. In multi-class cases, multiple LR models are trained using the one-vs-all strategy and $\boldsymbol{x}$ is classified into the class whose decision function has the largest value.

2.2. Ridge regression-based regression

Ridge regression (RR) is a simple but effective regression algorithm [1,23]. The problem formulation of RR is

$$\min_{\boldsymbol{\beta}}\|X\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2. \tag{7}$$

The difference between RR and ordinary least squares is that RR shrinks the coefficients in $\boldsymbol{\beta}$ using the Tikhonov regularization term in order to reduce the variance of the model. Under the TCTL framework, the RR problem can be formulated as minimizing the following objective function

$$J_{RR}(\boldsymbol{\beta}_S, \boldsymbol{\beta}_T) = \|X_S\boldsymbol{\beta}_S - \boldsymbol{y}_S\|_2^2 + \lambda_1\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2 + \lambda_2\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2 + \lambda(\|\boldsymbol{\beta}_S\|_2^2 + \|\boldsymbol{\beta}_T\|_2^2). \tag{8}$$

By setting the gradient of Eq. (8) to zero, the closed-form solution to the problem can be derived:

$$\begin{pmatrix}\boldsymbol{\beta}_S \\ \boldsymbol{\beta}_T\end{pmatrix} = \begin{pmatrix} X_S^{\mathrm{T}}X_S + \lambda_1 T_S^{\mathrm{T}}T_S + (\lambda_2 + \lambda)I & -\lambda_1 T_S^{\mathrm{T}}T_T - \lambda_2 I \\ -\lambda_1 T_T^{\mathrm{T}}T_S - \lambda_2 I & \lambda_1 T_T^{\mathrm{T}}T_T + (\lambda_2 + \lambda)I \end{pmatrix}^{-1} \times \begin{pmatrix} X_S^{\mathrm{T}}\boldsymbol{y}_S \\ \boldsymbol{0}\end{pmatrix}. \tag{9}$$

2.3. Standardization error-based model improvement (SEMI)

In [10], we proposed a strategy to improve the transfer ability of prediction models, namely standardization error-based model improvement (SEMI). The key idea of the strategy is to add a Tikhonov regularization term $\lambda\sum_{j=1}^{p}(w_j\beta_j)^2$ into the objective function of various prediction algorithms. w_j is the standardization error (SE) of the jth variable; i.e., the deviation between the jth standardized target variable and the jth source variable in transfer samples. SEMI penalizes the variables with large SEs, making the trained models rely more on well-standardized variables, and thus is less sensitive to variation in distribution. The strategy can also be combined with TCTL. First, the training and transfer samples are preprocessed using SNV. Next, w_j is defined as the deviation between the jth variable in the source and target transfer samples:

$$w_j = \sqrt{\sum_{i=1}^{n_t}(t_{S,ij} - t_{T,ij})^2}. \tag{10}$$

$t_{S,ij}$ means the element in the ith row (sample) and jth column (variable) of T_S. The mean of $\boldsymbol{w}$ can be scaled to 1. Then, $\boldsymbol{\beta}_S$ and $\boldsymbol{\beta}_T$ are learned by solving

$$\min_{\boldsymbol{\beta}_S, \boldsymbol{\beta}_T} L(X_S, \boldsymbol{y}_S, \boldsymbol{\beta}_S) + \lambda_1\|T_S\boldsymbol{\beta}_S - T_T\boldsymbol{\beta}_T\|_2^2 + \lambda_2\|\boldsymbol{\beta}_S - \boldsymbol{\beta}_T\|_2^2 + \lambda\sum_{j=1}^{p}w_j^2(\beta_{S,j}^2 + \beta_{T,j}^2). \tag{11}$$

Compared with Eq. (1), the Tikhonov regularization term has been included in Eq. (11). The objective functions and solutions of LR and RR in Sections 2.1 and 2.2 can be modified according to Eq. (11). In Section 4, we will show that the SEMI strategy can further improve the performance of TCTL.

It is worth noting that in the equations above, the intercept term of $\boldsymbol{\beta}$ is not considered for clarity. To include the term, a constant component ($x_0 = 1$) should be added to all samples. Then, an additional coefficient can be added to $\boldsymbol{\beta}_S$ and $\boldsymbol{\beta}_T$ ($\beta_{S,0}$ and $\beta_{T,0}$). However, the Tikhonov regularization terms in RR and SEMI should not include $\beta_{S,0}$ and $\beta_{T,0}$, because we do not expect the intercept terms to be shrunk. They should not be involved in the similarity term of TCTL either if the intercept terms in the source and target models are not expected to be similar.

3. Experimental details

To evaluate the performance of the proposed algorithm in the situations of instrumental variation and sensor drift over time, two e-nose datasets are adopted in this paper. Their composition and the data analysis procedures will be introduced in this section.

Table 1
Composition of the multi-device dataset. [10]

Group	Gas sample	#Samples	Notes
1	Acetone	16	8 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10, 20 ppm), 2 samples per concentration.
2	Hydrogen	18	9 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50 ppm), 2 samples per concentration.
3	Ammonia	14	7 concentrations (0.1, 0.2, 0.5, 1, 2, 5, 10 ppm), 2 samples per concentration.
4	Normal breath	80	Collected from 24 healthy subjects in different days, 2–6 samples per subject.
5	Acetone + breath	40	Acetone at 8 concentrations (0, 0.2, 0.3, 0.7, 1.7, 3.3, 5.0, 6.7 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects.
6	Hydrogen + breath	40	H_2 at 8 concentrations (0, 0.4, 0.7, 1.7, 4.2, 8.3, 12.5, 16.7 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects.
7	Ammonia + breath	40	NH_3 at 8 concentrations (0, 0.3, 0.8, 1.7, 2.5, 3.3, 4.2, 5.0 ppm), 5 samples per concentration, mixed with normal breath from 5 subjects.

