Learning distance to subspace for nearest subspace methods in high-dimensional data classification

Rui Zhu[a,b,c,*], Mingzhi Dong[c], Jing-Hao Xue[c]

[a] *Faculty of Actuarial Science and Insurance, Cass Business School, City, University of London, London EC1Y 8TZ, UK*

[b] *School of Mathematics, Statistics and Actuarial Sciences, University of Kent, Canterbury CT2 7FS, UK*

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

Abstract

Nearest subspace methods (NSM) are a category of classification methods widely applied to classify high-dimensional data. In this paper, we propose to improve the classification performance of NSM through learning tailored distance metrics from samples to class subspaces. The learned distance metric is termed as 'learned distance to subspace' (LD2S). Using LD2S in the classification rule of NSM can make the samples closer to their correct class subspaces while farther away from their wrong class subspaces. In this way, the classification task becomes easier and the classification performance of NSM can be improved. The superior classification performance of using LD2S for NSM is demonstrated on three real-world high-dimensional spectral datasets.

Keywords: NSM, distance to subspace, distance metric learning, orthogonal distance, score distance

[*] Corresponding author: Tel.: +44(0)1227 82 7008

Email addresses: `rui.zhu@city.ac.uk` (Rui Zhu), `mingzhi.dong.13@ucl.ac.uk` (Mingzhi Dong), `jinghao.xue@ucl.ac.uk` (Jing-Hao Xue)

1. Introduction

Classification of high-dimensional data is an important research topic [8, 9, 10, 27, 28]. Subspace-based classification methods have been widely applied to classify high-dimensional data. Face recognition [11, 4, 7], chemometrics [22, 2, 5, 27] and process control in engineering [14, 20, 15, 17] are famous application areas of subspace-based classification methods. In subspace-based classification methods, classes are first modelled by low-dimensional subspaces. Then the test sample is classified using a classification rule that measures the similarities between the test sample and the class subspaces, and the test sample is assigned to its most similar class.

The principal component (PC) subspaces are commonly adopted as the low-dimensional class subspaces. They are believed to be good representations of high-dimensional data, because most variable information in the data is extracted to the leading PCs and the redundant information in the original features is discarded.

Two distances associated with the PC subspaces are usually used in the classification rules: the *squared* orthogonal distance (OD^2) and the *squared* score distance (SD^2). OD^2 measures the squared orthogonal distance between a sample and a PC subspace [28], while SD^2 measures the squared Mahalanobis distance between the projection of a sample onto a PC subspace and the centre of the PC subspace. When the distances are used in the classification rule, the test sample is assigned to the class with the smallest score of the classification rule. In this paper, we term the PC subspace-based classification methods with the classification rule using distances "nearest subspace methods" (NSM).

The nearest subspace classifier (NSC) [11, 25, 4, 3, 13] and soft independent modelling of class analogy (SIMCA) [22, 2, 5, 18, 16, 12] are two famous examples of NSM. NSC and SIMCA both adopt PC subspace as the low-dimensional class subspace, however, they use different classification rules to classify a test sample. In NSC, OD^2 between the test sample and its projection on a class subspace is used as the classification rule. The test sample is assigned to the class with the smallest OD^2. In SIMCA, the linear combination of OD^2 and SD^2 is usually used as the classification rule. The test sample is assigned to the class with the smallest score of the linear combination.

However, the standard distances OD^2 and SD^2 may not always be able to capture or reflect well the mechanism underlying the semantic similarity or dissimilarity between the sample and the subspace. In fact, this is also the case with other generic distance metrics, such as the Euclidean distance and the Mahalanobis distance. This has led to the proposals of metric learning in the machine learning community, which enables automatic learning of a tailored distance metric from the data available.

More specifically, given the PC class subspaces, the distances used in the classification rule play vital roles in classification. Currently, OD^2 and SD^2 are the two distances widely used in the classification rule, both of which use predetermined distance metrics: OD^2 uses the Euclidean distance while SD^2 uses the Mahalanobis distance. However, different data usually prefer different distance metrics to reflect different semantic concepts of dissimilarity or similarity in the context of problems, and hence adapting the distance metrics to different data can be expected to improve the classification performance of NSM. On the other hand, distance metric learning methods emerging in the machine learning community provide us a tool to learn tailored distance metrics automatically from data and to improve the classification performance [23, 21, 26, 19, 24].

However, the existing distance metric learning methods in the literature aim to improve the classification methods that are based on distances between samples, such as k-nearest neighbours (kNN). Thus the distance metrics that they learned are for the distances between samples. But unfortunately the distance metrics used in NSM measure the distances between samples and class subspaces. This makes those established distance metric learning methods unable to be applied directly to NSM.

Therefore in this paper, we propose a distance metric learning method tailored for NSM to improve its classification performance. We first analyse the classification rules of NSM adopted in the literature, and we derive a general formulation for them. We show that the general formulation is based on two parameterisation matrices with different sizes; hence different classification rules of NSM in the literature can be shown actually using different distance metrics within the general formulation.

We define this general formulation as the distance metric from a sample to a class subspace, and propose a method of learning distance to subspace, to automatically learn the two parameterisation matrices that define the distance metric. Then, inspired by the distance metric learning strategy, we learn this distance metric based on a set of distance-to-subspace-based similarity/dissimilarity constraints: the samples are similar to their correct class subspaces while are dissimilar from the wrong class subspaces. Using the learned distance as the similarity measure, we aim to make the samples to be closer to their correct class subspaces while be farther away from their wrong class subspaces. We term this distance metric "learned distance to subspace (LD2S)".

The contributions of this paper are summarised as follows.

First, we are the first to derive a general formulation for the classification rules of nearest subspace methods used in literature. Based on the general formulation, we can design new classification rules, by specifying $\boldsymbol{M}_1^k$ and $\boldsymbol{M}_2^k$. This formulation is a guidance for researchers to design new classification rules for nearest subspace methods with better classification performance.

Second, based on the general formulation, we develop a novel distance metric learning method for nearest subspace methods. Most of the current literature of distance metric learning methods are only designed for classification methods based on distances between samples. Here we design a distance metric learning method for methods based on distances between a sample and a subspace. In this paper, we have shown an effective distance metric learning method, LS2D, to classify high-dimensional data.

To evaluate the effectiveness of LD2S, we compare the the classification performances of NSC [4], SIMCA [22, 2] and NSM with the classification rule learned from LD2S (NSM-LD2S) using three real-world high-dimensional datasets.

