



City Research Online

City St George's, University of London

Citation: Zhu, R., Dong, M. & Xue, J-H. (2018). Learning distance to subspace for the nearest subspace methods in high-dimensional data classification. Information Sciences, 481, pp. 69-80. doi: 10.1016/j.ins.2018.12.061

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://city-ref-test.eprints-hosting.org/id/eprint/21194/>

Link to published version: <https://doi.org/10.1016/j.ins.2018.12.061>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

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

25 methods” (NSM).

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

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

43 More specifically, given the PC class subspaces, the distances used in the
44 classification rule play vital roles in classification. Currently, OD^2 and SD^2
45 are the two distances widely used in the classification rule, both of which
46 use predetermined distance metrics: OD^2 uses the Euclidean distance while
47 SD^2 uses the Mahalanobis distance. However, different data usually prefer
48 different distance metrics to reflect different semantic concepts of dissimilar-
49 ity 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 (k NN). 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

75 class subspaces while are dissimilar from the wrong class subspaces. Using
 76 the learned distance as the similarity measure, we aim to make the samples
 77 to be closer to their correct class subspaces while be farther away from their
 78 wrong class subspaces. We term this distance metric “learned distance to
 79 subspace (LD2S)”.

80 The contributions of this paper are summarised as follows.

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

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

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

97 2. Methodology

98 2.1. NSM

99 2.1.1. PC class subspace

100 Given the training set of class k ($k = 1, 2$), $\mathbf{X}_k \in \mathbb{R}^{n_k \times p}$, we build the PC
 101 class subspace of the k th class by using the reduced singular value decompo-
 102 sition (SVD):

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

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

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

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

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

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

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

114 obtain

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

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

118 2.1.2. Two distances associated with the PC class subspace

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

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

$$\text{OD}_k^2 = \sum_{j=1}^p (e_j^{k,new})^2 = \mathbf{e}^{k,new} (\mathbf{e}^{k,new})^T, \quad (5)$$

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

128 Rewriting (4), we have

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

129 where $\mathbf{I}_p$ denotes the p -by- p identity matrix. The $\mathbf{e}^{k,new}$ can then be con-
 130 sidered as the difference vector between $\mathbf{x}_{(c)}^{k,new}$ and its projection on $\mathcal{L}_k$,
 131 $\mathbf{x}_{(c)}^{k,new} \mathbf{P}_k$. The orthogonal complement of $\mathcal{L}_k$ is $\mathcal{L}_k^\perp$ which has the projection

132 matrix $\mathbf{I}_p - \mathbf{P}_k$. Thus $\mathbf{e}^{k,new}$ is also the projection of $\mathbf{x}_{(c)}^{k,new}$ to the subspace
 133 $\mathcal{L}_k^\perp$. Since $\mathbf{e}^{k,new}$ is orthogonal to $\mathcal{L}_k$, the distance based on $\mathbf{e}^{k,new}$ is called
 the orthogonal distance. An illustration of OD_k^2 in a 3-dimensional feature

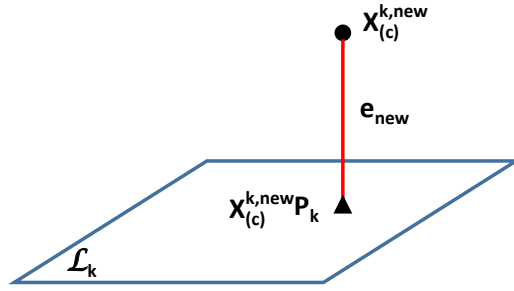


Figure 1: An illustration of OD_k^2 in a 3-dimensional feature space.

134
 135 space is shown in Figure 1. The new instance $\mathbf{x}_{(c)}^{k,new}$ is shown as the black
 136 dot; the class subspace $\mathcal{L}_k$ is shown as the dark blue 2-dimensional plane;
 137 and the projection of $\mathbf{x}_{(c)}^{k,new}$ to $\mathcal{L}_k$, $\mathbf{x}_{(c)}^{k,new} \mathbf{P}_k$, is shown as the black triangle.
 138 The residual $\mathbf{e}^{k,new}$ is represented by the red solid line segment, which is
 139 orthogonal to the plane $\mathcal{L}_k$. The square of the length of the red line segment
 140 is OD_k^2 .