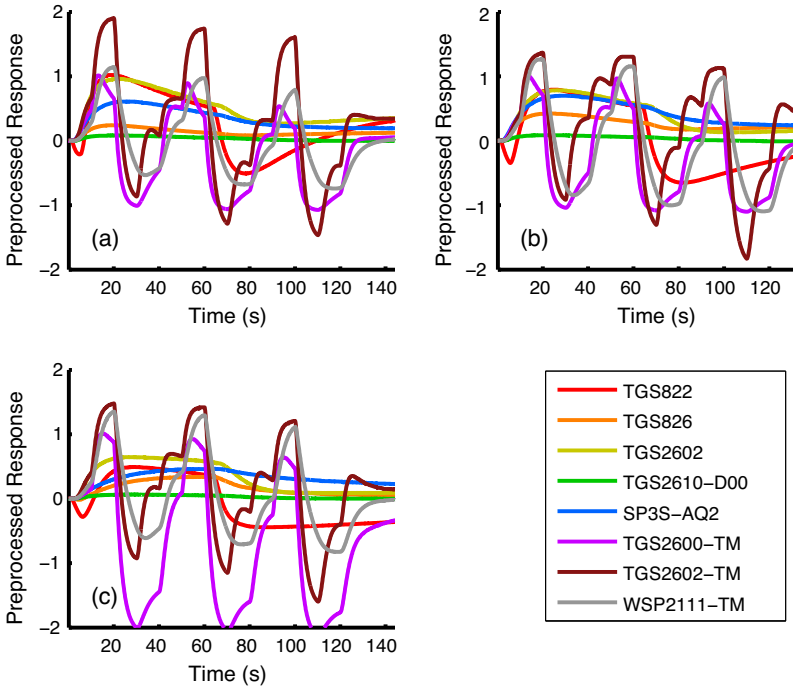

Fig. 1. A normal breath sample in the multi-device dataset measured with the three devices of the same model. Plots (a)–(c) correspond to devices 1–3, respectively. The names of the sensors are displayed in the legend. Sensors under temperature modulation are marked with "-TM".

3.1. Multi-device dataset

In our previous work [10], a dataset collected with three e-noses of the same model was described. The portable e-nose is equipped with an array of 11 sensors. Among them, eight metal oxide semiconductor (MOS) sensors with stable performance are chosen for analysis—namely, TGS822, TGS826, TGS2602, TGS2610-D00, SP3S-AQ2, TGS2600-TM, TGS2602-TM, and WSP2111-TM. The last three sensors with the suffix "-TM" are temperature-modulated sensors [24]. They are heated by a staircase voltage oscillated between 3 and 6 V. The measurement procedure of each gas sample lasts for 144 s. In the baseline stage (1 s), the baseline values of the sensors are recorded. In the injection stage (7 s), gas is drawn from a gas bag to the gas room. In the reaction stage (56 s), the responses of the sensors approach their steady states. The gas room is purged with clean air in the purge stage (80 s). The sampling frequency of the e-nose is 8 Hz. Further details of the e-nose are presented in [22].

A total of 248 samples were measured with each e-nose, which can be further divided into seven groups. Groups 1–3 include three chemicals at different concentrations. Groups 4–7 are breath samples from healthy and simulated patients, because disease screening and monitoring by breath analysis are important applications of e-noses [25,26,22]. The three chemicals mentioned above are typical breath biomarkers of certain diseases [10], so we mixed them with normal breath samples to simulate breath from patients. Details of the dataset are listed in Table 1.

After a gas sample is measured, the baseline values are subtracted from the sensors' responses to remove baseline drift. Fig. 1 shows the baseline-removed responses of the same breath sample measured with the three devices. We can see the inconsistency among the devices. We assign device 1 as the master device and devices 2 and 3 as the slaves. The goal is to enhance the prediction accuracies of slave devices using only labeled samples from the master device and transfer samples. The transfer samples will be chosen from the three types of chemicals (groups 1–3) using the Kennard-Stone algorithm [8,27]. If a prediction task is performed on groups 1–3, only the training samples will be used for selection to avoid overlap between transfer samples and test ones. The distribution of part of the samples is illustrated in Fig. 2. One can observe the variation in distribution of different devices. In addition, although the transfer samples and breath samples are different kinds of gases, their inter-device variation in distribution are roughly in the same direction. Therefore, it is possible to learn the models for breath samples from slave devices by aligning the transfer samples.

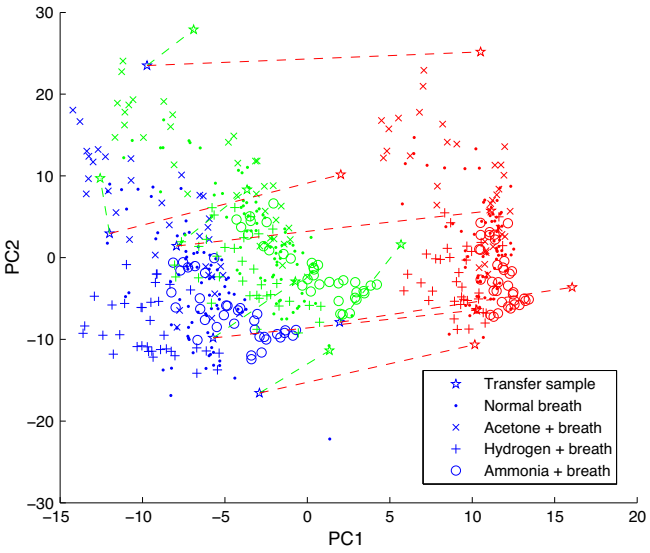

Fig. 2. Distribution of the samples in the multi-device dataset. For clarity, only groups 4–7 and the six transfer samples are shown. Blue, green, and red points indicate samples from device 1, 2, and 3, respectively. Dashed lines connect the transfer sample pairs between the master and slave devices. The *x*- and *y*-axes correspond to the first two dimensions of the principal components of the samples.