2. Methodology

2.1. NSM

2.1.1. PC class subspace

Given the training set of class k ($k = 1, 2$), $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times p}$, we build the PC class subspace of the kth class by using the reduced singular value decomposition (SVD):

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

where $\boldsymbol{X}_{k(c)}$ is the column-centred training set, the rows of $\boldsymbol{U}_{q_k} \in \mathbb{R}^{n_k \times q_k}$ ($q_k = \text{rank}(\boldsymbol{X}_{k(c)})$) are the standardised PC scores, $\boldsymbol{D}_{q_k} \in \mathbb{R}^{q_k \times q_k}$ is a diagonal matrix with singular values $d_1 \geq d_2 \geq \ldots \geq d_{q_k} \geq 0$ on the diagonal, and the columns of $\boldsymbol{V}_{q_k} \in \mathbb{R}^{p \times q_k}$ are the PCs. The PC score is defined as

$$\boldsymbol{T}_{q_k} = \boldsymbol{U}_{q_k} \boldsymbol{D}_{q_k} = \boldsymbol{X}_{k(c)} \boldsymbol{V}_{q_k} \in \mathbb{R}^{n_k \times q_k}. \tag{2}$$

If we select the first $r_k \leq q_k$ PCs to build the kth class subspace, then

$$\boldsymbol{X}_{k(c)} = \boldsymbol{U}_{r_k} \boldsymbol{D}_{r_k} \boldsymbol{V}_{r_k}^T + \boldsymbol{E}_k, \tag{3}$$

where $\boldsymbol{U}_{r_k} \in \mathbb{R}^{n_k \times r_k}$, $\boldsymbol{D}_{r_k} \in \mathbb{R}^{r_k \times r_k}$, $\boldsymbol{V}_{r_k} \in \mathbb{R}^{p \times r_k}$, and $\boldsymbol{E}_k \in \mathbb{R}^{n_k \times p}$ is the residual matrix when reconstructing the training samples $\boldsymbol{X}_{k(c)}$ using the first r_k PCs. The PC subspace spanned by the first r_k PCs is associated with a unique projection matrix $\boldsymbol{P}_k = \boldsymbol{V}_{r_k} \boldsymbol{V}_{r_k}^T \in \mathbb{R}^{p \times p}$. We denote the PC subspace for class k as $\mathcal{L}_k$.

Projecting a new sample $\boldsymbol{x}_{new} \in \mathbb{R}^{1 \times p}$ to the PC class subspace, we could obtain

$$\boldsymbol{x}_{(c)}^{k,new} = \boldsymbol{t}^{k,new}\boldsymbol{V}_{r_k}^T + \boldsymbol{e}^{k,new}, \tag{4}$$

where $\boldsymbol{x}_{(c)}^{k,new}$ is the centred $\boldsymbol{x}_{new}$ by the column means of $\boldsymbol{X}_k$, $\boldsymbol{t}^{k,new} \in \mathbb{R}^{1\times r}$ is the PC score of the new sample, and $\boldsymbol{e}^{k,new} \in \mathbb{R}^{1\times p}$ is the residual of reconstructing the new sample by the PC class subspace.

2.1.2. Two distances associated with the PC class subspace

Given the PC class subspaces, the new sample $\boldsymbol{x}_{new}$ is classified using a classification rule that is based on two distances related the PC class subspaces: the squared orthogonal distance (OD^2) and the squared score distance (SD^2). In this section, we discuss the calculation and the geometric intuition of OD^2 and SD^2.

The squared orthogonal distance. The squared orthogonal distance from $\boldsymbol{x}_{new}^c$ to the subspace of the kth class, OD_k^2, is defined based on the residual $\boldsymbol{e}^{k,new}$ in (4):

$$\mathrm{OD}_k^2 = \sum_{j=1}^{p}(e_j^{k,new})^2 = \boldsymbol{e}^{k,new}(\boldsymbol{e}^{k,new})^T, \tag{5}$$

which is the squared Frobenius norm of $\boldsymbol{e}^{k,new}$.

Rewriting (4), we have

$$\boldsymbol{e}^{k,new} = \boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k = \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{I}_p - \boldsymbol{P}_k), \tag{6}$$

where $\boldsymbol{I}_p$ denotes the p-by-p identity matrix. The $\boldsymbol{e}^{k,new}$ can then be considered as the difference vector between $\boldsymbol{x}_{(c)}^{k,new}$ and its projection on $\mathcal{L}_k$, $\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k$. The orthogonal complement of $\mathcal{L}_k$ is $\mathcal{L}_k^{\perp}$ which has the projection matrix $\boldsymbol{I}_p - \boldsymbol{P}_k$. Thus $\boldsymbol{e}^{k,new}$ is also the projection of $\boldsymbol{x}_{(c)}^{k,new}$ to the subspace $\mathcal{L}_k^{\perp}$. Since $\boldsymbol{e}^{k,new}$ is orthogonal to $\mathcal{L}_k$, the distance based on $\boldsymbol{e}^{k,new}$ is called the orthogonal distance. An illustration of OD_k^2 in a 3-dimensional feature

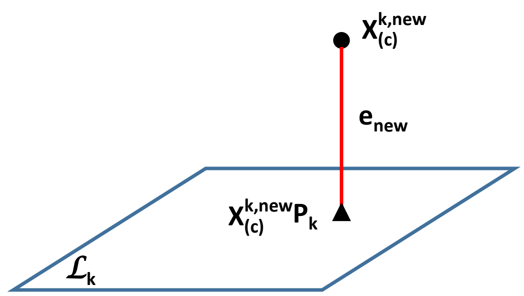


Figure 1: An illustration of OD_k^2 in a 3-dimensional feature space.

space is shown in Figure 1. The new instance $\boldsymbol{x}_{(c)}^{k,new}$ is shown as the black dot; the class subspace $\mathcal{L}_k$ is shown as the dark blue 2-dimensional plane; and the projection of $\boldsymbol{x}_{(c)}^{k,new}$ to $\mathcal{L}_k$, $\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k$, is shown as the black triangle. The residual $\boldsymbol{e}^{k,new}$ is represented by the red solid line segment, which is orthogonal to the plane $\mathcal{L}_k$. The square of the length of the red line segment is OD_k^2.

The squared score distance. The squared score distance to class k, SD_k^2, is defined as the Mahalanobis distance from the projection of $\boldsymbol{x}_{(c)}^{k,new}$ to the centre of the subspace $\mathcal{L}_k$:

$$\mathrm{SD}_k^2 = \sum_{i=1}^{r_k} (t_i^{k,new}/d_i)^2 = \boldsymbol{t}^{k,new}\boldsymbol{D}_{r_k}^{-2}(\boldsymbol{t}^{k,new})^T, \tag{7}$$

where $\boldsymbol{D}_{r_k}$ is the diagonal matrix of singular values in (3). SD_k^2 is the reweighted squared Frobenius norm of $\boldsymbol{t}^{k,new}$ with weights $1/d_i$ $(i = 1, 2, \ldots, r)$ and $1/d_1 \leq 1/d_2 \leq \ldots \leq 1/d_{r_k}$. An illustration of SD_k^2 in a 3-dimensional

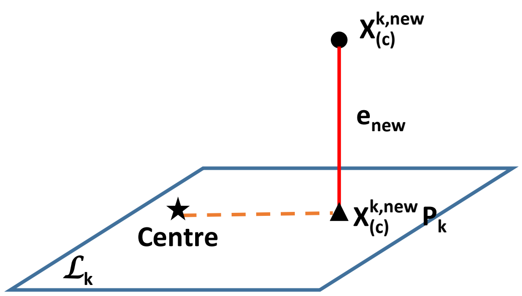


Figure 2: An illustration of SD_k^2 in a 3-dimensional feature space.

feature space is shown in Figure 2. In addition to the symbols in Figure 1, the centre of the class subspace, $\mathcal{L}_k$, is shown as the black star, and the orange dashed line connects the centre of the class subspace and the projection of $\boldsymbol{x}_{(c)}^{k,new}$ to the class subspace. The SD_k^2 is then the reweighted length of the orange dashed line.

