Proceedings of the 7th International Symp. On Olfaction and Electronic Nose, Brighton, UK, July 20-24, 2000

Drift Reduction For Metal-Oxide Sensor Arrays Using Canonical Correlation Regression And Partial Least Squares

R Gutierrez-Osuna

Computer Science Department, Wright State University, Dayton, OH 45435, USA

Abstract. The transient response of metal-oxide sensors exposed to mild odours can be oftentimes highly correlated with the behaviour of the array during the preceding wash and reference cycles. Since wash/reference gases are virtually constant overtime, variations in their transient response can be used to estimate the amount of sensor drift present in each experiment. We perform canonical correlation analysis and partial least squares to find a subset of "latent variables" that summarize the linear dependencies between odour and wash/reference responses. Ordinary least squares regression is then used to subtract these "latent variables" from the odour response. Experimental results on an odour database of four cooking spices, collected on a 10-sensor array over a period of three months, show significant improvements in predictive accuracy.

1. Introduction

Besides selectivity and sensitivity constraints, sensor drift constitutes the main limitation of current gas sensor array instruments [1]. In order to improve baseline stability, it is customary to expose the sensor array to a sequence of wash and reference gases prior to sampling a target odour. Since wash/reference (W/R) gases can be assumed to remain stable over time, variations in sensor response during these stages may be utilized, at no extra cost, as measures of the drift associated with each experiment and, therefore, compensated for in software.

The instrument employed in this study is an array of ten commercial metal-oxide sensors. During operation, the array is initially washed for ten seconds with room air bubbled through a two-per-cent n-butanol dilution in order to flush residues from previous odour samples. This wash stage is followed by a three-minute reference stage with charcoal-filtered air, which provides a baseline for the subsequent ninety-second odour stage, in which the array is finally exposed to the target odour. Figure 1 illustrates a typical sensor transient response obtained with this procedure.

To preserve information from the sensor dynamics, we apply the windowed time slicing (WTS) technique [2, 3] to compress each exponential-shaped transient down to four weighted integrals, as shown in the lower portion of Figure 1. With ten sensors, this compression technique yields 40-dimensional feature vectors x_W, x_R and y for the wash, reference and odour transients, respectively. x_W and x_R are subsequently combined to form a regression vector x with 80 dimensions.

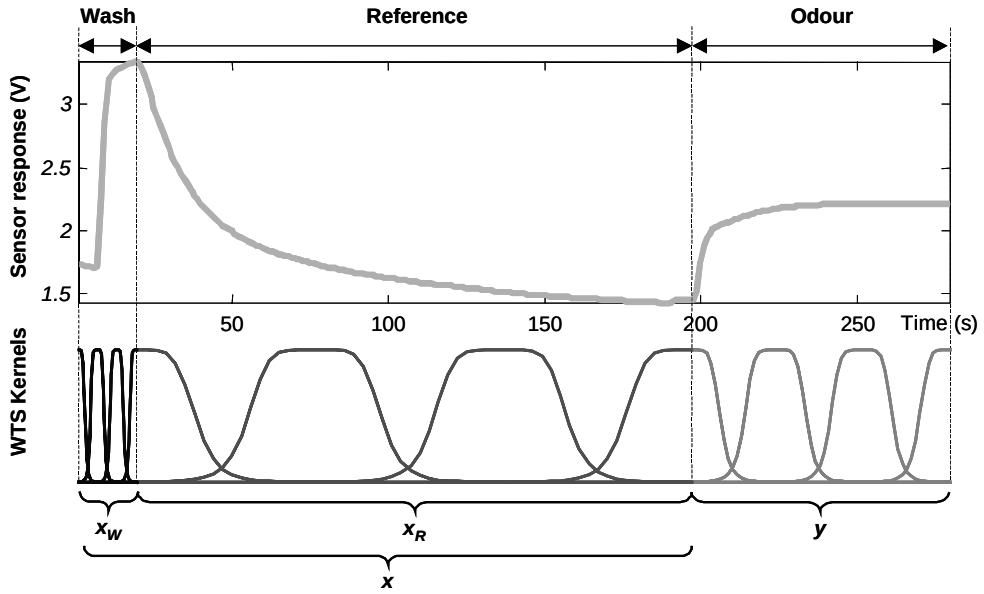

Figure 1. Extraction of window-time-slicing features from the wash, reference and odour stages