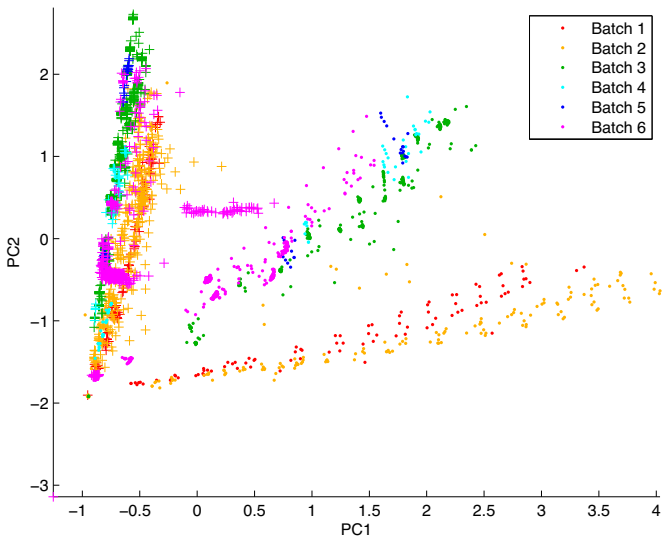

Fig. 3. Example of the drift across batches 1–6 in the long-term drift dataset. Dots and plus signs represent ammonia and acetaldehyde samples, respectively. Different colors indicate different batches. The *x*- and *y*-axes correspond to the first two dimensions of the principal components of the samples.

Four classification tasks and six regression tasks are designed for the dataset. The classification tasks include a multi-class task (distinguishing the three chemicals) and three binary-class tasks (distinguishing normal breath samples from each mixed group. Note that in data groups 5–7, the 5 samples with 0 ppm chemical mixed are removed.) In each task, we randomly choose an equal size of samples from each class as training samples, with the size being half of the number of samples in the class with the fewest samples. The random experiment is repeated 20 times. Then, the average accuracy is computed. With regard to the six regression tasks, the concentrations of chemicals in data groups 1–3 and 5–7 are predicted. The leave-one-out strategy is used within each task. Then, the average root mean square error (RMSE) is computed.

Selected geometry features extracted from sensors' response curves are proved to be more effective than traditional features such as steady-state responses [28]. Based on the results in [28], 18 phase features [29] and 36 derivative features are extracted from each temperature-modulated sensor, and 21 derivative features are extracted from each ordinary sensor. Thus, the length of the final feature vector for each sample is 267. The phase features are calculated in each "stair" of the response curves of the temperature-modulated sensors. The derivative features are downsampled from the derivative of the response curves. A more detailed description of the features may be found in [28].

3.2. Long-term drift dataset

A large e-nose dataset containing long-term drift was introduced in [3] and [30]. The dataset, which is accessible in [31], was collected using an array of 16 MOS sensors under strictly controlled operating conditions. Six chemicals (ammonia, acetaldehyde, acetone, ethylene, ethanol and toluene) at different concentrations (5–1000 ppm) were measured over a period of 36 months, summing up to 13910 samples in total. The authors of [3] split the dataset into 10 batches in chronological order. The goal is to enhance the classification accuracy of the gas type on batch t ($t = 2, 3, \ldots, 10$), where sensor drift has taken place.

The experimental setting in this paper is similar to setting 1 in [15] and [16]. It also resembles how e-noses are usually used in practical applications. All samples in batch 1 are adopted as labeled training samples. In batch t, meanwhile, only a small group of transfer samples are available, which link the drifted batch with batch 1. However, transfer samples were not explicitly collected in this dataset. Therefore, we first find the candidate transfer samples for batch t, which are defined as the overlapping samples in batch 1 and t (samples with the same gas type and concentration). If a sample was measured multiple times, the measurement closest to the median of all measurements is chosen. Then, the actual transfer samples are selected using the Kennard–Stone algorithm. Note that because the magnitudes of the variables are greatly varied in this dataset, it is better to preprocess the candidate transfer samples with SNV before selection. Because the sample composition in each batch is not identical (summarized in [15]), the gas type and concentration of transfer samples selected for each target batch is different.

In the dataset, eight features are extracted from each sensor's response curve, including two steady-state features and six exponential moving average features [3]. Thus, the length of the final feature vector for each sample is 128. To visually inspect the drift across batches, we draw a scatter diagram in Fig. 3. It can be found that the ammonia samples (dots) drift roughly in the upper-left direction as the batch index increases (except batch 2), whereas the drift of acetaldehyde samples (plus signs) is not obvious. There are also some outliers that do not comply with the general trend. These observations suggest the complexity of this dataset: the drift is class-dependent and cannot be well described by a simple model [16].

4. Results and discussion

4.1. Multi-device dataset

Calibration transfer from device 1 (master device) to devices 2 and 3 (slave devices) is discussed in this section. Six transfer samples were selected by the Kennard–Stone algorithm, because we found that further increasing the number of transfer samples could not improve the accuracy significantly and would increase the complexity of the future calibration transfer process. (Note that the numbers of the training samples in the four classification tasks are 21, 36, 36, 36; those for the six regression tasks are 15, 17,

