On the orthogonal distance to class subspaces for high-dimensional data classification

Rui Zhu[a], Jing-Hao Xue[a,*]

[a]*Department of Statistical Science, University College London, London WC1E 6BT, UK*

Abstract

The orthogonal distance from an instance to the subspace of a class is a key metric for pattern classification by the class subspace-based methods. There is a close relationship between the orthogonal distance and the residual standard deviation of a test instance from the class subspace. In this paper, we shall show that an established and widely-used relationship, between the residual standard deviation and the sum of squares of the residual PC scores, is not precise, and thus can lead to incorrect results, for the inference of high-dimensional data which nowadays are common in practice.

Keywords: Classification, high-dimensional data, orthogonal distance, principal component analysis (PCA), soft independent modelling of class analogy (SIMCA).

1. Introduction

In class subspace-based classification methods, a subspace is first learned in the training phase for each class separately from its training data. Then in

*Corresponding author. Tel.: +44-20-7679-1863; Fax: +44-20-3108-3105

Email addresses: r.zhu.12@ucl.ac.uk (Rui Zhu), jinghao.xue@ucl.ac.uk (Jing-Hao Xue)

the test phase, these learned class subspaces are utilised to predict the label of a new test instance, by comparing the distances from the test instance to the class subspaces, in terms of certain distance metrics. For example, in a widely-used classifier for spectral data called soft independent modelling of class analogy (SIMCA) [28], principal component (PC) subspaces are learned for individual classes. Similar to SIMCA, another popular PCA-based classification approach has been extensively adopted in process control in engineering, such as fault detection and diagnosis [20, 16, 15, 25]. Besides classification methods, some clustering methods also aim to seek low-dimensional subspaces for better clustering results [13, 23, 22].

In the above two classification approaches, associated with the PC subspaces, two distance metrics (or statistics) are often adopted to achieve pattern classification [3, 17, 18, 20, 16, 15, 25, 29]: 1) the orthogonal distance (OD), also known as the Q-statistic or the squared prediction error, i.e. the squared orthogonal Euclidean distance from a test instance to a PC subspace; and 2) the score distance (SD), also known as the Hotelling's T^2 statistic, i.e. the squared Mahalanobis distance from the projection of a test instance to the centre of a PC subspace [17]. The distributions of OD and SD have also been studied extensively, in order to find a proper acceptance area for classification; recent work includes [17], [18], [19], [30] and [21]. Also in recent years, a linear combination of these two distances is often used to classify a test instance: the test instance is assigned to the class with the minimum value of the linear combination [3].

There is a close relationship between the OD (from a test instance to a class subspace) and the residual standard deviation of the test instance to

the class subspace. Moreover, Maesschalck et al. [9] show that the residual standard deviation based on the residual matrix can be equivalently calculated from using the residual PC scores based on the PC score matrix. This work has been cited over a hundred times, including methodological developments [4, 10, 8], reviews [24, 14] and applications [5, 2, 6, 27, 7]. The recent work studying the distributions of OD and SD [17, 18, 19] also adopted the formulae in [9] following [10].

However in this paper, we shall point out that the relationship presented in [9], between the residual standard deviation and the sum of squares of the residual PC scores, is *not* precise for the inference of high-dimensional data.

To distinguish the training and test scenarios, we shall establish the notation of two ODs, respectively, as follows.

1. The OD $v^{k,l}$ from the *training* instance l to the subspace of class k that was learned from all training instances. It is closely related to the residual standard deviation $s^{k,0}$ of class k, which will be defined in Section 2.1.
2. The OD $v^{k,new}$ from the new *test* instance to the subspace of class k. It is closely related to the residual standard deviation $s^{k,new}$ of the new test instance to class k, which will be defined in Section 2.2.

In short, the difference between $v^{k,l}$ and $v^{k,new}$ is that $v^{k,l}$ is the OD for the training instance while $v^{k,new}$ is the OD for the test instance.

The contributions of this paper are as follows. First, although Maesschalck et al. [9] establish formulae for $s^{k,0}$ and $s^{k,new}$ using the residual PC scores, we shall show that their formula for $s^{k,new}$ is only precise when the training data of class k have more instances than predictor features, i.e. when

the number of instances (denoted by n_k) is larger than the number of features (denoted by p). In other words, we shall show that, when the training data of class k are high-dimensional (i.e. $n_k \leq p$, also called "large p, small n" in the statistical literature), the calculation of $s^{k,new}$ in [9] is not precise.

Second, because of the above results, we shall point out that, for high-dimensional data, although the OD $v^{k,l}$ can be accurately calculated by following the (precise) formula of the residual standard deviation $s^{k,0}$ in [9], the OD $v^{k,new}$ cannot be accurately calculated by following the (imprecise) formulae of the residual standard deviation $s^{k,new}$ in [9]. Consequently, inference results of the studies that calculated the ODs for high-dimensional data using the formulae in [9] can be imprecise.

Because nowadays high-dimensional data are commonly present in pattern-recognition tasks, it is of great interest to practitioners to point out the imprecise calculation of the ODs for high-dimensional data if we follow the formulae in [9], as well as to suggest that the formulae in [28] should be adopted in this "large p, small n" paradigm.

2. The calculations of OD in [9]

The following calculations are all for class k. The subscripts p, q and r denote the number of columns in matrices $\boldsymbol{U}$, $\boldsymbol{D}$, $\boldsymbol{V}$ and $\boldsymbol{T}$; for example, $\boldsymbol{V}_p$ indicates that there are p columns in matrix $\boldsymbol{V}_p$ of class k.

2.1. The training phase of class k

Suppose $\boldsymbol{X} \in \mathbb{R}^{n_k \times p}$ is the training set of class k, in which there are n_k training instances (or say training samples) and each instance is represented by a p-dimensional data vector. To build the PC subspace of class k, we

apply the reduced singular value decomposition (SVD) to the column-centred training set $\boldsymbol{X}_{(c)}$:

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_q \boldsymbol{D}_q (\boldsymbol{V}_q)^T \ , \tag{1}$$

where $\boldsymbol{U}_q \in \mathbb{R}^{n_k \times q}$ and $\boldsymbol{V}_q \in \mathbb{R}^{p \times q}$ are the two matrices containing left and right singular vectors as columns, respectively, and $\boldsymbol{D}_q \in \mathbb{R}^{q \times q}$ is a diagonal matrix with singular values $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \geq 0\}$. The parameter $q \leq \min(p, n_k - 1)$ is the rank of $\boldsymbol{X}_{(c)}$.