2.1.3. The classification rules

In NSC, the classification rule is

$$\mathrm{OD}_k^2. \tag{8}$$

NSC assigns $\boldsymbol{x}_{new}$ to the class with the smallest OD_k^2.

In SIMCA, a linear combination of OD_k^2 and SD_k^2 is often used as the classification rule [2]:

$$\gamma \left(\frac{\mathrm{OD}_k}{c^k_{\mathrm{OD}^2}} \right)^2 + (1-\gamma) \left(\frac{\mathrm{SD}_k}{c^k_{\mathrm{SD}^2}} \right)^2, \tag{9}$$

where $\gamma \in [0,1]$ and $c^k_{\mathrm{OD}^2}$ and $c^k_{\mathrm{SD}^2}$ are the cutoff values of OD_k^2 and SD_k^2 calculated from the training set of the kth class. When $\gamma = 1$, (9) only depends on OD_k^2, and is the same as (8) if the cutoff value $c^k_{\mathrm{OD}^2}$ in (9) is one. When $\gamma = 0$, (9) only depends on SD_k^2. In practice, the value of γ can be set by the users based on their prior knowledge of the importance of OD_k^2 and SD_k^2, or can be tuned by cross-validation using the training set.

2.2. A general formulation for the classification rules for NSM

Although the classification rules in NSM are in different forms, as shown in (8) and (9), we shall show that they can be written using the following general formulation:

$$\boldsymbol{x}^{k,new}_{(c)} \boldsymbol{M}^k_1 (\boldsymbol{x}^{k,new}_{(c)})^T - \boldsymbol{t}^{k,new} \boldsymbol{M}^k_2 (\boldsymbol{t}^{k,new})^T, \tag{10}$$

with different $\boldsymbol{M}^k_1 \in \mathbb{R}^{p\times p}$ and $\boldsymbol{M}^k_2 \in \mathbb{R}^{r_k \times r_k}$. In this section, we derive this general formulation based on the classification rules (8) and (9), and show $\boldsymbol{M}^k_1$ and $\boldsymbol{M}^k_2$ for (8) and (9), respectively. Based on the derived general formulation of the classification rules, we will define the distance to subspace and propose a method to learn the distance to subspace in the next section.

Substituting (6) into (5), we obtain

$$\begin{aligned}\mathrm{OD}_k^2 &= (\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k)(\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k)^T \\ &= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - 2\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k(\boldsymbol{x}_{(c)}^{k,new})^T + \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k^2(\boldsymbol{x}_{(c)}^{k,new})^T \\ &= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k(\boldsymbol{x}_{(c)}^{k,new})^T \\ &= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T, \end{aligned} \tag{11}$$

which indicates that OD_k^2 is the difference between the squared Frobenius norm of $\boldsymbol{x}_{(c)}^{k,new}$ and the squared Frobenius norm of $\boldsymbol{t}^{k,new}$. This is intuitive if we think about the right-angled triangle formed by $\boldsymbol{x}_{(c)}^{k,new}$, $\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k$ and the centre of $\mathcal{L}_k$ in Figure 2.

Then the classification rule (8) can be written as

$$\begin{aligned}&\boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T \\ &= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{M}_{1(NSC)}^k(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}\boldsymbol{M}_{2(NSC)}^k(\boldsymbol{t}^{k,new})^T,\end{aligned} \tag{12}$$

where $\boldsymbol{M}_{1(NSC)}^k = \boldsymbol{I}_p$ and $\boldsymbol{M}_{2(NSC)}^k = \boldsymbol{I}_{r_k}$. Equation (12) indicates that the classification rule of NSC provides equal weights to the p dimensions in the linear combination of the original features $\boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T$ and also equal weights to the r_k dimensions in the linear combination of the scores $\boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T$.

Similarly, for the classification rule of SIMCA, we substitute (11) to (9):

$$\begin{aligned}
&\frac{\gamma}{(c^k_{\mathrm{OD}^2})^2}(\boldsymbol{x}^{k,new}_{(c)}(\boldsymbol{x}^{k,new}_{(c)})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T) + \frac{1-\gamma}{(c^k_{\mathrm{SD}^2})^2}\boldsymbol{t}^{k,new}\boldsymbol{D}_r^{-2}(\boldsymbol{t}^{k,new})^T \\
&= \frac{\gamma}{(c^k_{\mathrm{OD}^2})^2}\boldsymbol{x}^{k,new}_{(c)}(\boldsymbol{x}^{k,new}_{(c)})^T - \sum_{i=1}^{r}(-\frac{1-\gamma}{(c^k_{\mathrm{SD}^2})^2} + \frac{\gamma}{(c^k_{\mathrm{OD}^2})^2 d_i^2})t_i^2 \\
&= \boldsymbol{x}^{k,new}_{(c)}\boldsymbol{M}^k_{1(S)}(\boldsymbol{x}^{k,new}_{(c)})^T - \boldsymbol{t}^{k,new}\boldsymbol{M}^k_{2(S)}(\boldsymbol{t}^{k,new})^T, \qquad (13)
\end{aligned}$$

where $\boldsymbol{M}^k_{1(S)} = \frac{1}{h_1}\boldsymbol{I}_p$, $h_1 = \frac{\gamma}{(c^k_{\mathrm{OD}^2})^2}$ and $\boldsymbol{M}^k_{2(S)}$ is an r_k-by-r_k diagonal matrix with $(-\frac{1-\gamma}{(c^k_{\mathrm{SD}^2})^2} + \frac{\gamma}{(c^k_{\mathrm{OD}^2})^2 d_i^2})$ on the diagonals (d_i's are the singular values in $\boldsymbol{D}$ with $d_1 \geq d_2 \geq \ldots \geq d_{r_k} \geq 0$). Different from the classification rule of NSM in (12), the rule in (13) indicates that the classification rule of SIMCA provides equal weights to the p dimensions in the linear combination of the the original features $\boldsymbol{x}^{k,new}_{(c)}(\boldsymbol{x}^{k,new}_{(c)})^T$, while providing different weights to the r_k dimensions in the linear combination of the scores $\boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T$.