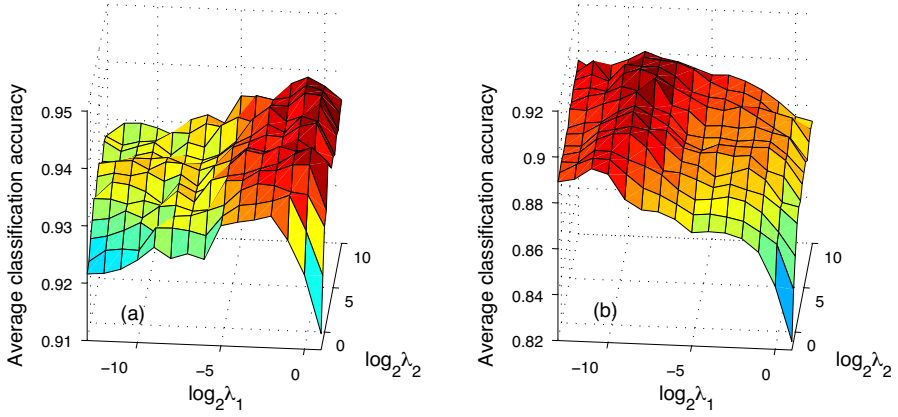

Fig. 4. Average classification accuracy of the proposed method with different parameter settings. Plots (a) and (b) correspond to devices 2 and 3 in the multi-device dataset, respectively. Parameters λ_1 and λ_2 in Eq. (11) are searched in $\{2^{-13}, 2^{-12}, \ldots, 2^{1}\}$ and $\{2^{-2}, 2^{-1}, \ldots, 2^{10}\}$, respectively.

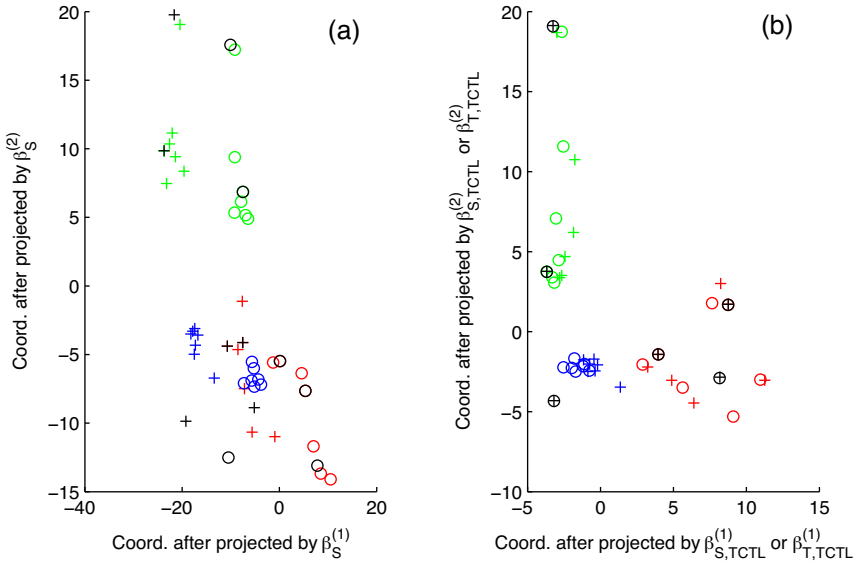

Fig. 5. Samples in three classes are projected by various models. Markers in different colors are samples from different classes, except the black ones, which represent transfer samples. Circles and plus signs are samples from the master and slave, respectively. In plot (a), samples from both devices are projected by the master model learned by LR. In plot (b), the master samples are projected by the master model ($\boldsymbol{\beta}_S$) learned by LR under TCTL, whereas the slave samples are projected by the slave model ($\boldsymbol{\beta}_T$) learned by it. $\boldsymbol{\beta}^{(i)}$ means a model learned for the ith class.

13, 39, 39, 39.) In the proposed algorithm, the parameters λ_1, λ_2 and λ (Eq. (11)) are related to the results. We empirically set the regularization parameter λ to 0.1 in this dataset. Then, λ_1 and λ_2 were changed in the range of $\{2^{-13}, 2^{-12}, \ldots, 2^{1}\}$ and $\{2^{-2}, 2^{-1}, \ldots, 2^{10}\}$, respectively, to observe the change in average classification accuracies. The results are shown in Fig. 4. The base classifier was logistic regression as introduced in Section 2.1. It can be noted that the optimal value of λ_1 for device 2 is larger than that for device 3. λ_1 controls the weight of the transfer sample term in the objective function of TCTL. Increasing its value means that more emphasis is put on requiring that the transfer samples from both devices are close in the projected space. In fact, devices 1 and 2 were made in a batch, whereas device 3 was made earlier. Its gas route was slightly different from the other two devices; Its sensor array was also more aged. Thus, the variation of the distribution from device 1 to device 3 is more complex. The knowledge cannot be fully transferred from device 1 to device 3 by simply aligning the six transfer samples. Thus, a smaller λ_1 is better, as a larger one may cause overfitting. The variation of distribution from device 1 to 2 can be better modeled by the transfer samples, so a relatively larger λ_1 is beneficial. From Fig. 4, we also find that the average classification accuracy is slightly improved as λ_2 becomes larger.

Fig. 5 illustrates an example of the effect of TCTL. The samples in three classes are from data groups 1–3. In plot (a), half of the master samples are used to train two classification models to distinguish class 1 or 2 from other classes with the one-vs-all strategy. Then, the other half of the samples from both master and slave devices are projected by the two models and shown in the plot. We find that the samples from different devices are not clustered, so it is not feasible to use the master model to directly classify the slave data. In plot (b), TCTL is used to simultaneously learn both master and slave models. When the samples from different devices are projected by their corresponding models, the samples in the same class lie in the same region. TCTL achieves this by aligning the six transfer samples in the projected space, which can be discovered in the plot.

The accuracies of the four classification tasks and their averages are listed in Tables 2 and 3 . TCTL is compared with several other methods. The parameter λ in the SEMI strategy and λ_1, λ_2 in TCTL were tuned by grid search. For “no transfer”, slave data were directly predicted by the classification model trained on master

Table 2
Comparison of classification accuracy (%) on device 2 of the multi-device dataset.