In PCA, the rows of $\boldsymbol{T}_q = \boldsymbol{U}_q \boldsymbol{D}_q \in \mathbb{R}^{n_k \times q}$ are known as PC scores and the columns of $\boldsymbol{V}_q$ are known as PCs. Suppose the first r $(r \leq q)$ PCs are selected to build the PC subspace for class k, then

$$\boldsymbol{X}_{(c)} = \boldsymbol{T}_r (\boldsymbol{V}_r)^T + \boldsymbol{E} \ , \tag{2}$$

where $\boldsymbol{T}_r \in \mathbb{R}^{n_k \times r}$; $\boldsymbol{V}_r \in \mathbb{R}^{p \times r}$; and $\boldsymbol{E} \in \mathbb{R}^{n_k \times p}$ is the training residual matrix of class k.

In [9], the residual standard deviation of class k is expressed in two forms:

$$s^{k,0} = \sqrt{\frac{1}{\mathrm{DoF}^{k,0}} \sum_{l=1}^{n_k} \sum_{j=1}^{p} (e_{lj})^2} = \sqrt{\frac{1}{\mathrm{DoF}^{k,0}} \sum_{l=1}^{n_k} \sum_{i=r+1}^{q} (t_{li})^2} \tag{3}$$

where $\mathrm{DoF}^{k,0} = (q - r)(n_k - r - 1)$, e_{lj} is the (l, j)-entry of residual matrix $\boldsymbol{E}$ representing the residual of the lth instance for the jth variable, and t_{li} is the (l, i)-entry of score matrix $\boldsymbol{T}_q$ representing the score of the lth instance for the ith PC.

The OD from the lth training instance to the subspace of class k, $v^{k,l}$, is

originally defined as $\sum_{j=1}^{p}(e_{lj})^2$. Thus $\sum_{l=1}^{n_k} v^{k,l}$ is proportional to $(s^{k,0})^2$,

$$\sum_{l=1}^{n_k} v^{k,l} = (s^{k,0})^2(q-r)(n_k-r-1). \tag{4}$$

In [9], it follows from (3) that $\sum_{l=1}^{n_k} v^{k,l}$ can be calculated as

$$\sum_{l=1}^{n_k} v^{k,l} = \sum_{l=1}^{n_k}\sum_{i=r+1}^{q} (t_{li})^2 \ . \tag{5}$$

2.2. The test phase for class k

In the test (prediction) phase, to decide whether a new instance $\boldsymbol{x}^{new}$ belongs to class k or not, $\boldsymbol{x}^{new}$ is first centred by using the means of the variables of the training data $\boldsymbol{X}$ of class k, and the result is denoted by $\boldsymbol{x}_{(c)}^{k,new}$. Then projecting $\boldsymbol{x}_{(c)}^{k,new}$ to the PC subspace of class k with the selected r PCs, we can obtain

$$\boldsymbol{x}_{(c)}^{k,new} = \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T + \boldsymbol{e}^{k,new} \ , \tag{6}$$

where $\boldsymbol{t}_r^{k,new} \in \mathbb{R}^{1\times r}$ and $\boldsymbol{e}^{k,new} \in \mathbb{R}^{1\times p}$ are two vectors of the PC score and the residual, respectively, of the new instance when it is fitted to the subspace of class k.

In [9], the residual standard deviation of the new instance is also expressed in two forms:

$$s^{k,new} = \sqrt{\frac{1}{\mathrm{DoF}^{k,new}}\sum_{j=1}^{p}(e_j^{k,new})^2} = \sqrt{\frac{1}{\mathrm{DoF}^{k,new}}\sum_{i=r+1}^{q}(t_i^{k,new})^2}, \tag{7}$$

where $\mathrm{DoF}^{k,new} = (q-r)$, $e_j^{k,new}$ and $t_i^{k,new}$ denote the jth element of the

residual vector $\boldsymbol{e}^{k,new}$ and the ith element of the PC score vector $\boldsymbol{t}_r^{k,new}$, respectively.

The OD from the new instance to the subspace of class k, $v^{k,new}$, is originally defined as $\sum_{j=1}^{p}(e_j^{k,new})^2$. Thus $v^{k,new}$ is proportional to $(s^{k,new})^2$,

$$v^{k,new} = (s^{k,new})^2(q-r). \tag{8}$$

In [9], it follows from (7) that $v^{k,new}$ can be written as

$$v^{k,new} = \sum_{i=r+1}^{q}(t_i^{k,new})^2 \ . \tag{9}$$

To determine the class of $\boldsymbol{x}^{new}$, the residual standard deviation $s^{k,new}$ of $\boldsymbol{x}^{new}$ is compared to the residual standard deviation $s^{k,0}$ of the training instances of class k [9]. The F-test statistic used in [9] to determine whether the two residual variances are significantly different is expressed as

$$F^{k,new} = \frac{(s^{k,new})^2}{(s^{k,0})^2} = \frac{\sum_{i=r+1}^{q}(t_i^{k,new})^2 \ (n_k - r - 1)}{\sum_{l=1}^{n_k}\sum_{i=r+1}^{q}(t_{li})^2} \ . \tag{10}$$

3. Discussion of $v^{k,l}$ and $v^{k,new}$

The calculations for $v^{k,0}$ and $v^{k,new}$ in [9] use formulae (5) and (9), respectively. We shall show that, while formula (5) is correct for both the cases of $n_k > p$ and $n_k \leq p$, formula (9) is only valid when $n_k > p$.

3.1. $v^{k,l}$

The OD $v^{k,l}$ is originally defined on the basis of the residual matrix $\boldsymbol{E}$. The calculation of $v^{k,l}$ in (5), which was defined in [9], is on the basis of the

PC score matrix $\boldsymbol{T}_r$. This is due to the relationship that

$$\sum_{l=1}^{n_k}\sum_{j=1}^{p}(e_{lj})^2 = \sum_{l=1}^{n_k}\sum_{i=r+1}^{q}(t_{li})^2 \ . \tag{11}$$

This relationship is true for both the cases of $n_k > p$ and $n_k \leq p$, as we shall show in the following two subsections, respectively.

3.1.1. $n_k > p$

When $n_k > p$, we have $q = p$ (assume that no feature is a linear combination of others), and thus $\boldsymbol{V}_q \in \mathbb{R}^{p\times p}$ is a square matrix. It follows that $\boldsymbol{V}_q(\boldsymbol{V}_q)^T = (\boldsymbol{V}_q)^T\boldsymbol{V}_q = \boldsymbol{I}_p$.