2. Drift reduction approach

Whereas variance in the W/R vector x is primarily due to drift, variance in y is a result of the combined effect of drift and the odour-specific interaction with the sensing material. Therefore, we seek to remove variance in y that can be explained by (correlated with) x while preserving odour-related variance. We propose a drift reduction algorithm that consists of three steps:

1. Find linear projections $\tilde{x} = Ax$ and $\tilde{y} = By$ that are maximally correlated:
$$\{A, B\} = \arg\max[\rho(Ax, By)] \quad (1)$$
Vectors $\tilde{x}$ and $\tilde{y}$ are low-dimensional projections that summarize the linear dependencies between x and y.
2. Find a regression model $y_{pred} = W_{OLS}\tilde{y}$ using Ordinary Least Squares (OLS):
$$W_{OLS} = \arg\min\left|y - W\tilde{y}\right|^2 \quad (2)$$
The OLS prediction vector y_{pred} contains the variance in the odour vector y that can be explained by $\tilde{y}$ and, indirectly, by the W/R vector x.
3. Deflate y and use the residual z as a drift-reduced odour vector for classification purposes:
$$z = y - y_{pred} = y - W_{OLS}By \quad (3)$$

To find the projection matrices A and B for the first step of the algorithm we employ Canonical Correlation Analysis (CCA) and Partial Least Squares (PLS.) CCA is a multivariate statistical technique closely related to Principal Components Analysis (PCA.) Whereas PCA finds the directions of maximum variance of each separate x- or y-space, CCA identifies directions where x and y co-vary. It can be shown [4] that the columns of the projection matrices A and B are the eigenvectors of:

$$\begin{aligned}\left(\Sigma_{xx}^{-1}\Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}-\lambda I\right)a=0\\ \left(\Sigma_{yy}^{-1}\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}-\lambda I\right)b=0\end{aligned} \tag{4}$$

where $\{\Sigma_{xx},\Sigma_{yy}\}$ are the "within-space" covariance matrices and $\{\Sigma_{xy},\Sigma_{yx}\}$ are the "between-space" covariance matrices. The first pair of "latent variables" (canonical variates) $\tilde{x}_1 = a_1 x$ and $\tilde{y}_1 = b_1 y$ corresponds to the eigenvectors a_1 and b_1 associated with the largest eigenvalue of (4). This eigenvalue λ_1, identical for both equations, is the squared canonical correlation coefficient between $\tilde{x}_1$ and $\tilde{y}_1$. Subsequent canonical variates are associated with decreasing eigenvalues and are uncorrelated with previous variates. CCA will find as many pairs of canonical variates as the minimum of P and Q, the dimensionality of *x* and *y*, respectively.

Alternatively, one may use PLS to find the canonical variates (scores) $\tilde{x}_k$ and $\tilde{y}_k$. It must be noted that, whereas CCA finds all the eigenvector pairs (loadings) a_k and b_k simultaneously, PLS operates sequentially, extracting one pair of loadings/scores at a time using a NIPALS algorithm [5]. Deflation of *x* and *y* in PLS is also performed sequentially, after the extraction of each pair of loadings/scores. Furthermore, CCA maximizes the correlation coefficient between each pair ($\tilde{x}_k$, $\tilde{y}_k$) while PLS takes into consideration both correlation and variance in *x* and *y* [6]. Both CCA and PLS versions of the drift-reduction algorithm have been implemented and are reported in this article.

3. Preliminary analysis

The proposed drift-reduction method is evaluated on an odour database of four spices (ginger, red pepper, cumin and cloves) containing 374 examples collected during 24 different days over a period of three months. The array consists of ten metal-oxide sensors from Capteur® (models AA14, AA20, G5, AA25, G7, LG9 and LG10) and Figaro® (models 2620, 2610 and 2611.)

We perform a preliminary analysis by feeding the entire database to the drift-reduction algorithm. The resulting correlation coefficients for each pair of latent variables $\rho(\tilde{x}_k,\tilde{y}_k)$ are shown in Figure 2. This result indicates that PLS is able to reduce correlations between *x* and *y* with fewer components than CCA, a reasonable result since PLS performs deflation sequentially, after extraction of each "latent variable." The figure also suggests that there are 10 major canonical variates (interestingly, the same as the number of sensors), although Bartlett's sequential test [4] indicates that there are 21 statistically significant variates.