	Group 1 vs. 2 vs. 3 [a]	4 vs. 5	4 vs. 6	4 vs. 7	Average
No transfer	62.86 ± 11.93	76.91 ± 13.77	79.41 ± 9.96	62.06 ± 8.31	70.31 ± 3.81
Only SNV	94.29 ± 4.26	87.50 ± 5.55	90.29 ± 7.21	86.32 ± 3.61	89.60 ± 2.36
Var. stdd.	93.10 ± 5.46	89.26 ± 5.07	90.88 ± 7.13	87.06 ± 4.99	90.08 ± 2.56
TCTL	97.38 ± 3.62	**91.62 ± 4.89**	92.79 ± 3.99	92.21 ± 4.30	93.50 ± 2.09
Var. stdd. + SEMI	**99.29 ± 2.33**	87.50 ± 6.74	89.41 ± 7.29	83.24 ± 7.21	89.86 ± 3.62
TCTL + SEMI	98.10 ± 2.85	91.18 ± 4.48	**93.24 ± 4.28**	**92.50 ± 4.10**	**93.75 ± 2.06**
Train with slave	99.29 ± 1.74	91.91 ± 3.68	93.68 ± 3.85	93.97 ± 4.21	94.71 ± 2.38

Bold values indicate the best results.
[a] 3-class classification problem. The subsequent three columns correspond to binary-class problems.

Table 3
Comparison of classification accuracy (%) on device 3 of the multi-device dataset.

	Group 1 vs. 2 vs. 3[a]	4 vs. 5	4 vs. 6	4 vs. 7	Average
No transfer	33.33 ± 0.00	62.21 ± 9.98	70.59 ± 15.33	70.59 ± 14.6	59.18 ± 7.25
Only SNV	77.62 ± 11.78	72.06 ± 7.66	84.85 ± 8.76	80.15 ± 10.62	78.67 ± 4.77
Var. stdd.	84.52 ± 10.91	77.50 ± 8.11	88.09 ± 7.30	83.82 ± 7.48	83.48 ± 3.61
TCTL	84.05 ± 10.51	79.85 ± 7.03	87.21 ± 7.03	88.38 ± 6.51	84.87 ± 3.67
Var. stdd. + SEMI	**96.43 ± 4.05**	**81.18 ± 7.29**	87.06 ± 6.91	87.94 ± 9.15	88.15 ± 3.77
TCTL + SEMI	95.95 ± 8.36	81.03 ± 8.01	**92.50 ± 4.91**	**90.74 ± 4.08**	**90.05 ± 2.81**
Train with slave	99.52 ± 1.47	91.03 ± 5.18	91.32 ± 4.52	93.24 ± 4.49	93.78 ± 2.98

Bold values indicate the best results.
[a] 3-class classification problem. The subsequent three columns correspond to binary-class problems.

Table 4
RMSE of the regression tasks on device 2 of the multi-device dataset.

	Group 1	Group 2	Group 3	Group 5	Group 6	Group 7	Average
No transfer	1.0747	8.9395	5.0125	2.3193	7.9286	3.0213	4.7160
Only SNV	0.9569	7.8591	0.5975	0.6483	3.7534	1.1514	2.4945
Var. stdd.	1.4456	6.5774	1.0140	0.7361	**1.8228**	1.0361	2.1053
TCTL	0.8496	3.6217	1.0162	0.6583	4.1052	1.1401	1.8985
Var. stdd. + SEMI	0.5279	1.5593	0.7299	**0.5028**	2.5161	1.2144	1.1751
TCTL + SEMI	**0.3355**	**0.7636**	**0.4443**	0.5148	2.8840	**0.9030**	**0.9742**
Train with slave	0.6161	3.1674	0.3643	0.4011	1.7559	1.0533	1.2263

Bold values indicate the best results.

Table 5
RMSE of the regression tasks on device 3 of the multi-device dataset.

	Group 1	Group 2	Group 3	Group 5	Group 6	Group 7	Average
No transfer	2.7903	4.5576	3.8079	2.1211	8.5552	3.7641	4.2660
Only SNV	1.7860	12.4726	3.3463	0.8980	3.1779	1.0873	3.7947
Var. stdd.	2.1295	9.9353	2.1740	0.6911	2.4169	1.2938	3.1068
TCTL	0.5923	3.0735	**0.9525**	1.0247	3.8046	**1.0689**	1.7527
Var. stdd. + SEMI	0.7693	2.8323	4.6216	**0.3751**	3.5504	1.1358	2.2141
TCTL + SEMI	**0.2611**	**2.0506**	1.3306	0.4082	**2.3377**	1.5247	**1.3188**
Train with slave	0.5464	2.2178	0.4919	0.5845	1.9160	0.8544	1.1018

Bold values indicate the best results.

data. The accuracy is not good, proving the inconsistency between slave and master devices. The SNV transform can reduce simple additive and multiplicative variations between devices, so the accuracy is improved with "only SNV". Variable standardization ("var. stdd.") is commonly used for calibration transfer of e-noses. The robust fit method [8,9] was adopted to standardize slave data after it was preprocessed by SNV. The accuracy is further enhanced. However, the proposed TCTL method achieves better results, which surpasses variable standardization by 3.42 and 1.39 percentage points on devices 2 and 3, respectively. The highest average accuracies on both devices 2 and 3 are obtained by the combination of TCTL and SEMI, which are close to the results obtained by the classification models trained on slave data. Moreover, the calibration transfer accuracies on device 3 are generally lower than those on device 2, indicating that the variation of distribution from device 1 to 3 is more complex.