Let $\boldsymbol{x}^l_{(c)} \in \mathbb{R}^{1\times p}$ denote the l-th training instance in class k, i.e. the l-th row of $\boldsymbol{X}_{(c)}$. For every $\boldsymbol{x}^l_{(c)}$ $(l = 1, \ldots, n_k)$, we have $\boldsymbol{x}^l_{(c)} = \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$ and

$$\begin{aligned}
\sum_{j=1}^{p}(e_{lj})^2 &= ||\boldsymbol{x}^l_{(c)} - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_r(\boldsymbol{V}_r)^T||_2^2 \\
&= ||\boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_r(\boldsymbol{V}_r)^T||_2^2 \\
&= ||\boldsymbol{t}^l_q(\boldsymbol{V}_q)^T - \boldsymbol{t}^l_r(\boldsymbol{V}_r)^T||_2^2 \\
&= \sum_{i=r+1}^{q}(t_{li})^2 \ ,
\end{aligned} \tag{12}$$

where $||\cdot||_2$ denotes the Euclidean norm of a vector, and $\boldsymbol{t}^l_q$ and $\boldsymbol{t}^l_r$ are the lth row of $\boldsymbol{T}_q$ and $\boldsymbol{T}_r$, respectively. Therefore (11) and thus (5) are correct when $n_k > p$.

3.1.2. $n_k \leq p$

When $n_k \leq p$, we have $q = \text{rank}(\boldsymbol{X}_{(c)}) \leq n_k - 1 < p$, and thus $\boldsymbol{V}_q \in \mathbb{R}^{p \times q}$ is not square. It follows that $(\boldsymbol{V}_q)^T \boldsymbol{V}_q = \boldsymbol{I}_q$ but $\boldsymbol{V}_q (\boldsymbol{V}_q)^T \neq \boldsymbol{I}_p$.

Suppose we apply the full SVD to $\boldsymbol{X}_{(c)}$:

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_{n_k} \hat{\boldsymbol{D}}_p (\boldsymbol{V}_p)^T \ , \tag{13}$$

where $\boldsymbol{U}_{n_k} \in \mathbb{R}^{n_k \times n_k}$ and $\boldsymbol{V}_p \in \mathbb{R}^{p \times p}$ denote the two matrices containing n_k left and p right singular vectors as columns, respectively, and $\hat{\boldsymbol{D}}_p \in \mathbb{R}^{n_k \times p}$ is a matrix with singular values $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n_k - 1} \geq \lambda_{n_k} = 0\}$ on the main diagonal.

To make the explanation more clear, we expand $\hat{\boldsymbol{D}}_p \in \mathbb{R}^{n_k \times p}$ to a square matrix $\boldsymbol{D}_p \in \mathbb{R}^{p \times p}$ by adding zeros because the singular values associated with the last $(p - q)$ PCs are zeros when $n_k \leq p$. Matrix $\boldsymbol{U}_{n_k} \in \mathbb{R}^{n_k \times n_k}$ is also expanded to $\boldsymbol{U}_p \in \mathbb{R}^{n_k \times p}$ using $(p - n_k)$ unit-length column vectors that are randomly calculated to be orthogonal to the previous column vectors. Thus we have

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_{n_k} \hat{\boldsymbol{D}}_p (\boldsymbol{V}_p)^T \ = \boldsymbol{U}_p \boldsymbol{D}_p (\boldsymbol{V}_p)^T, \tag{14}$$

where $\boldsymbol{U}_p \in \mathbb{R}^{n_k \times p}$ and $\boldsymbol{V}_p \in \mathbb{R}^{p \times p}$ denote the matrices containing p left and p right singular vectors, respectively, and $\boldsymbol{D}_p \in \mathbb{R}^{p \times p}$ is a diagonal matrix with singular values $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q \geq \lambda_{q+1} = \cdots = \lambda_p = 0\}$. Since $\boldsymbol{V}_p \in \mathbb{R}^{p \times p}$ is square, we have $\boldsymbol{V}_p (\boldsymbol{V}_p)^T = (\boldsymbol{V}_p)^T \boldsymbol{V}_p = \boldsymbol{I}_p$.

Let $\boldsymbol{T}_p = \boldsymbol{U}_p \boldsymbol{D}_p \in \mathbb{R}^{n_k \times p}$ denote the PC scores. Let t_{li} denote the (l, i)-entry of score matrix $\boldsymbol{T}_p$ representing the score of the lth instance for the ith PC.

Let $\boldsymbol{m}^l$ denote the residual from using the first q PCs to reconstruct $\boldsymbol{x}^l_{(c)}$: $\boldsymbol{m}^l = \boldsymbol{x}^l_{(c)} - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$. We calculate the sum of squares of the residuals in $\boldsymbol{m}^l$ for the l-th instance:

$$\begin{aligned} ||\boldsymbol{m}^l||_2^2 &= ||\boldsymbol{x}^l_{(c)} - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{x}^l_{(c)}\boldsymbol{V}_p(\boldsymbol{V}_p)^T - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{t}^l_p(\boldsymbol{V}_p)^T - \boldsymbol{t}^l_q(\boldsymbol{V}_q)^T||_2^2 \ . \end{aligned} \tag{15}$$

The sum of $||\boldsymbol{m}^l||_2^2$ for all n_k training instances is

$$\sum_{l=1}^{n_k} ||\boldsymbol{m}^l||_2^2 = \sum_{l=1}^{n_k}\sum_{i=q+1}^{p} (t_{li})^2 = \sum_{i=q+1}^{p} (\lambda_i)^2 \ . \tag{16}$$

The second equation in (16) can be shown as follows. $\boldsymbol{X}_{(c)} = \boldsymbol{U}_p\boldsymbol{D}_p(\boldsymbol{V}_p)^T \Rightarrow$ $(\boldsymbol{U}_p)^T\boldsymbol{X}_{(c)}\boldsymbol{V}_p = \boldsymbol{D}_p \Rightarrow (\boldsymbol{U}_p)^T\boldsymbol{T}_p = \boldsymbol{D}_p$. For the ith singular value λ_i in $\boldsymbol{D}_p$, we have $(\lambda_i)^2 = (\boldsymbol{u}_i^T\boldsymbol{t}_i)^2 = \boldsymbol{t}_i^T\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{t}_i = \boldsymbol{t}_i^T\boldsymbol{t}_i = \sum_{l=1}^{n_k}(t_{li})^2$, where $\boldsymbol{u}_i$ and $\boldsymbol{t}_i$ are the ith columns of $\boldsymbol{U}_p$ and $\boldsymbol{T}_p$, respectively.

Since the last $(p-q)$ singular values are zeros, $\sum_{l=1}^{n_k} ||\boldsymbol{m}^l||_2^2 = 0$. Because each term in the sum $\sum_{l=1}^{n_k} ||\boldsymbol{m}^l||_2^2$ is nonnegative, $||\boldsymbol{m}^l||_2^2 = 0$ for all l ($l = 1, \ldots, n_k$). Thus we have $\boldsymbol{x}^l_{(c)} = \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$, which means that the first q PCs can perfectly reconstruct the training instances in class k. Using the same proof as in (12), we can show that (11) and thus (5) are also true for $n_k \leq p$.

Therefore, $v^{k,l}$ can be correctly calculated by using (5) for both the cases of $n_k > p$ and $n_k \leq p$.