2.3. Learning distance to subspace

We define the general formulation (10) as the distance from $\boldsymbol{x}_{new}$ to the kth class subspace. Hence we assign $\boldsymbol{x}_{new}$ to the nearest class subspace based on the distance to subspace defined in (10).

The distance to subspace for the kth class defined in (10) depends on two matrices: $\boldsymbol{M}^k_1$ and $\boldsymbol{M}^k_2$. It can be treated as the difference between two squared distances: $\boldsymbol{x}^{k,new}_{(c)}\boldsymbol{M}^k_1(\boldsymbol{x}^{k,new}_{(c)})^T$ is the squared distance from $\boldsymbol{x}^{k,new}_{(c)}$ to the centre of the class subspace $\mathcal{L}_k$, and $\boldsymbol{t}^{k,new}\boldsymbol{M}^k_2(\boldsymbol{t}^{k,new})^T$ is the squared distance from the projection of $\boldsymbol{x}^{k,new}_{(c)}$ to $\mathcal{L}_k$ to the centre of $\mathcal{L}_k$.

The matrices $\boldsymbol{M}^k_1$ and $\boldsymbol{M}^k_2$ are of great importance for classification. Instead of determining $\boldsymbol{M}^k_1$ and $\boldsymbol{M}^k_2$ manually as in [22] and [2], distance metric learning methods offer us a path to learn more appropriate distance metrics automatically from the training data to improve the classification performance.

Distance metric learning methods aim to learn distance metrics based on a set of similarity/dissimilarity constraints: the samples from the same class should be similar while the samples from different classes should be dissimilar. Thus the samples from the same class are close together while the samples from different classes are farther away from each other, based on the distance metric learned from the training data. In this way, the classification task becomes easier and we can expect better classification performance using the learned distance metrics.

Established distance metric learning methods are sample-based, i.e. the distances that they learned are measured between samples. However, in NSM, the distance is calculated between a sample and a class subspace. Thus we need to develop a new method of learning the distance metric from sample to subspace, to learn the distance metrics in NSM. The learned distance metrics are termed "learned distance to subspace (LD2S)". Inspired by the constraints used in established distance metric learning methods, we propose the following set of similarity/dissimilarity constraints for LD2S: the samples should be similar to their true class while dissimilar from the wrong classes. In other words, we aim to learn $\boldsymbol{M}_1^k$ and $\boldsymbol{M}_2^k$, such that the samples are close to their true classes while farther away from the wrong classes.

2.3.1. Distance metric

In this section, we briefly review the definition of distance metric. Given a set of data points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ in $\mathbb{R}^{1\times p}$ with a set of labels $\{y_1, y_2, ..., y_N\}$, the distance metric $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ should satisfy the following properties:

1. $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ (non-negativity),
2. $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ if and only if $\boldsymbol{x}_i = \boldsymbol{x}_j$ (identity),
3. $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = d(\boldsymbol{x}_j, \boldsymbol{x}_i)$ (symmetry),
4. $d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq d(\boldsymbol{x}_i, \boldsymbol{x}_k) + d(\boldsymbol{x}_j, \boldsymbol{x}_k)$ (triangle inequality), where $\boldsymbol{x}_k$ is an instance that is different to $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

A distance metric is known as a pseudo metric when the second property is relaxed to: $d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ if $\boldsymbol{x}_i = \boldsymbol{x}_j$.

Most of the metric learning algorithms aim to learn a Mahalanobis distance-like pseudo metric:

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)\boldsymbol{M}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T}, \tag{14}$$

which is parameterised by $\boldsymbol{M}$. The matrix $\boldsymbol{M}$ is set to be positive semidefinite to ensure that $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a pseudo metric. If $\boldsymbol{M}$ is the inverse of the sample variance, then $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the Mahalanobis distance. If $\boldsymbol{M}$ is the identity matrix, then $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is exactly the Euclidean distance.

2.3.2. Distance to subspace

Different from the distance metric between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ defined in (14), we define the squared distance metric between a sample $\boldsymbol{x}$ and a class subspace $\mathcal{L}_k$ using the general formulation in (10):

$$d^2(\boldsymbol{x}, \mathcal{L}_k) = \boldsymbol{x}^k_{(c)}\boldsymbol{M}^k_1(\boldsymbol{x}^k_{(c)})^T - \boldsymbol{t}^k\boldsymbol{M}^k_2(\boldsymbol{t}^k)^T, \tag{15}$$

where $\boldsymbol{x}_{(c)}^k$ denotes the sample mean-centred by the mean of the training samples of the kth class, $\boldsymbol{M}_1^k \in \mathbb{R}^{p \times p}$ is the parameterisation matrix for the distance in the original feature space of the kth class, $\boldsymbol{t}^k$ is the PC score of the sample when projected to the PC subspace of the kth class, and $\boldsymbol{M}_2^k \in \mathbb{R}^{r_k \times r_k}$ is the parameterisation matrix for the distance in the PC subspace of the kth class. Then $d^2(\boldsymbol{x}, \mathcal{L}_k)$ can be treated as the difference between the squared distance from the sample (column-centred by the column means of class k) to the centre of $\mathcal{L}_k$ and the squared distance from the projection of the sample to the centre of $\mathcal{L}_k$.

2.3.3. Learned distance to subspace

To learn good distance metrics between samples and class subspaces, we propose the following similarity/dissimilarity constraints: the samples are similar to their correct class subspaces while are dissimilar to the wrong class subspaces. To formulate the constraints, we define the following similarity/dissimilarity sets:

$\boldsymbol{S} = \{(\boldsymbol{x}_i, \mathcal{L}_k) \mid \boldsymbol{x}_i \text{ belongs to class } k\}$, and

$\boldsymbol{D} = \{(\boldsymbol{x}_i, \mathcal{L}_k) \mid \boldsymbol{x}_i \text{ does not belong to class } k\}$.

In the following part, the training samples from class 1 are denoted by subscript $1(i)$, i.e. $\boldsymbol{x}_{1(i)} \in \mathbb{R}^{1 \times p}$ and $\boldsymbol{X}_1 = [\boldsymbol{x}_{1(1)}^T, \ldots, \boldsymbol{x}_{1(n_1)}^T]^T \in \mathbb{R}^{n_1 \times p}$, and the training samples from class 2 are denoted by subscript $2(j)$, i.e. $\boldsymbol{x}_{2(j)} \in \mathbb{R}^{1 \times p}$ and $\boldsymbol{X}_2 = [\boldsymbol{x}_{2(1)}^T, \ldots, \boldsymbol{x}_{2(n_2)}^T]^T \in \mathbb{R}^{n_2 \times p}$. Thus the similarity/dissimilarity sets become

$\boldsymbol{S} = \{(\boldsymbol{x}_{1(i)}, \mathcal{L}_1), (\boldsymbol{x}_{2(j)}, \mathcal{L}_2) \mid i = 1, 2, \ldots, n_1, j = 1, 2, \ldots, n_2\}$, and

$\boldsymbol{D} = \{(\boldsymbol{x}_{1(i)}, \mathcal{L}_2), (\boldsymbol{x}_{2(j)}, \mathcal{L}_1) \mid i = 1, 2, \ldots, n_1, j = 1, 2, \ldots, n_2\}$.