The results of the regression tasks are presented in Tables 4 and 5 . The parameter λ in ridge regression and the SEMI strategy as well as λ_1, λ_2 in TCTL were tuned by grid search. Other experimental settings were the same as those in the classification tasks. TCTL results in a reduction of RMSE by 9.82% and 43.59% compared with variable standardization on devices 2 and 3, respectively. The lowest average RMSE is achieved by TCTL + SEMI. Note that some results of TCTL + SEMI are even better than the model trained with slave data, which may mainly be caused by three reasons: (1) The size of the dataset is small, so some results may not reflect the true ability of a model; (2) The SEMI strategy encourages the use of stable variables, which can improve the performance of the model; (3) TCTL + SEMI utilizes the information contained in the transfer samples of two devices, which adds additional regularity to the model and can possibly improve its accuracy.

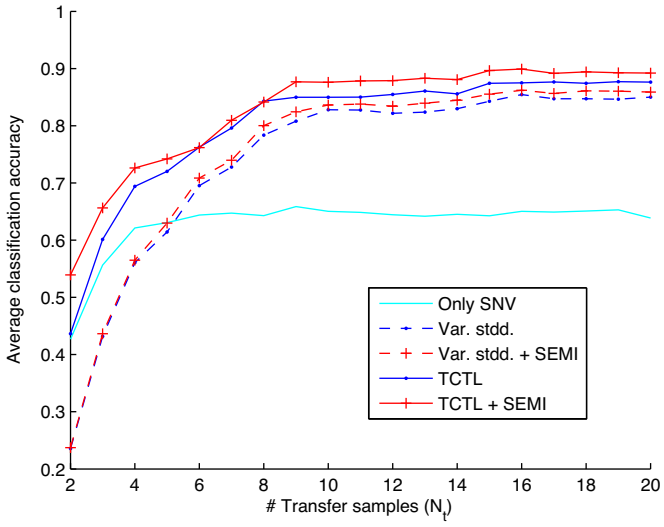

Fig. 6. Performance of several transfer sample-based methods on the long-term drift dataset with respect to different numbers of transfer samples. The *x*-axis represents the number of transfer samples used to transfer knowledge from batch 1 to batch t ($t = 2, 3, \ldots, 10$). The *y*-axis represents the average classification accuracy of batches 2–10.

4.2. Long-term drift dataset

Several researchers have investigated this dataset with state-of-art drift correction algorithms [3,15,16]. However, to the best of our knowledge, our work is the first transfer sample-based method tested on it. Fig. 6 illustrates the performance of several methods as the number of transfer samples (n_t) increases. Logistic regression was used as the base classifier. Direct standardization [5,7] was used as the variable standardization strategy. It is a multivariate strategy, and we found that it yielded better accuracy than univariate strategies such as robust fitting. Ridge regression was the regression algorithm for variable standardization. The parameter λ in ridge regression and the SEMI strategy as well as λ_1, λ_2 in TCTL were tuned by grid search. Fig. 6 shows that as n_t increases, the average classification accuracy generally rises. However, it becomes relatively stable when $n_t \geq 10$. The order of performance is TCTL > variable standardization > only SNV when more than five transfer samples are used. If $n_t \leq 5$, variable standardization has lower accuracy than only SNV, which may be caused by overfitting. However, TCTL is unaffected by the problem for the dataset. Moreover, the SEMI strategy can further improve the accuracy of both variable standardization and TCTL.

The performance of our methods and other typical methods is compared in Table 6. For "no transfer", data in batches 2–10 were directly predicted by the classification model trained on batch 1. The accuracy is unsatisfactory because of the drift across batches. For CC-PCA and OSC, the data in each batch were first centered and scaled using the mean and standard deviation calculated from batch 1. For CC-PCA, ethanol was chosen as the transfer sample, and the number of components to remove is three [11]. Wise's implementation of OSC [32] was adopted to calculate three OSC components to remove using labeled samples in batch 1. The tolerance of data variance was set to 90% [32]. The parameters were all tuned by grid search. The improvement in the average accuracy achieved by the two CC-based methods is not large, possibly because the drift in the dataset is complex, and it is hard for CC-based methods to precisely separate the directions of useful information and drift. Significantly better accuracy is achieved by the classifier ensemble method [3]. However, when testing a sample in batch t, it requires the labels of all samples from batches 1 to $t-1$ to train $t-1$ base classifiers. Retrieving labels from many samples is sometimes a demanding job in real-life applications. The same difficulty exists in domain adaptation extreme learning machine (DAELM) [16]. In Table 6, the number after DAELM is the number of labeled samples needed in batch t. A good result was obtained by DAELM at the cost of 30 selected labeled target samples in batch t. Although a higher accuracy was obtained by a variant of DAELM in [16], it is not listed here because up to 50 labeled and all unlabeled target samples were needed to train the model, which was also not recommended by the author. Meanwhile, manifold regularization with combined geodesic flow kernels (ML-comGFK) [15] requires only unlabeled samples from batches 2 to t, which is easier to collect. However, the accuracy of ML-comGFK is not as high as classifier ensemble and DAELM.

For the methods in the last five rows of Table 6, 10 transfer samples were utilized. In fact, because the number of candidate transfer samples between batch 1 and batches 8 and 9 is not sufficient, the actual n_t for the two target batches are 8 and 6, respectively. (Note that the number of training samples is 445, i.e. the number of samples in batch 1.) TCTL + SEMI outperforms DAELM in average accuracy with only one-third of the auxiliary samples needed. Variable standardization also has better accuracy than classifier ensemble. The probable reason is that variable standardization and TCTL make use of transfer samples, which contain mapping information of variables between batch 1 and batch t. Thus, fewer transfer samples are needed than labeled target samples when comparable transfer performance is achieved.