3.2. $v^{k,new}$

The OD $v^{k,new}$ is originally defined in terms of the residual vector $\boldsymbol{e}^{k,new}$ [28], while following [9] $v^{k,new}$ is formulated in (9) by using the PC score $\boldsymbol{t}_r^{k,new}$ of the new sample. We shall show that the formula (9) is valid when $n_k > p$ but not valid when $n_k \leq p$, in the following two subsections, respectively.

3.2.1. $n_k > p$

When $n_k > p$, we have $q = p$, and thus $\boldsymbol{V}_q \in \mathbb{R}^{p\times p}$ is a square matrix. As before, $\boldsymbol{V}_q(\boldsymbol{V}_q)^T = (\boldsymbol{V}_q)^T\boldsymbol{V}_q = \boldsymbol{I}_p$. Since $\boldsymbol{x}_{(c)}^{k,new} = \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$, we have

$$\sum_{j=1}^{p}(e_j^{k,new})^2 = \sum_{i=r+1}^{q}(t_i^{k,new})^2 \ . \tag{17}$$

Using a proof similar to (12) by replacing $\boldsymbol{x}_{(c)}^{l}$ with $\boldsymbol{x}_{(c)}^{k,new}$, we can readily show that (17) and thus (9) are correct for $n_k > p$.

3.2.2. $n_k \leq p$

When $n_k \leq p$, we have $q = \text{rank}(\boldsymbol{X}_{(c)}) < p$, and thus $\boldsymbol{V}_q \in \mathbb{R}^{p\times q}$ is not square. Again, it follows that $(\boldsymbol{V}_q)^T\boldsymbol{V}_q = \boldsymbol{I}_q$ but $\boldsymbol{V}_q(\boldsymbol{V}_q)^T \neq \boldsymbol{I}_p$.

Let $\boldsymbol{m}^{k,new}$ denote the residual from using the q PC vectors to reconstruct $\boldsymbol{x}_{(c)}^{k,new}$: $\boldsymbol{m}^{k,new} = \boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$. We calculate the sum of squares

of the residuals in $\boldsymbol{m}^{k,new}$:

$$\begin{aligned} ||\boldsymbol{m}^{k,new}||_2^2 &= ||\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_p(\boldsymbol{V}_p)^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T||_2^2 \\ &= \sum_{i=q+1}^{p} (t_i^{k,new})^2 \ , \end{aligned} \tag{18}$$

where $||\cdot||_2$ denotes the Euclidean norm of a vector.

However, unlike the case for the training data, $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is not necessarily equal to zero for a p-dimensional test instance. Thus $\boldsymbol{x}_{(c)}^{k,new} \neq \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$, which means that the new test instance cannot be perfectly reconstructed by the first q PC vectors.

Hence, if we rewrite

$$\begin{aligned} \boldsymbol{x}_{(c)}^{k,new} &= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T + \boldsymbol{m}^{k,new} \\ &= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T + (\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T) + \boldsymbol{m}^{k,new} \ , \end{aligned} \tag{19}$$

we have

$$\begin{aligned} \boldsymbol{e}^{k,new} &= (\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T) + \boldsymbol{m}^{k,new} \\ &= (\boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T) + (\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T) \\ &= \boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T \end{aligned} \tag{20}$$

and

$$\begin{aligned}\sum_{j=1}^{p}(e_j^{k,new})^2 &= ||\boldsymbol{e}^{k,new}||_2^2 \\ &= ||\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T||_2^2 \\ &= \sum_{i=r+1}^{p}(t_i^{k,new})^2 \\ &= \sum_{i=r+1}^{q}(t_i^{k,new})^2 + \sum_{i=q+1}^{p}(t_i^{k,new})^2 \ . \end{aligned} \tag{21}$$

Comparing (21) with (17), we can find an additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ in (21), and this term may not be zero. It follows that (17) and thus (9) are not valid when $n_k \leq p$.

When $n_k \leq p$, $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is hard to estimate because the last $(p-q)$ PCs are randomly calculated by satisfying the orthogonal condition. Nevertheless, it can be harmful to the classification of the new instance of high-dimensional "large p, small n" data, if we use (9) to calculate $v^{k,new}$ which omits $\sum_{i=q+1}^{p}(t_i^{k,new})^2$, because the decision making for classification is based on $v^{k,new}$.

4. Experiments

In the following experiments, take SIMCA as an example: we compare the SIMCA with the OD defined originally in [28] (denoted by SIMCA) and the SIMCA with the OD calculated by following [9] (denoted by SIMCA-D), evaluating them on both simulated and real datasets. We aim to show that the additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ can be important for classifying high-

dimensional data. To simplify the experiment settings, we discuss the effect of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ on two-class classification in the experiments. The effect of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ on multi-class classification can be readily extended.

4.1. Classification rule

New test instances can be classified by following the classification rule of the robust SIMCA (RSIMCA) [3], which is a linear combination of the OD and the SD of a new test instance (Here our notations of OD and SD are both for *squared* distances). That is, a new test instance is classified to the class with the minimum value of

$$\gamma\frac{\mathrm{OD}^k}{c_{\mathrm{OD}}^k}+(1-\gamma)\frac{\mathrm{SD}^k}{c_{\mathrm{SD}}^k}\ , \tag{22}$$

where $\mathrm{OD}^k = v^{k,new}$; $\mathrm{SD}^k = (\boldsymbol{t}_r^{k,new})^T\boldsymbol{\Lambda}_r^{-1}\boldsymbol{t}_r^{k,new}$, in which $\boldsymbol{\Lambda}_r$ is the diagonal matrix of the r largest eigenvalues for the PC subspace; $c_{\mathrm{SD}}^k = \chi^2_{r;0.975}$; and $c_{\mathrm{OD}}^k = (\hat{\mu}+\hat{\sigma}z_{0.975})^3$, in which $\hat{\mu}$ and $\hat{\sigma}$ are the mean and the standard deviation of the square roots of $v^{k,l}$.

Since OD^k is the only term that is different between SIMCA and SIMCA-D, the value of the second term in (22) does not affect the difference between SIMCA and SIMCA-D. We force the value of the second term in (22) to zero by setting $\gamma = 1$, to simplify the experiments.

4.2. Validation criterion

We use the overall misclassification percentage (MP) as the validation criterion following the experiments in [3]. We use the one-assignment-rule suggested in [3], i.e. a test sample is assigned to one of the known classes

with the smallest F-value, to simplify the calculation of the MP and obtain unambiguous final results. The MP is defined as

$$\text{MP} = \sum_{k=1}^{K} n_k^t / N^t \ , \tag{23}$$

where n_k^t denotes the the number of wrongly assigned test samples in class k and N^t denotes the total number of test samples.