One straightforward way to find tailored distance metrics is to minimise the sum of the distances between the samples and the class subspaces that fall into the similarity set $\boldsymbol{S}$, while maximise the sum of those that fall into the dissimilarity set $\boldsymbol{D}$. However, simply optimising the sums of the distances suffers from losing the information in individual samples. Hence, instead of treating all training samples together, we aim to make the difference between the distance to the wrong class and the distance to the correct class large enough for each training sample by using the following constraints:

$$\begin{aligned} & d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_2) - d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_1) \geq 1, \text{ for } i = 1, \ldots, n_1, \text{ and} \\ & d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_1) - d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_2) \geq 1, \text{ for } j = 1, \ldots, n_2. \end{aligned} \tag{16}$$

In this way, the samples can be classified more easily. In addition, to enhance the generalisation ability of the learned distance metrics, we add slack variables $\xi_{1(i)}$ and $\xi_{2(j)}$ to the constraints and aim to solve the following optimisation problem:

$$\min_{\xi_{1(i)}, \xi_{2(j)}, \boldsymbol{M}_1^k, \boldsymbol{M}_2^k} \quad \sum_{i=1}^{n_1} \xi_{1(i)} + \sum_{j=1}^{n_2} \xi_{2(j)} \tag{17}$$

$$\text{s.t.} \quad d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_2) - d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_1) \geq 1 - \xi_{1(i)}, \quad \xi_{1(i)} \geq 0, \tag{18}$$

$$d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_1) - d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_2) \geq 1 - \xi_{2(j)}, \quad \xi_{2(j)} \geq 0, \tag{19}$$

$$\boldsymbol{M}_1^k \succeq 0 \text{ and } \boldsymbol{M}_2^k \succeq 0, \tag{20}$$

where $\boldsymbol{M}_1^k \succeq 0$ and $\boldsymbol{M}_2^k \succeq 0$ denote that $\boldsymbol{M}_1^k$ and $\boldsymbol{M}_2^k$ are positive semidefinite. The constraints in (18) and (19) can be rewritten as

$$
\begin{aligned}
&\xi_{1(i)} \geq [1 + d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_1) - d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_2)]_+ \text{ and} \\
&\xi_{2(j)} \geq [1 + d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_2) - d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_1)]_+,
\end{aligned}
$$

where $[l]_+ = \max(0, l)$. Hence the optimisation problem is equivalent to

$$
\begin{aligned}
\min_{\boldsymbol{M}_1^k, \boldsymbol{M}_2^k} \quad & \sum_{i=1}^{n_1} [1 + d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_1) - d^2(\boldsymbol{x}_{1(i)}, \mathcal{L}_2)]_+ + \\
& \sum_{j=1}^{n_2} [1 + d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_2) - d^2(\boldsymbol{x}_{2(j)}, \mathcal{L}_1)]_+ \\
& \text{s.t. } \boldsymbol{M}_1^k \succeq 0, \quad \boldsymbol{M}_2^k \succeq 0.
\end{aligned} \tag{21}
$$

The hinge losses used in (21) only penalise the samples that do not satisfy (16), while assign zero loss for the samples that satisfy (16) using NSM. In this way, the hinge loss makes full use of the effectiveness of NSM. It is worth noting that the hinge loss has also been popularly used in other distance-based classifiers, such as support vector machine (SVM) and large margin nearest neighbour (LMNN) classification [21].

Suppose $\boldsymbol{M}_1^{k*}$ and $\boldsymbol{M}_2^{k*}$ $(k = 1, 2)$ denote the solutions of (21). Then the learned distance from a test sample $\boldsymbol{x}_{new}$ to the kth class subspace is

$$
d^2(\boldsymbol{x}_{new}, \mathcal{L}_k) = \boldsymbol{x}_{(c)}^{k,new} \boldsymbol{M}_1^{k*} (\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new} \boldsymbol{M}_2^{k*} (\boldsymbol{t}^{k,new})^T. \tag{22}
$$

We compare $d^2(\boldsymbol{x}_{new}, \mathcal{L}_1)$ and $d^2(\boldsymbol{x}_{new}, \mathcal{L}_2)$, and assign $\boldsymbol{x}_{new}$ to the class with the smallest squared distance.

Considering the nature of spectral data, i.e. high-dimensional feature and small sample size, learning the full matrices, $\boldsymbol{M}_1^k$ with $p(p+1)/2$ parameters and $\boldsymbol{M}_2^k$ with $r_k(r_k+1)/2$ parameters, could easily suffer from the overfitting problem. In (12) and (13), $\boldsymbol{M}_{1(NSC)}^k = \boldsymbol{I}_p$ and $\boldsymbol{M}_{1(S)}^k = \frac{1}{h_1}\boldsymbol{I}_p$ are identity matrices with common coefficients 1 and $1/h_1$ for all dimensions, respectively. Therefore, in this paper, we learn $\boldsymbol{M}_1^k = c_k\boldsymbol{I}_p$(with $c_k \geq 0$) and $\boldsymbol{M}_2^k = \mathrm{diag}(m_{21}^k, m_{22}^k, \ldots, m_{2r_k}^k)$ (with each element nonnegative), as natural and practically-interpretable extensions of those used in (12) and (13).

3. Experiments

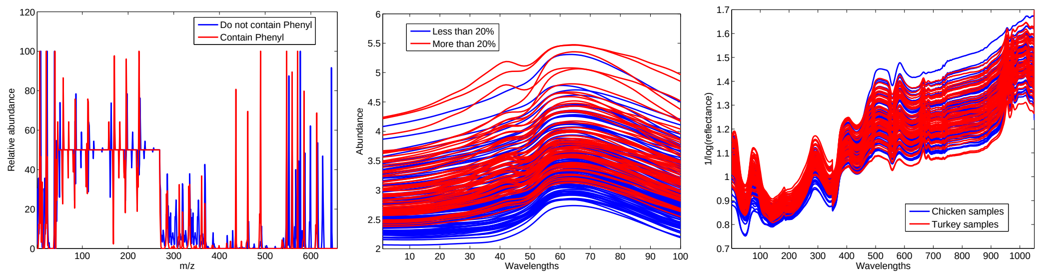
In the following experiments, NSC, SIMCA and NSM with distance measurement (22) (NSM-LD2S) are compared using high-dimensional spectral data, the Phenyl dataset, the fat dataset [6] and the meat dataset [1]. We also compare the classification results of the nearest subspace methods with those of naive Bayes (NB), k nearest neighbours (kNN) and support vector machine (SVM), to show the effectiveness of the nearest subspace methods to classify high-dimensional data.

3.1. Datasets

The number of samples in each class and the number of features for the three high-dimensional spectral datasets are summarised in Table 1.