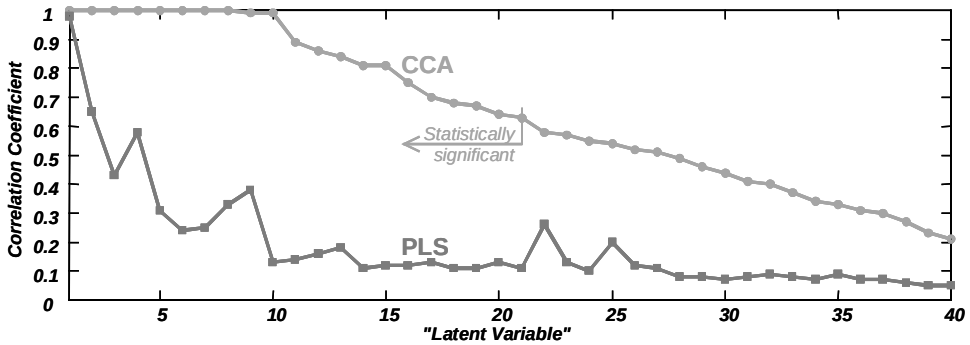

Figure 2. Correlation coefficient $\rho(\tilde{x}_k,\tilde{y}_k)$ for each pair of "latent variables"

To illustrate the performance of the algorithm, we select the first 10 "latent variables" and plot the trajectory of the four WTS features over time before and after drift reduction. Figure 3 shows these trajectories for the AA20 sensor. Examples are sorted by class –notice the four distinctive blocks— and, within each class, by date. Drift-reduced trajectories have been shifted vertically for visualization purposes. Both CCA and PLS versions of the algorithm significantly reduce the linear component of the drift. In other words, the algorithm is able to reduce drift-related variance (noise) while preserving odour-related variance (information.) It is also interesting to notice that the first WTS window is deflated almost entirely, a reasonable result since it is highly correlated with the last WTS window of the reference transient.

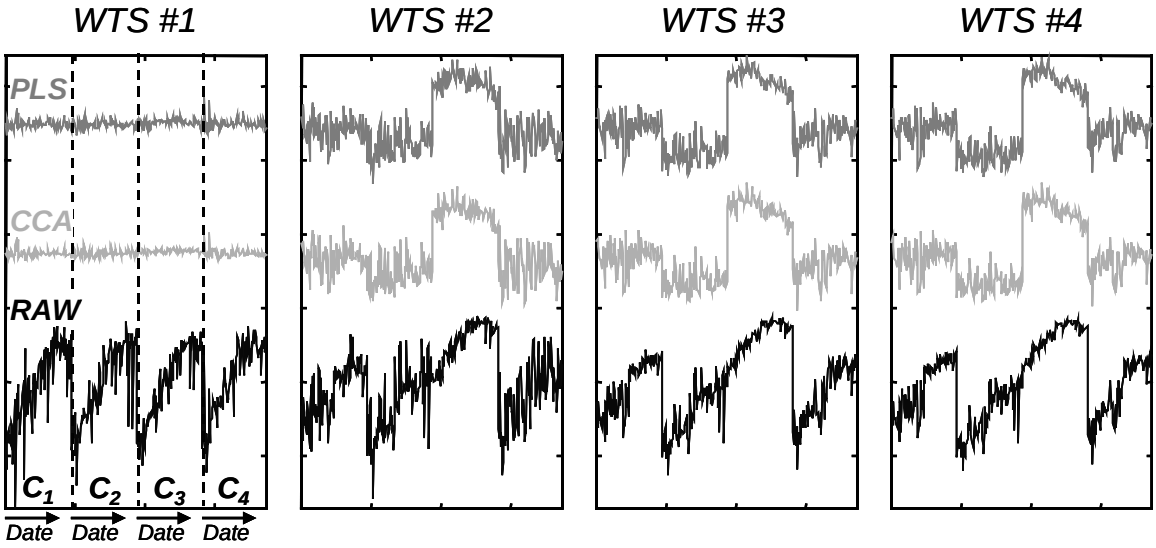

Figure 3. Trajectory of WST features before and after drift reduction

4. Analysis of predictive accuracy

A more interesting and formal assessment of performance is predictive accuracy, that is, the ability of a subsequent pattern classifier to correctly classify previously unseen test data. In the context of drift-reduction, predictive accuracy should be measured by using test examples that were collected on different days than those used for training, as illustrated in Figure 4. Formally, given a database collected over M days, we seek to classify data from day k using data from days i through j, with $i<j<k$. We denote by W (width) the number of days used for training ($W=j-i+1$) and D (distance) the number of days between the training and test days ($D=k-j$.) These two parameters, also shown in Figure 4, can significantly affect the predictive accuracy of a classifier:

- With increasing values of W, the training set incorporates data from more days, which may allow a classifier to automatically average out the drift component.
- With increasing values of D, the effects of drift accumulate on the test set, which may result in lower predictive accuracy.