4.3. Datasets

4.3.1. Simulated datasets

Simulated datasets are generated by following the experiments in [18]. Assume that a sample vector $\boldsymbol{x}$ is the sum of two independent normal random components:

$$\boldsymbol{x} = \boldsymbol{\delta} + \boldsymbol{\epsilon} \ , \tag{24}$$

where

$$\boldsymbol{\delta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \text{ and } \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I}) \ . \tag{25}$$

Based on the above assumption, the samples of the two classes are drawn from $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 + \sigma_1^2 \boldsymbol{I})$ and $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2 + \sigma_2^2 \boldsymbol{I})$, respectively.

Two sets of parameters, simulation A and simulation B, are devised to show the following two situations, respectively: 1) $\sum_{i=q+1}^{p} (t_i^{k,new})^2$ is not important for classification; and 2) $\sum_{i=q+1}^{p} (t_i^{k,new})^2$ may be important for classification. The details of the two simulation settings are summarised in Table 1.

For each simulation setting, we generate 20 datasets with different n_k/p ratios to explore the difference between SIMCA and SIMCA-D with respect

Table 1: Simulation settings. Notation: K, number of classes; D, number of datasets; n_k, number of samples in each class

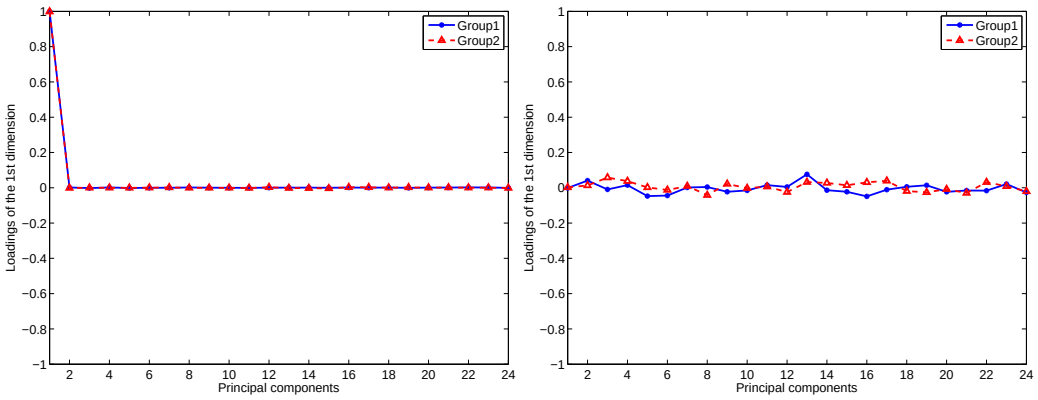
	Simulation A	Simulation B
$\boldsymbol{\mu}_1$	$\mathbf{0}_p$	$\mathbf{0}_p$
$\boldsymbol{\mu}_2$	$(10, \mathbf{0}_{p-1}^T)^T$	$(10, \mathbf{0}_{p-1}^T)^T$
$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$	$\begin{bmatrix} 5000 & 0.1 & 0.1 & \cdots & 0.1 \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1 \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1 \end{bmatrix}_{p\times p}$	$\begin{bmatrix} 0.1 & 0.1 & 0.1 & \cdots & 0.1 \\ 0.1 & 5000 & 0.1 & \cdots & 0.1 \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1 \end{bmatrix}_{p\times p}$
$\sigma_1^2 = \sigma_2^2$	0.1	0.1
K	2	2
D	20	20
n_k	50	50

to p. In each dataset, 50 samples are generated for each class, from which 25 samples are selected as the training set and the rest as the test set, i.e. n_1 and n_2 are fixed to 25 for all the datasets. The 20 n_k/p ratios are 1.5, 1, 0.7, 0.5, 0.3, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.009, 0.008, 0.007, 0.006 and 0.005; and the corresponding p's are 17, 25, 36, 50, 83, 250, 278, 313, 417, 500, 625, 833, 1250, 2500, 2778, 3125, 3571, 4167 and 5000. Among these settings, $n_k/p = 1.5$ (i.e. $p = 17$) indicates a low-dimensional dataset while other ratios indicate high-dimensional datasets.

It is clear in Table 1 that the only difference between simulation A and simulation B is the values of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, which determines the importance of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ for classification. In both simulations, the first dimensions of the feature vectors contain major discriminative information since $\mu_{11} = 0$ and $\mu_{21} = 10$, while other dimensions contain little discriminative information since $\mu_{1i} = \mu_{2i} = 0$ $(i \neq 1)$. Therefore, the variance of the first dimension determines how the discriminative information between two

classes is distributed to the PCs. The discriminative information left in the residuals for classification is determined by the discriminative information in the first few PCs used in the class subspace.

If the first dimension has the largest variance and the discriminative information is concentrated on the first PC which is definitely used in the class subspace, i.e. $(\mathbf{\Sigma}_1)_{11} = (\mathbf{\Sigma}_2)_{11} = 5000$ in simulation A, then $\sum_{j=1}^{p}(e_j^{k,new})^2$ is not very discriminative (or say unimportant for classification) and so is $\sum_{i=q+1}^{p}(t_i^{k,new})^2$. In contrast, if the first dimension has a small variance and contributes randomly to the PCs, i.e. $(\mathbf{\Sigma}_1)_{11} = (\mathbf{\Sigma}_2)_{11} = 0.1$ in simulation B, then the discriminative information may not be concentrated on the first few PCs that are used in the class subspace. In this case, $\sum_{j=1}^{p}(e_j^{k,new})^2$ can be discriminative (or say important for classification) and so be $\sum_{i=q+1}^{p}(t_i^{k,new})^2$.



(a) Simulation A. (b) Simulation B.

Figure 1: The loading plots of the first dimension.

Here we show an example to demonstrate the above argument. Two datasets with $p = 1250$ are generated. Applying PCA separately to the two classes of each dataset, we obtain the PCs for each class. We record the

first entries of all the PCs in each class, i.e. $\boldsymbol{V}_q(1,:)$, and plot them against the PCs sorted in decreasing order of singular values, as shown in Figure 1 for simulation A and simulation B, respectively. These loadings indicate the contributions of the first dimensions of the feature vectors to the PCs.

In simulation A, the absolute loadings of the first PC are close to one while those of other PCs are close to zeros, which indicates that the discriminative information between the two classes is concentrated on the the first PC. Since the first PC is definitely used to build the class subspace, $\sum_{j=1}^{p}(e_j^{k,new})^2$ contains little discriminative information from the first dimension. Thus, as a part of $\sum_{j=1}^{p}(e_j^{k,new})^2$, $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is not important for classification.

In simulation B, the loadings are distributed randomly around zero, which indicates that the discriminative information is spread over all PCs. Therefore, $\sum_{j=1}^{p}(e_j^{k,new})^2$ may contain discriminative information important for classification and so be $\sum_{i=q+1}^{p}(t_i^{k,new})^2$.