From the results of the multi-device dataset and the long-term drift dataset, we can also find that TCTL is always better than variable standardization. As explained in Section 2, the two methods are related. Variable standardization can be regarded as estimating a matrix that transforms target variables into the source space. TCTL merges the matrix into the objective function in order to reduce the number of coefficients to estimate without limiting the shape of the matrix (diagonal or banded), so it can better

Table 6
Comparison of classification accuracy (%) on the long-term drift dataset.

	Batch 2	3	4	5	6	7	8	9	10	Average
No transfer	88.59	66.96	40.99	54.82	43.22	44.40	31.63	45.74	39.11	50.61
CC-PCA [11]	90.92	40.86	47.20	59.39	56.74	56.71	36.39	45.32	37.72	52.36
OSC [6]	88.10	66.71	54.66	53.81	65.13	63.71	36.05	40.21	40.08	56.50
Ensemble [3]	74.36	87.83	**93.79**	95.43	69.17	69.72	91.84	76.38	65.50	80.45
ML-comGFK [15]	80.25	74.99	78.79	67.41	77.82	71.68	49.96	50.79	53.79	67.28
DAELM (30) [16]	87.98	95.74	85.16	95.99	**94.14**	**83.51**	**86.90**	**100.00**	53.62	87.00
Only SNV	86.09	72.89	65.84	84.77	74.91	62.28	50.00	45.11	43.53	65.05
Var. stdd.	96.06	**98.74**	81.99	95.94	78.70	61.53	76.53	96.60	59.08	82.80
Var. stdd. + SEMI	95.42	97.23	82.61	96.95	79.22	65.46	77.55	97.02	61.25	83.63
TCTL	96.86	94.58	86.34	98.48	85.30	78.74	77.21	88.94	58.47	84.99
TCTL + SEMI	**96.95**	96.85	91.30	**98.98**	86.78	82.51	86.05	83.19	**65.75**	**87.60**

Bold values indicate the best results.

model the variation in distribution and is less prone to overfitting. Compared with device 2 in the multi-device dataset, the improvement of TCTL seems more obvious on device 3, where the variation in distribution is more complex. The author of [16] also suggested that drift compensation at the decision level is better than correcting sensor responses owing to the nonlinear dynamic or chaotic behavior of drift. Another observation is that TCTL can be further improved after combination with the SEMI strategy.

5. Conclusion

A novel algorithm framework called transfer sample based coupled task learning (TCTL) was proposed in this paper to predict e-nose data subject to instrumental variation or sensor drift. The two problems were considered as variation of the data distribution in the feature space, so they could be handled uniformly by TCTL. By incorporating transfer samples into its objective function, TCTL integrates the commonly used variable standardization method into its model learning process. Compared with variable standardization, dealing with distribution variation at the model level can be beneficial, especially when the variation is complex. Experiments on a multi-device dataset and a long-term drift dataset suggested that TCTL outperformed typical variable standardization algorithms. On the long-term drift dataset with large sensor drift, TCTL also outperformed several recent drift correction algorithms with much fewer auxiliary samples needed, which shows its superiority in both precision and ease of use in real-life applications. The standardization error-based model improvement (SEMI) strategy proposed in our previous work was also proved to be effective. It could enhance the performance of TCTL and variable standardization.

TCTL is an extensible framework that can be combined with various classification and regression algorithms, for example, logistic regression and ridge regression, which have been demonstrated in this paper. It is also applicable to other problems with different source and target distribution when transfer samples are available, such as calibration transfer of spectroscopic data [4]. Further study is needed on adaptive parameter selection of the regularization parameters based on the characteristics of data. It is also possible to exploit useful information from unlabeled drifted samples [14,15] because they are easy to acquire.

Acknowledgements

The work is partially supported by the GRF fund from the HKSAR Government, the central fund from Hong Kong Polytechnic University, Shenzhen Science and Technology Project (R2013A059), the NSFC fund (61332011, 61272292, 61271344), Shenzhen Fundamental Research fund (JCYJ20130401152508661, JCYJ20140508160910917), and Key Laboratory of Network Oriented Intelligent Computation, Shenzhen, China.