Table 1: The number of samples in each class, n_1 and n_2, and the number of features p for the three high-dimensional spectral datasets.

	n_1	n_2	p
Phenyl	300	300	658
Fat	122	71	100
Meat	54	55	1050



(a) The Phenyl dataset. (b) The fat dataset. (c) The meat dataset.

Figure 3: The plots of the spectra of the three datasets.

3.1.1. The Phenyl dataset

The Phenyl dataset is available in the ‘chemometrics’ R package, which contains 300 spectra with the phenyl substructure and 300 spectra without the phenyl substructure. The spectra are measured at 658 wavelengths. To avoid confusing, the spectra of two instances from two classes are shown in Figure 3a.

3.1.2. The fat dataset

The fat dataset contains 193 spectra of finely chopped meat, measured at 100 wavelengths [6]. The fat dataset consists of 122 spectra of meat samples with less than 20% fat and 71 spectra of meat samples with more than 20% fat. The spectra of all samples are shown in Figure 3b.

3.1.3. The meat dataset

The meat dataset [1] contains the spectra of five classes of meat samples, measured at 1050 wavelengths. We select the chicken and turkey meat samples from the original dataset in the experiments, because they contain similar chemical components and are hard to classify. The new meat dataset contains the spectra of 55 chicken samples and the spectra of 54 turkey samples. The spectral of all samples are shown in Figure 3c.

3.2. Experiment settings

The classification performances of the three methods are shown for five different ratios of training set size/feature dimension: $n_1/p = n_2/p = 0.1$, 0.2, 0.3, 0.4 and 0.5.

For the Phenyl dataset, we randomly select 100 samples with Phenyl structure and 100 samples without Phenyl structure. For illustrative purposes, we select the first 100 dimensions from the 658 feature dimensions for the experiments in this paper, i.e. $p = 100$.

For the fat dataset, we use all the 120 meat samples with less than 20% fat and 71 meat samples with more than 20% fat in the dataset. We also use all the dimensions of the fat dataset, i.e. $p = 100$.

For the meat dataset, we use all the 55 chicken samples and 54 turkey samples in the dataset. Again for illustrative purposes, we also select the first 100 dimensions from the 350 dimensions for the experiments in this paper, i.e. $p = 100$.

Therefore, as $p = 100$ for each of the three datasets, the five training set sizes are $n_1 = n_2 = 10$, 20, 30, 40 and 50. The samples to form a training set are randomly selected from a dataset. The rest samples in the datasets are used as test samples.

In NSC, SIMCA and NSM-LD2S, the numbers of PCs, r_k, are tuned by 5-fold cross-validation using the training set to minimise the classification error. More specifically, for each value of r_k, we calculate the mean classification error of the 5-fold cross-validation. The value with the minimum mean classification error is chosen as the number of PCs.

In SIMCA, $c^k_{OD} = (\hat{\mu} + \hat{\sigma} z_{0.975})^{3/2}$, where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and the standard deviation of the orthogonal distances in of the training samples in class k; and $c^k_{SD} = \sqrt{\chi^2_{n_k;0.975}}$. The weight γ is also tuned by 5-fold cross-validation using the training data.

In NSM-LD2S, the optimisation problem (21) is solved by 'cvx' in MATLAB.

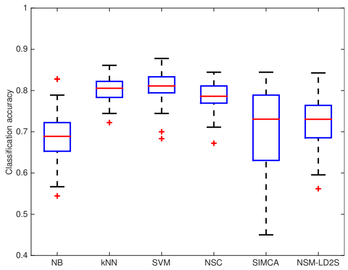
In SVM, the radial basis function (RBF) kernel is adopted. The scale parameter of the RBF kernel and the penalty factor C are tune by 5-fold cross-validation. The values of the two parameters to be chosen are set to 10, 10^2 and 10^3. In kNN, the number of nearest neighbours is tuned by 5-fold cross-validation. The values to be chosen are set to 3, 5 and 7. In NB, the prior probability of each class is set as the proportion of the number of training samples of that class over the total number of training samples.

All the random training/test splits and the subsequent experiments are repeated 100 times and the classification accuracies of the test data are recorded.

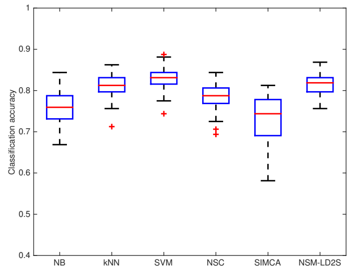
3.3. Results

3.3.1. The Phenyl dataset

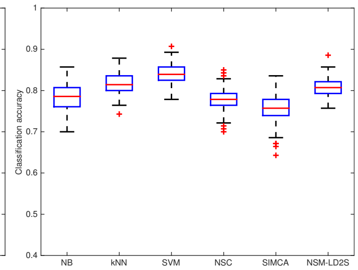
The classification results of the Phenyl dataset demonstrate the superior classification performance of NSM-LD2S, as shown in Figure 4 and Figure 5, compared with NSC and SIMCA over all n_k/p ratios. It is clear that SVM performs better than the three nearest subspace methods for this dataset. kNN and NB are also better than the three nearest subspace methods when n_k/p becomes large.



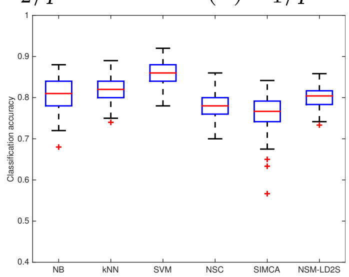
(a) $n_1/p = n_2/p = 0.1$.



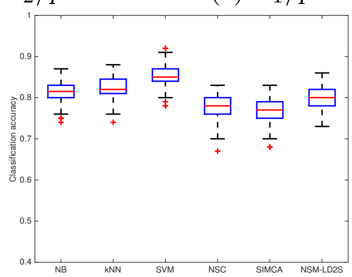
(b) $n_1/p = n_2/p = 0.2$.



(c) $n_1/p = n_2/p = 0.3$.



(d) $n_1/p = n_2/p = 0.4$.



(e) $n_1/p = n_2/p = 0.5$.

Figure 4: Classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the Phenyl dataset.

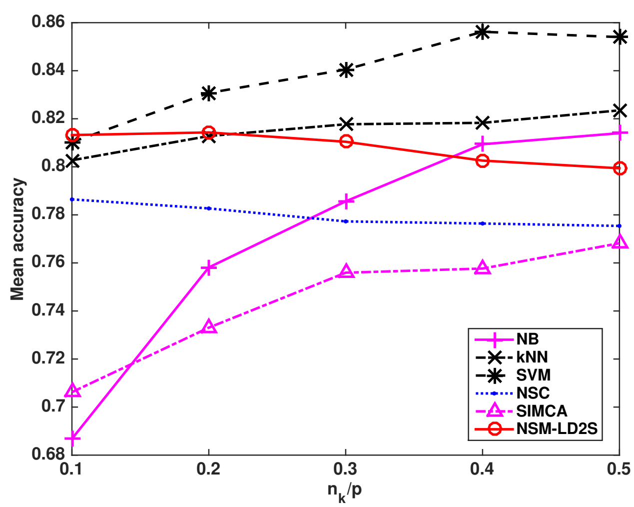


Figure 5: Mean classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the Phenyl dataset.