4.3.2. Real datasets

A low-dimensional dataset (the iris data) and three high-dimensional datasets (the Phenyl data, the meat data and the fat data) are used in the experiments.

The iris dataset [12] contains 150 samples with three classes: each class contains 50 samples. Each sample is described by four features.

The Phenyl dataset is provided in the R package, 'chemometrics'. The dataset consists of 600 mass spectrum of chemical components, with 300 compounds contain the phenyl substructure and 300 compounds do not contain the substructure. Each spectra contains 658 mass spectral features. We randomly select 100 samples from the Phenyl dataset for our experiments,

with 50 contain the phenyl substructure and 50 do not contain the structure.

The meat dataset [1] consists of 108 spectra of meat spectra measured at 1051 wavelengths, with 55 chicken samples and 54 turkey samples.

The fat dataset [11] consists of 193 spectra of finely chopped meat, with 122 meat samples of less than 20% fat and 71 samples of larger than 20% fat. Each spectrum is measured at 100 wavelengths.

4.4. Experiment settings

For the iris data and the Phenyl data, we randomly select 25 samples from each class to generate the training set. For the meat data, we randomly select 27 chicken samples and 27 turkey samples for training. For the fat data, we randomly select 35 samples of less than 20% fat and 35 samples of larger than 20% fat for training. The remaining samples of each dataset generate the test set.

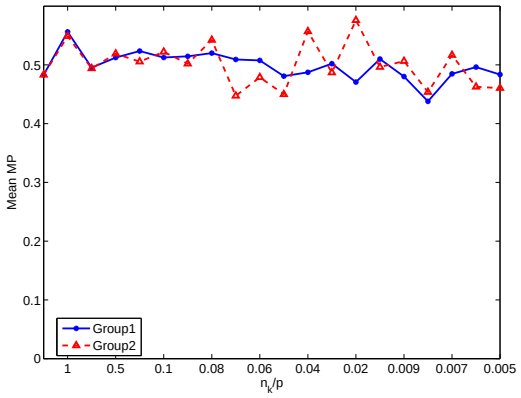
We repeat this procedure 100 times and perform the two methods, SIMCA and SIMCA-D, on each training-test split.

In both methods, the number of PCs are chosen using the criterion that the variance explained is more than 85% for all classes. Thus the numbers of PCs, r, are the same for the two methods.

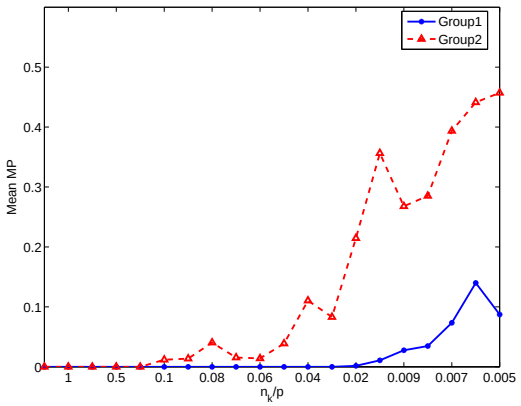
4.5. Results

4.5.1. Simulated datasets

To explore the effect of the n_k/p ratio on the performances of SIMCA and SIMCA-D, we plot the the mean MP against the n_k/p ratio in Figure 2 for simulation A and simulation B, respectively. It is clear that the mean MPs of SIMCA and SIMCA-D are the same when $n_k/p = 1.5$, i.e. in the



(a) Simulation A.



(b) Simulation B.

Figure 2: The plots of mean MP against n_k/p.

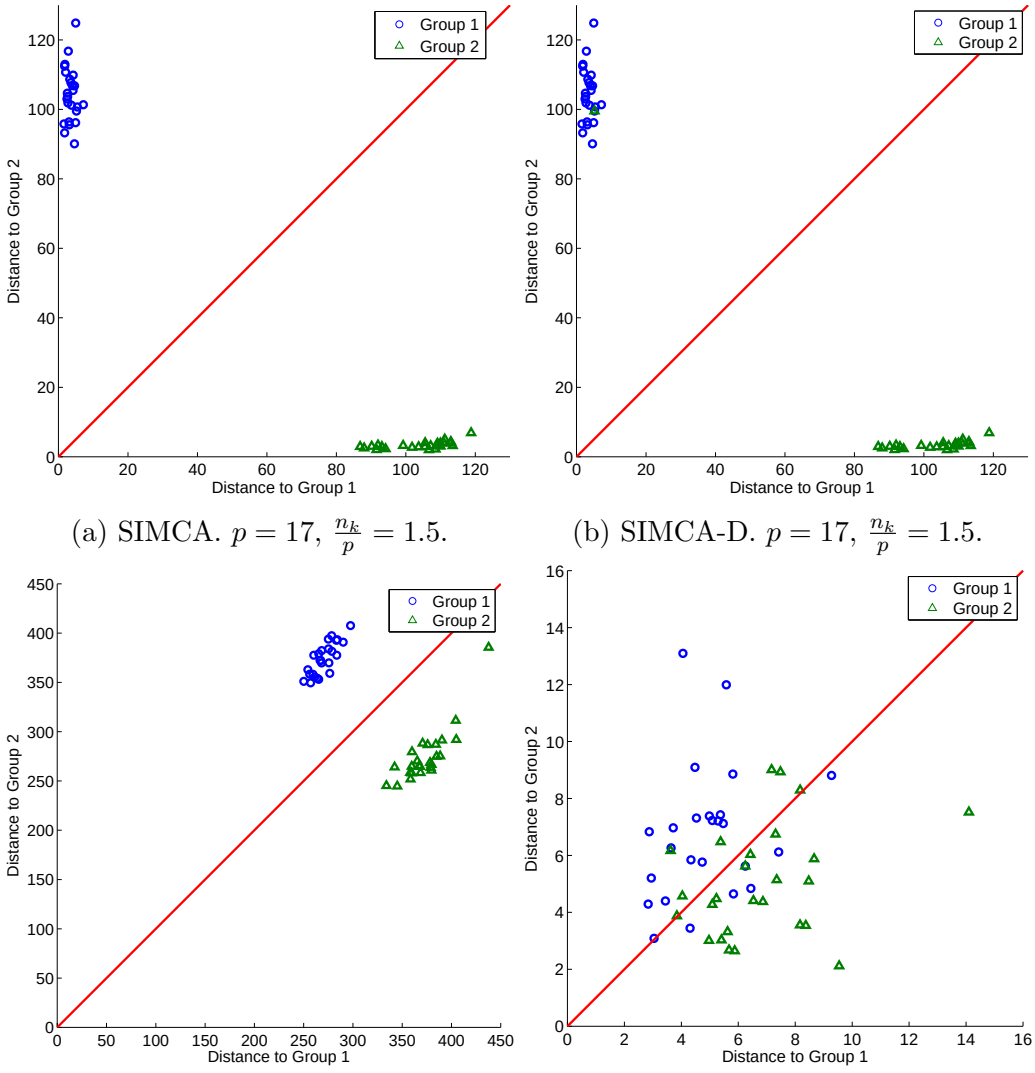
low-dimensional situation, in each of the simulation settings, as indicated by the leftmost points in each panel of Figure 2.