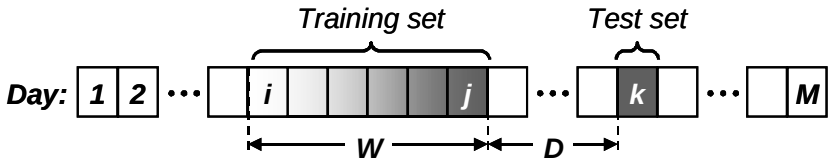

Figure 4. Illustration of W and D for predictive accuracy measurements

To analyze the effect on predictive accuracy of these two parameters along with the proposed drift-reduction technique, we use a k-Nearest-Neighbor (kNN) classifier [7]. This voting rule classifies an unlabeled example by choosing the majority class among the *k* closest examples in the training set. Our implementation uses the Euclidean metric, with the number of neighbors being set to one-half the average number of examples per class in the training set. To limit the "curse of dimensionality," the feature vectors *y* and *z* are pre-processed with PCA, and the largest eigenvalues containing 99% of the variance are selected. PCA eigenvectors are computed from the training set and then used to project test data. Depending on the number of training examples (a function of W,) this results in 6 to 10 principal components being retained for *y* and 3 to 6 principal components for *z*. As shown in the previous section, the drift-reduction algorithm significantly deflates the first WST feature, which explains why fewer principal components are required to total the same percentage of the variance.

It must be noted that this new analysis is more challenging than the preliminary validation shown in Figure 3, in which the entire dataset (days 1 through 24) was passed to the algorithm. In the more realistic scenario described in this section, only past training data (days *i* through *j*) and current test data (day *k*) are used to perform drift reduction. Intuitively, with fewer training data, fewer "latent variables" should be used to deflate the odour vector *y*. In fact, using ten "latent variables" yields only modest improvements in predictive accuracy compared to raw data. After some experimentation, we determined that three was a suitable number of "latent variables." We also noticed that in order to eliminate the linear component of the drift (see Figure 3) it was necessary to incorporate timing information into the drift reduction algorithm, especially for large values of D. For this reason, we decided to augment the regression vector *x* with the time stamp of each sample. This had not been necessary in the previous section, where the entire sequence of days was presented to the drift-reduction algorithm. With the selection of the first three "latent variables" and the addition of time stamps, we were able to obtain significant improvements in predictive accuracy, which are reported below.

The effect of distance on predictive accuracy is shown in Figure 5 for different values of the width parameter. As expected, the performance of the raw data decreases considerably with increasing distances (older training data.) After the application of CCA- or PLS-based drift reduction, predictive accuracy does not decrease with distance, an indication that the algorithm is able to compensate for drift. In fact, CCA and PLS present a slight increase in predictive accuracy with distance. This counter-intuitive result could be attributed to the fact that both training data and unlabeled test data are used to estimate the drift model $y_{pred}=W_{OLS}By$, making the slope $W_{OLS}B$ more accurate with increasing distances as a result of the time stamps. This conjecture, however, deserves further attention.