However, it is conceivable that, for certain other datasets, the classification performance of NSM-LD2S cannot always be better than those of NSC and SIMCA, in particular under small n_k/p ratios. In the following two sections, we show two examples that NSM-LD2S performs worse than NSC and SIMCA for small n_k/p ratios but better for large n_k/p ratios. This is because there are more parameters in NSM-LD2S to be learned than in NSC and SIMCA, and NSM-LD2S needs more training samples to achieve good classification performance for some data. In addition, the classification performances of NB, kNN and SVM are also not always better than the nearest subspace methods. The following two examples can also demonstrate this argument.

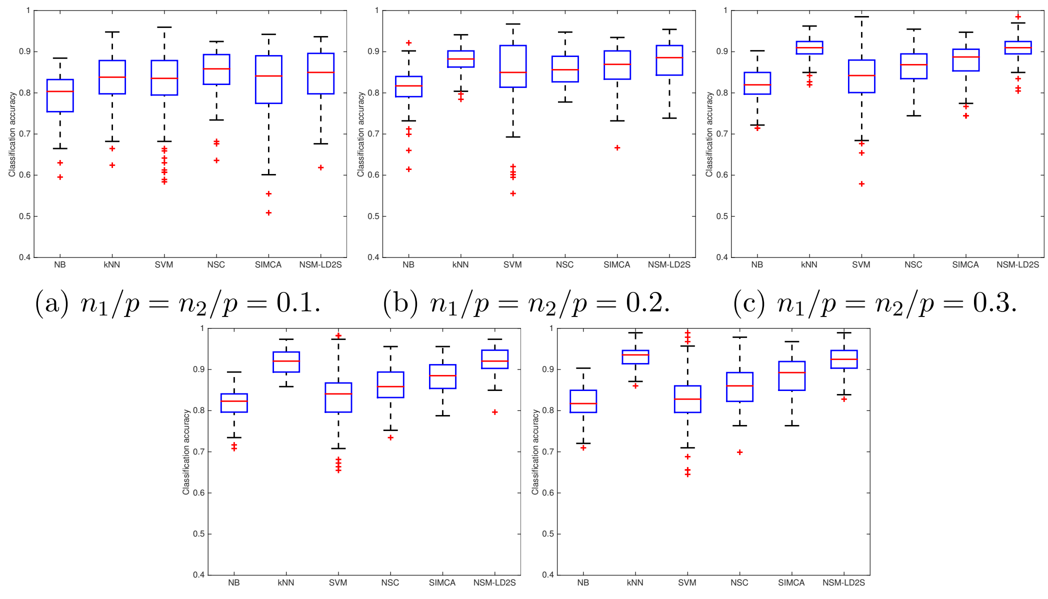
3.3.2. The fat dataset

In the fat dataset, the classification performance of NSM-LD2S and SIMCA are worse than NSC when $n_k/p = 0.1$ and are better than NSC when $n_k/p \geq 0.2$, as shown in Figure 6 and Figure 7. NSM-LD2S provides the best classification performance when $n_k/p \geq 0.2$.

It is obvious that NB has the worst mean classification accuracies for all n_k/p ratios. kNN performs similarly to NSM-LD2S. SVM performs similarly to SIMCA when $n_k/p = 0.1$ and performs worse than the three nearest subspace methods for all other n_k/p ratios.

3.3.3. The meat dataset

Compared with the fat dataset, the classification accuracies of the three methods for the meat dataset show a stronger effect of the n_k/p ratios. When $n_k/p < 0.4$, NSM-LD2S performs much worse than NSC and SIMCA, espe-



(a) $n_1/p = n_2/p = 0.1$. (b) $n_1/p = n_2/p = 0.2$. (c) $n_1/p = n_2/p = 0.3$.

(d) $n_1/p = n_2/p = 0.4$. (e) $n_1/p = n_2/p = 0.5$.

Figure 6: Classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the fat dataset.

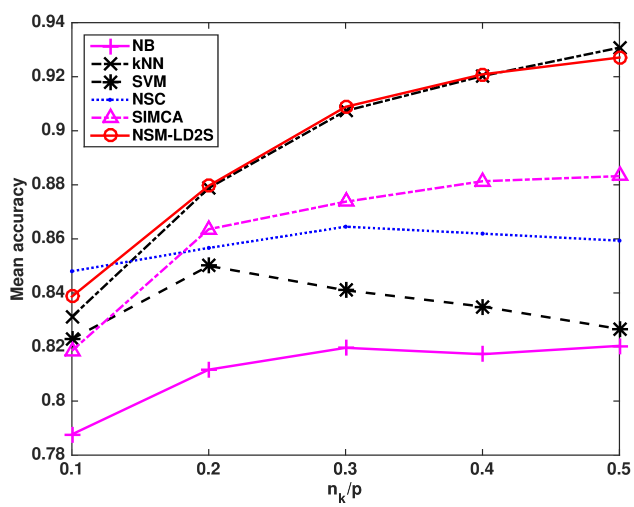
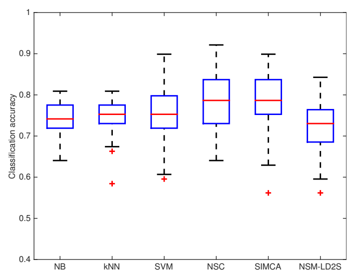
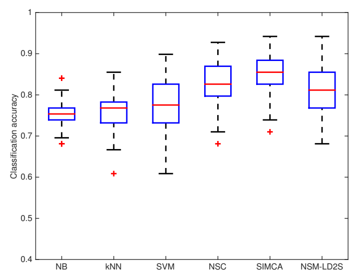


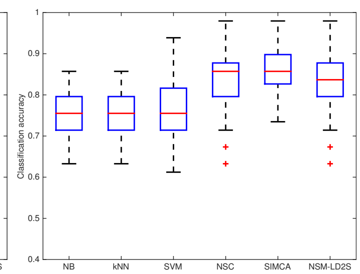
Figure 7: Mean classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the fat dataset.



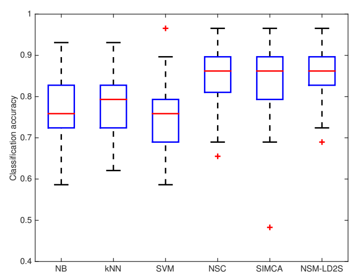
(a) $n_1/p = n_2/p = 0.1$.