However, the relative performances of SIMCA and SIMCA-D are different for the two simulations when $n_k/p \leq 1$, i.e. in the high-dimensional situation.

In simulation A, the mean MPs of the two methods are similar for all n_k/p ratios, as shown in Figure 2a. This indicates that ignoring $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ in the calculation of the OD does not affect the classification results in this simulation, because in this case $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is not important for classification. In addition, since the residuals are not discriminative, the mean MP varies around 0.5.

In simulation B, the difference between the mean MPs of the two methods becomes larger as n_k/p becomes smaller (i.e. when the data are higher dimensional), as shown in Figure 2b. Since in this simulation the first few PCs used in class subspaces contain little discriminative information, the residual $\sum_{j=1}^{p}(e_j^{k,new})^2$ is important for classification. SIMCA performs pretty well for almost all the n_k/p ratios because $\sum_{j=1}^{p}(e_j^{k,new})^2$ captures the discriminative information for classification. In contrast, SIMCA-D, which only uses $\sum_{i=r+1}^{q}(t_i^{k,new})^2$ for classification and ignores $\sum_{i=q+1}^{p}(t_i^{k,new})^2$, cannot capture the discriminative information in $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ and can be suboptimal in classification, especially when n_k/p is small (i.e. when the data dimension is high). For example, the mean MP of SIMCA-D worsens to around 0.4 when n_k/p decreases to 0.008.

In addition for simulation B, we show an example of how $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ affects the classification performance using the Coomans' plots. Figure 3 shows the Coomans' plots of the test samples on one training-test split of



(a) SIMCA. $p = 17$, $\frac{n_k}{p} = 1.5$.

(b) SIMCA-D. $p = 17$, $\frac{n_k}{p} = 1.5$.

(c) SIMCA. $p = 1250$, $\frac{n_k}{p} = 0.02$.

(d) SIMCA-D. $p = 1250$, $\frac{n_k}{p} = 0.02$.

Figure 3: Coomans' plots.

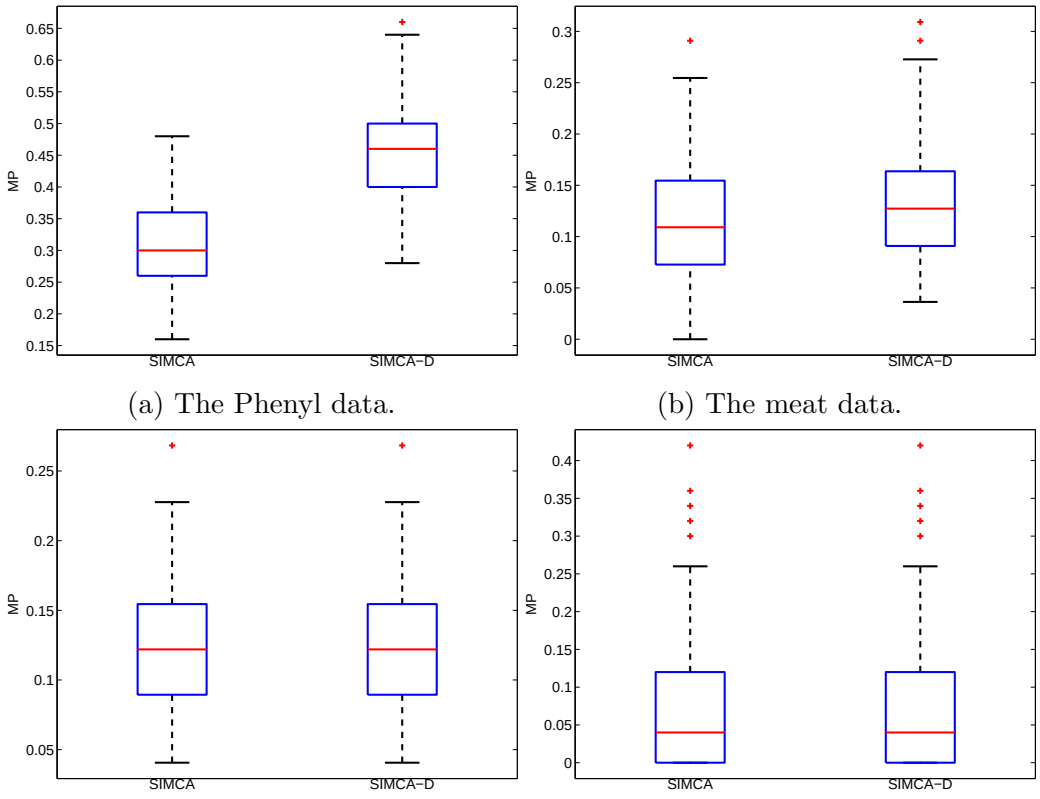
each simulated dataset. The Coomans' plot [26] shows the orthogonal distance from the test samples to two class subspaces at the same time. In our experiments, the horizontal and vertical axes denote the ODs to Group 1 and Group 2, respectively. In Figure 3, the red reference line divides the Coomans' plot into two parts: in the upper triangular part, the distance to Group 1 is smaller than that to Group 2; in the lower triangular part, it is the other way around.

Since SIMCA and SIMCA-D have the same q and r, the Coomans' plots reflect the difference between the ODs of these two methods.

When $n_k/p = 1.5$ (i.e. low-dimensional), the Coomans' plots of the two methods are the same. When $n_k/p = 0.02$ (i.e. high-dimensional), the Coomans' plots of the two methods are different. We observe large differences between the values of ODs in Figure 3c and Figure 3d, which indicates that the value of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is large. Including $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ can perfectly separate the two groups as shown in Figure 3c; however, omitting $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ results in a mixture of the two groups as shown in Figure 3d. This indicates that the additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is important for classification in this high-dimensional simulated dataset.

4.5.2. Real datasets

Figure 4 shows the box plots of the MP for the real datasets. In the high-dimensional Phenyl data and the high-dimensional meat data, SIMCA-D provides worse classification performance than the original SIMCA. However, in the high-dimensional fat data, SIMCA-D and SIMCA provides the same classification results. The results suggest that SIMCA-D can provide worse classification results than SIMCA for some high-dimensional real datasets. In



(a) The Phenyl data.

(b) The meat data.

(c) The fat data.

(d) The iris data.

Figure 4: The box plots of the MP for the real datasets.

the low-dimensional iris dataset, the two methods provide the same results. This pattern for the real datasets is consistent with that for the simulated datasets.