References

[1] S. Marco, A. Gutiérrez-Gálvez, Signal and data processing for machine olfaction and chemical sensing: a review, IEEE Sens. J. 12 (2012) 3189–3214.
[2] S. Di Carlo, M. Falasconi, Drift correction methods for gas chemical sensors in artificial olfaction systems: techniques and challenges, in: InTech, 2012, pp. 305–326.
[3] A. Vergara, S. Vembu, T. Ayhan, M.A. Ryan, M.L. Homer, R. Huerta, Chemical gas sensor drift compensation using classifier ensembles, Sens. Actuators B: Chem. 166 (2012) 320–329.
[4] R.N. Feudale, N.A. Woody, H. Tan, A.J. Myles, S.D. Brown, J. Ferré, Transfer of multivariate calibration models: a review, Chemometr. Intell. Lab. 64 (2002) 181–192.
[5] Y. Wang, D.J. Veltkamp, B.R. Kowalski, Multivariate instrument standardization, Anal. Chem. 63 (1991) 2750–2756.
[6] S. Wold, H. Antti, F. Lindgren, J. Öhman, Orthogonal signal correction of near-infrared spectra, Chemometr. Intell. Lab. 44 (1998) 175–185.
[7] O. Tomic, H. Ulmer, J.-E. Haugen, Standardization methods for handling instrument related signal shift in gas-sensor array measurement data, Anal. Chim. Acta 472 (2002) 99–111.
[8] L. Zhang, F. Tian, C. Kadri, B. Xiao, H. Li, L. Pan, H. Zhou, On-line sensor calibration transfer among electronic nose instruments for monitoring volatile organic chemicals in indoor air quality, Sens. Actuators: B. Chem. 160 (2011) 899–909.
[9] S. Deshmukh, K. Kamde, A. Jana, S. Korde, R. Bandyopadhyay, R. Sankar, N. Bhattacharyya, R. Pandey, Calibration transfer between electronic nose systems for rapid in situ measurement of pulp and paper industry emissions, Anal. Chim. Acta 841 (2014) 58–67.
[10] K. Yan, D. Zhang, Improving the transfer ability of prediction models for electronic noses, Sens. Actuators B: Chem. 220 (2015) 115–124.
[11] T. Artursson, T. Eklöv, I. Lundström, P. Mårtensson, M. Sjöström, M. Holmberg, Drift correction for gas sensors using multivariate methods, J. Chemometr. 14 (2000) 711–723.
[12] A.-C. Romain, J. Nicolas, Long term stability of metal oxide-based gas sensors for e-nose environmental applications: an overview, Sens. Actuators B: Chem. 146 (2010) 502–506.
[13] M. Padilla, A. Perera, I. Montoliu, A. Chaudry, K. Persaud, S. Marco, Drift compensation of gas sensor array data by orthogonal signal correction, Chemometr. Intell. Lab. 100 (2010) 28–35.
[14] S. De Vito, G. Fattoruso, M. Pardo, F. Tortorella, G. Di Francia, Semi-supervised learning techniques in artificial olfaction: a novel approach to classification problems and drift counteraction, IEEE Sens. J. 12 (2012) 3215–3224.
[15] Q. Liu, X. Li, M. Ye, S.S. Ge, X. Du, Drift compensation for electronic nose by semi-supervised domain adaption, IEEE Sens. J. 14 (2014) 657–665.
[16] L. Zhang, D. Zhang, Domain adaptation extreme learning machines for drift compensation in e-nose systems, IEEE Trans. Instrum. Meas. (2015).
[17] J.H. Kalivas, G.G. Siano, E. Andries, H.C. Goicoechea, Calibration maintenance and transfer using Tikhonov regularization approaches, Appl. Spectrosc. 63 (2009) 800–809.
[18] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.
[19] R. Caruana, Multitask learning, Mach. Learn. 28 (1997) 41–75.
[20] T. Evgeniou, M. Pontil, Regularized multi-task learning, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Seattle, Washington, USA, 2004, pp. 109–117.
[21] Y. Binfeng, J. Haibo, Near-infrared calibration transfer via support vector machine and transfer learning, Anal. Methods 7 (2015) 2714–2725.
[22] K. Yan, D. Zhang, D. Wu, H. Wei, G. Lu, Design of a breath analysis system for diabetes screening and blood glucose level prediction, IEEE Trans. Biomed. Eng. 61 (2014) 2787–2795.
[23] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, New York, 2009.
[24] S. Hosseini-Golgoo, F. Hossein-Babaei, Assessing the diagnostic information in the response patterns of a temperature-modulated tin oxide gas sensor, Meas. Sci. Technol. 22 (2011) 035201.
[25] A. D'Amico, C. Di Natale, R. Paolesse, A. Macagnano, E. Martinelli, G. Pennazza, M. Santonico, M. Bernabei, C. Roscioni, G. Galluccio, R. Bono, E. Finazzi Agrò, S. Rullo, Olfactory systems for medical applications, Sens. Actuators B: Chem. 130 (2008) 458–465.
[26] D. Guo, D. Zhang, N. Li, L. Zhang, J. Yang, A novel breath analysis system based on electronic olfaction, IEEE Trans. Biomed. Eng. 57 (2010) 2753–2763.
[27] R.W. Kennard, L.A. Stone, Computer aided design of experiments, Technometrics 11 (1969) 137–148.
[28] K. Yan, D. Zhang, Feature selection and analysis on correlated gas sensor data with recursive feature elimination, Sens. Actuators B: Chem. 212 (2015) 353–363.
[29] E. Martinelli, C. Falconi, A. D'Amico, C. Di Natale, Feature extraction of chemical sensors in phase space, Sens. Actuators B: Chem. 95 (2003) 132–139.
[30] I. Rodriguez-Lujan, J. Fonollosa, A. Vergara, M. Homer, R. Huerta, On the calibration of sensor arrays for pattern recognition using the minimal number of experiments, Chemometr. Intell. Lab. 130 (2014) 123–134.
[31] A. Vergara, Gas Sensor Array Drift Dataset at Different Concentrations Data Set, 2015 https://archive.ics.uci.edu/ml/datasets/Gas+Sensor+Array+Drift+Dataset+at+Different+Concentrations.
[32] Orthogonal Signal Correction Matlab Code, 2015 http://www.eigenvector.com/MATLAB/OSC.html.

Biographies

Ke Yan received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China. He is currently working toward the Ph.D. degree with the Department of Electronic Engineering, Graduate School at Shenzhen, Tsinghua University, Shenzhen, China. His research interests include machine olfaction and pattern recognition.

David Zhang (F'09) graduated in Computer Science from Peking University. He received his M.Sc. in Computer Science in 1982 and his Ph.D. in 1985 from the Harbin Institute of Technology (HIT). From 1986 to 1988 he was a postdoctoral

fellow at Tsinghua University and then an associate professor at the Academia Sinica, Beijing. In 1994 he received his second Ph.D. in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. Currently, he is a Chair professor at the Hong Kong Polytechnic University where he is the Founding Director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. He also serves as Visiting Chair professor in Tsinghua University, and Adjunct professor in Peking University, Shanghai Jiao Tong University, HIT, and the University of Waterloo. He is the Founder and Editor-in-Chief, International Journal of Image and Graphics (IJIG); Book Editor, Springer International Series on Biometrics (KISB); Organizer, the International Conference on Biometrics Authentication (ICBA); Associate Editor of more than ten international journals including IEEE Transactions and Pattern Recognition; and the author of more than 10 books and 200 journal papers. Professor Zhang is a Croucher senior research fellow, Distinguished Speaker of the IEEE Computer Society, and a fellow of both IEEE and IAPR.