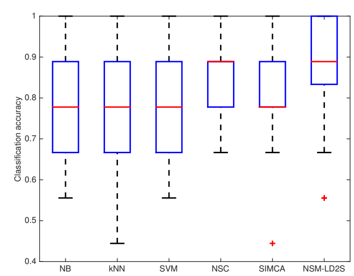
(b) $n_1/p = n_2/p = 0.2$.



(c) $n_1/p = n_2/p = 0.3$.



(d) $n_1/p = n_2/p = 0.4$.



(e) $n_1/p = n_2/p = 0.5$.

Figure 8: Classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the meat dataset.

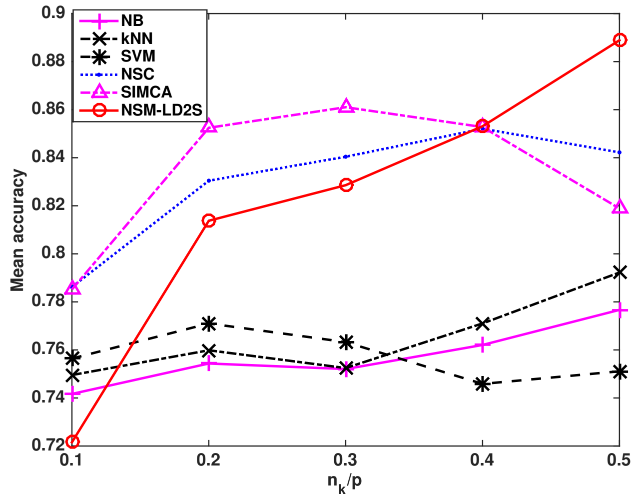


Figure 9: Mean classification accuracies of NB, kNN, SVM, NSC, SIMCA and NSM-LD2S for the meat dataset.

cially for $n_k/p = 0.1$. However, when $n_k/p = 0.5$, the classification accuracies of NSM-LD2S become much better than those of NSC and SIMCA, as shown in Figure 8(e) and Figure 9. The classification results of the meat dataset suggest that NSM-LD2S needs $n_k/p > 0.4$ to achieve superior classification performance for the meat dataset.

Similarly to the fat dataset, NB and SVM have the worst classification performances for $n_k/p > 0.1$ for the meat dataset. kNN performs worse than the nearest subspace methods for the meat dataset.

3.3.4. Summary of the results

The experiments show that using the learned distance metrics from data can provide superior classification results, compared with using predetermined distance metrics, when the n_k/p ratio is large enough. For data with small n_k/p ratios, using the distance measurement based on LD2S may perform poorly in classification since the n_k/p ratio is not large enough to learn all the parameters in LD2S.

It is worth noting that the nearest subspace methods are effective to classify high-dimensional data. One important reason is that they find the low-dimensional subspace representation for each class to extract the most informative feature. Our proposed LD2S is an additional step to improve the classification performance of the nearest subspace methods, based on the feature-extracted data. LD2S can obtain better distance measurements between a sample and a subspace, which has a positive effect on classification accuracies. As demonstrated by the experiment results, NSM-LD2S can achieve better classification accuracies than NSM and SIMCA, which shows the effectiveness of LD2S in addition to feature extraction in NSM and SIMCA.

4. Conclusion

We have proposed a general formulation of distance to subspace, i.e. the distance from a sample to a PC class subspace. Based on this formulation, we have proposed a simple but effective LD2S method that can learn tailored distance metrics adaptively from data, for the classification rule of NSM. The classification performances on three datasets demonstrate the effectiveness of learning distance metrics from data when the n_k/p ratio is large enough. The current LD2S is designed for binary classification. A multi-class version of LD2S is needed for more general and practical cases and we identify this as our future work.

Acknowledgement

The authors 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] 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.

[3] Y. Chi, Nearest subspace classification with missing data, in: Signals, Systems and Computers, 2013 Asilomar Conference on, IEEE, 2013, pp. 1667–1671.

[4] Y. Chi, F. Porikli, Connecting the dots in multi-class classification: From nearest subspace to collaborative representation, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3602–3609.

[5] C. Durante, R. Bro, M. Cocchi, A classification tool for N-way array based on SIMCA methodology, Chemometrics and Intelligent Laboratory Systems 106 (1) (2011) 73–85.

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

[7] K. Fukui, A. Maki, Difference subspace and its generalization for subspace-based methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (11) (2015) 2164–2177.

[8] P. Hall, D. M. Titterington, J.-H. Xue, Median-based classifiers for high-dimensional data, Journal of the American Statistical Association 104 (488) (2009) 1597–1608.

[9] P. Hall, J.-H. Xue, Incorporating prior probabilities into high-dimensional classifiers, Biometrika 97 (1) (2010) 31–48.

[10] P. Hall, J.-H. Xue, On selecting interacting features from high-dimensional data, Computational Statistics & Data Analysis 71 (2014) 694–708.

[11] K.-C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 684–698.

[12] C. Mees, F. Souard, C. Delporte, E. Deconinck, P. Stoffelen, C. Stévigny, J.-M. Kauffmann, K. De Braekeleer, Identification of coffee leaves using FT-NIR spectroscopy and SIMCA, Talanta 177 (2018) 4–11.

[13] J.-X. Mi, D.-S. Huang, B. Wang, X. Zhu, The nearest-farthest subspace classification for face recognition, Neurocomputing 113 (2013) 241–250.

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

[15] 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.

[16] I. Nejadgholi, M. Bolic, A comparative study of PCA, SIMCA and Cole model for classification of bioimpedance spectroscopy measurements, Computers in biology and medicine 63 (2015) 42–51.

[17] 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.

[18] A. Sgarbossa, C. Costa, P. Menesatti, F. Antonucci, F. Pallottino, M. Zanetti, S. Grigolato, R. Cavalli, A multivariate SIMCA index as discriminant in wood pellet quality assessment, Renewable Energy 76 (2015) 258–263.

[19] Q. Tian, S. Chen, L. Qiao, Ordinal margin metric learning and its extension for cross-distribution image data, Information Sciences 349 (2016) 50–64.

[20] 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.

[21] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, Journal of Machine Learning Research 10 (2009) 207–244.

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

[23] E. P. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning with application to clustering with side-information, Advances in Neural Information Processing Systems 15 (2003) 505–512.

[24] J. Yu, D. Tao, J. Li, J. Cheng, Semantic preserving distance metric learning and applications, Information Sciences 281 (2014) 674–686.

[25] L. Zhang, W.-D. Zhou, B. Liu, Nonlinear nearest subspace classifier, in: International Conference on Neural Information Processing, Springer, 2011, pp. 638–645.

[26] P. Zhu, Q. Hu, W. Zuo, M. Yang, Multi-granularity distance metric learning via neighborhood granule margin maximization, Information Sciences 282 (2014) 321–331.

[27] 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 (2017) 1–14.

[28] R. Zhu, J.-H. Xue, On the orthogonal distance to class subspaces for high-dimensional data classification, Information Sciences 417 (2017) 262–273.