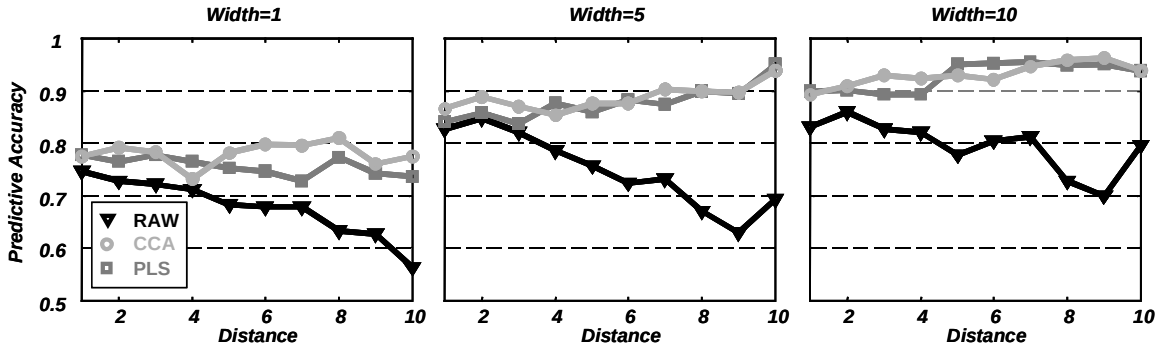

Figure 5. Effect of Distance on predictive accuracy for different Widths

The effect of width is shown in Figure 6. The three datasets yield higher predictive accuracy with increasing values of width (larger training sets.) Higher slopes in the CCA and PLS curves indicate that the drift-reduction technique allows the subsequent pattern classifier to make better use of additional training data up to a saturation point of 95% predictive accuracy, which can be clearly observed in the rightmost plot of Figure 6. No significant differences in predictive accuracy between CCA and PLS can be found in these two figures.

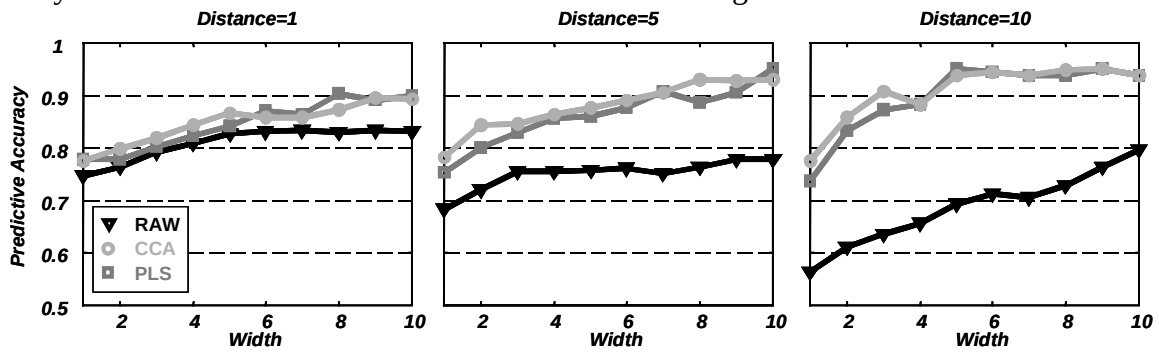

Figure 6. The effect of Width on predictive accuracy for different Distances

5. Conclusions and future work

We have proposed a drift-reduction algorithm that takes advantage of drift information contained in the pre-wash/reference stages of a typical sampling cycle. The algorithm has been shown to reduce drift-related variance in the sensors and, therefore, enhance class discrimination and predictive accuracy of a subsequent pattern classifier.

Our experience has shown that the predictive-accuracy performance of the algorithm is sensitive to the number of "latent variables," which has been manually selected in the current implementation. Automated selection of "latent variables" by cross-validation techniques constitutes the next natural step of this work. Further improvements could be obtained by augmenting the regression vector x with temperature, humidity and mass flow information, as well as transient information from appropriate calibration mixtures that could be added to the sampling cycle. Finally, the proposed algorithm is limited to linear dependencies between W/R and odour stages. Non-linear extensions of CCA [8] and PLS [9] will be required for situations where these relationships are markedly non-linear.

6. Acknowledgements

This research was supported by awards WSU/OBR 0409 and NSF/CAREER 9984426. We would like to acknowledge Alex Perera (Universitat de Barcelona) for his valuable comments and suggestions on earlier drafts of this document.

7. References

[1] Gardner J W and Bartlett P N 1999 *Electronic Noses: Principles and Applications* (Oxford: Oxford Univ. Press)
[2] Kermani B G 1996 *On Using Artificial Neural Networks and Genetic Algorithms to Optimize Performance of an Electronic Nose* Ph.D. Dissertation, North Carolina State University.
[3] Gutierrez-Osuna R and Nagle H T 1999 *IEEE Trans. Syst. Man Cybern. B* 29(5), 626-632.
[4] Dillon W R and Goldstein M 1984 Multivariate Analysis: Methods and applications (New York: Wiley)
[5] Geladi P and Kowalski B R 1986 Analytica Chimica Acta 185 1-17.
[6] Burnham A J et al. 1999 J Chemometrics 13 49-65
[7] Duda R O and Hart P E 1973 *Pattern classification and Scene Analysis* (New York: Wiley)
[8] Lai P L and Fyfe C 1999 Neural Networks 12(10), 1391-1397.
[9] Malthouse E C et al. 1997 Computers chem. Engng 21(8) 875-890