5. Conclusion

We have investigated the formulae in [9] of calculating two ODs, $v^{k,l}$ and $v^{k,new}$. We have shown that the formula for $v^{k,new}$ in [9] is not valid for high-dimensional data (i.e. when $n_k \leq p$). The experiments on both the simulated datasets and the real datasets have confirmed that the formula following [9] can result in worse classification performance than the original one in [28]. Therefore, we suggest that the original formulae in [28] for calculating the ODs, rather than the formulae in [9], should be used for the classification of high-dimensional data which have more features than samples (i.e. when $n_k \leq p$).

Acknowledgment

The authors would like to thank the reviewers for their constructive comments.

References

[1] T. Arnalds, J. McElhinney, T. Fearn, G. Downey, A hierarchical discriminant analysis for species identification in raw meat by visible and near infrared spectroscopy, Journal of Near Infrared Spectroscopy 12 (3) (2004) 183–188.

[2] S. Bicciato, A. Luchini, C. Di Bello, PCA disjoint models for multiclass cancer analysis using gene expression data, Bioinformatics 19 (5) (2003) 571–578.

[3] K. V. Branden, M. Hubert, Robust classification in high dimensions based on the SIMCA method, Chemometrics and Intelligent Laboratory Systems 79 (1) (2005) 10–21.

[4] A. Candolfi, R. De Maesschalck, D. Jouan-Rimbaud, P. Hailey, D. Massart, The influence of data pre-processing in the pattern recognition of excipients near-infrared spectra, Journal of Pharmaceutical and Biomedical Analysis 21 (1) (1999) 115–132.

[5] A. Candolfi, R. De Maesschalck, D. Massart, P. Hailey, A. Harrington, Identification of pharmaceutical excipients using NIR spectroscopy and SIMCA, Journal of Pharmaceutical and Biomedical Analysis 19 (6) (1999) 923–935.

[6] Q. Chen, J. Zhao, H. Zhang, X. Wang, Feasibility study on qualitative and quantitative analysis in tea by near infrared spectroscopy with multivariate calibration, Analytica Chimica Acta 572 (1) (2006) 77–84.

[7] N. C. da Silva, M. F. Pimentel, R. S. Honorato, M. Talhavini, A. O. Maldaner, F. A. Honorato, Classification of Brazilian and foreign gasolines adulterated with alcohol using infrared spectroscopy, Forensic science international 253 (2015) 33–42.

[8] M. Daszykowski, K. Kaczmarek, I. Stanimirova, Y. Vander Heyden,

B. Walczak, Robust SIMCA-bounding influence of outliers, Chemometrics and Intelligent Laboratory Systems 87 (1) (2007) 95–103.

[9] R. De Maesschalck, A. Candolfi, D. Massart, S. Heuerding, Decision criteria for soft independent modelling of class analogy applied to near infrared data, Chemometrics and Intelligent Laboratory Systems 47 (1) (1999) 65–77.

[10] R. De Maesschalck, D. Jouan-Rimbaud, D. L. Massart, The Mahalanobis distance, Chemometrics and Intelligent Laboratory Systems 50 (1) (2000) 1–18.

[11] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer Science & Business Media, 2006.

[12] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (2) (1936) 179–188.

[13] L. Jiao, F. Shang, F. Wang, Y. Liu, Fast semi-supervised clustering with enhanced spectral embedding, Pattern Recognition 45 (12) (2012) 4358–4369.

[14] N. Kumar, A. Bansal, G. Sarma, R. K. Rawal, Chemometrics tools used in analytical chemistry: An overview, Talanta 123 (2014) 186–199.

[15] B. Mnassri, E. M. EI Adel, B. Ananou, M. Ouladsine, Fault detection and diagnosis based on PCA and a new contribution plot, IFAC Proceedings Volumes 42 (8) (2009) 834–839.

[16] B. Mnassri, M. Ouladsine, et al., Reconstruction-based contribution approaches for improved fault diagnosis using principal component analysis, Journal of Process Control 33 (2015) 60–76.

[17] A. L. Pomerantsev, Acceptance areas for multivariate classification derived by projection methods, Journal of Chemometrics 22 (11-12) (2008) 601–609.

[18] A. L. Pomerantsev, O. Y. Rodionova, Concept and role of extreme objects in PCA/SIMCA, Journal of Chemometrics 28 (5) (2014) 429–438.

[19] A. L. Pomerantsev, O. Y. Rodionova, On the type II error in SIMCA method, Journal of Chemometrics 28 (6) (2014) 518–522.

[20] M. Rafferty, X. Liu, D. M. Laverty, S. McLoone, Real-time multiple event detection and classification using moving window PCA, IEEE Transactions on Smart Grid 7 (5) (2016) 2537–2548.

[21] O. Y. Rodionova, P. Oliveri, A. L. Pomerantsev, Rigorous and compliant approaches to one-class classification, Chemometrics and Intelligent Laboratory Systems 159 (2016) 89–96.

[22] R. Shang, Z. Zhang, L. Jiao, C. Liu, Y. Li, Self-representation based dual-graph regularized feature selection clustering, Neurocomputing 171 (2016) 1242–1253.

[23] R. Shang, Z. Zhang, L. Jiao, W. Wang, S. Yang, Global discriminative-based nonnegative spectral clustering, Pattern Recognition 55 (2016) 172–182.

[24] V. Uríčková, J. Sádecká, Determination of geographical origin of alcoholic beverages using ultraviolet, visible and infrared spectroscopy: A review, Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy 148 (2015) 131–137.

[25] P. Van den Kerkhof, J. Vanlaer, G. Gins, J. F. Van Impe, Analysis of smearing-out in contribution plot based fault isolation for statistical process control, Chemical Engineering Science 104 (2013) 285–293.

[26] B. G. Vandeginste, D. L. Massart, Handbook of Chemometrics and Qualimetrics, Elsevier Science, 1998.

[27] E. E. Waddell, M. R. Williams, M. E. Sigman, Progress toward the determination of correct classification rates in fire debris analysis II: utilizing soft independent modeling of class analogy (SIMCA), Journal of Forensic Sciences 59 (4) (2014) 927–935.

[28] S. Wold, Pattern recognition by means of disjoint principal components models, Pattern Recognition 8 (3) (1976) 127–139.

[29] R. Zhu, K. Fukui, J.-H. Xue, Building a discriminatively ordered subspace on the generating matrix to classify high-dimensional spectral data, Information Sciences 382-383 (2017) 1–14.

[30] Y. Zontov, O. Y. Rodionova, S. Kucheryavskiy, A. Pomerantsev, DD-SIMCA: A MATLAB GUI tool for data driven SIMCA approach, Chemometrics and Intelligent Laboratory Systems 167 (2017) 23–28.