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

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

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

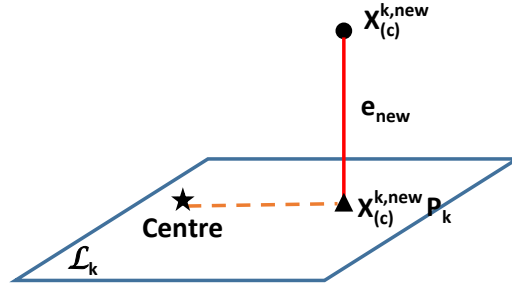


Figure 2: An illustration of SD_k^2 in a 3-dimensional feature space.

146
 147 feature space is shown in Figure 2. In addition to the symbols in Figure 1,
 148 the centre of the class subspace, $\mathcal{L}_k$, is shown as the black star, and the or-
 149 ange dashed line connects the centre of the class subspace and the projection
 150 of $\mathbf{x}_{(c)}^{k,new}$ to the class subspace. The SD_k^2 is then the reweighted length of the
 151 orange dashed line.

152 2.1.3. The classification rules

153 In NSC, the classification rule is

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

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

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

$$\gamma \left(\frac{OD_k}{c_{OD^2}^k} \right)^2 + (1 - \gamma) \left(\frac{SD_k}{c_{SD^2}^k} \right)^2, \quad (9)$$

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

163 2.2. A general formulation for the classification rules for NSM

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

$$\mathbf{x}_{(c)}^{k,new} \mathbf{M}_1^k (\mathbf{x}_{(c)}^{k,new})^T - \mathbf{t}^{k,new} \mathbf{M}_2^k (\mathbf{t}^{k,new})^T, \quad (10)$$

167 with different $\mathbf{M}_1^k \in \mathbb{R}^{p \times p}$ and $\mathbf{M}_2^k \in \mathbb{R}^{r_k \times r_k}$. In this section, we derive this
 168 general formulation based on the classification rules (8) and (9), and show
 169 $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ for (8) and (9), respectively. Based on the derived general
 170 formulation of the classification rules, we will define the distance to subspace
 171 and propose a method to learn the distance to subspace in the next section.

Substituting (6) into (5), we obtain

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

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

Then the classification rule (8) can be written as

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

176 where $\mathbf{M}_{1(NSC)}^k = \mathbf{I}_p$ and $\mathbf{M}_{2(NSC)}^k = \mathbf{I}_{r_k}$. Equation (12) indicates that
177 the classification rule of NSC provides equal weights to the p dimensions
178 in the linear combination of the original features $\mathbf{x}_{(c)}^{k,new} (\mathbf{x}_{(c)}^{k,new})^T$ and also
179 equal weights to the r_k dimensions in the linear combination of the scores
180 $\mathbf{t}^{k,new} (\mathbf{t}^{k,new})^T$.

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

$$\begin{aligned}
& \frac{\gamma}{(c_{\text{OD}^2}^k)^2} (\mathbf{x}_{(c)}^{k,\text{new}} (\mathbf{x}_{(c)}^{k,\text{new}})^T - \mathbf{t}^{k,\text{new}} (\mathbf{t}^{k,\text{new}})^T) + \frac{1-\gamma}{(c_{\text{SD}^2}^k)^2} \mathbf{t}^{k,\text{new}} \mathbf{D}_r^{-2} (\mathbf{t}^{k,\text{new}})^T \\
&= \frac{\gamma}{(c_{\text{OD}^2}^k)^2} \mathbf{x}_{(c)}^{k,\text{new}} (\mathbf{x}_{(c)}^{k,\text{new}})^T - \sum_{i=1}^r \left(-\frac{1-\gamma}{(c_{\text{SD}^2}^k)^2} + \frac{\gamma}{(c_{\text{OD}^2}^k)^2 d_i^2} \right) t_i^2 \\
&= \mathbf{x}_{(c)}^{k,\text{new}} \mathbf{M}_{1(S)}^k (\mathbf{x}_{(c)}^{k,\text{new}})^T - \mathbf{t}^{k,\text{new}} \mathbf{M}_{2(S)}^k (\mathbf{t}^{k,\text{new}})^T, \tag{13}
\end{aligned}$$

181 where $\mathbf{M}_{1(S)}^k = \frac{1}{h_1} \mathbf{I}_p$, $h_1 = \frac{\gamma}{(c_{\text{OD}^2}^k)^2}$ and $\mathbf{M}_{2(S)}^k$ is an r_k -by- r_k diagonal matrix
182 with $(-\frac{1-\gamma}{(c_{\text{SD}^2}^k)^2} + \frac{\gamma}{(c_{\text{OD}^2}^k)^2 d_i^2})$ on the diagonals (d_i 's are the singular values in
183 $\mathbf{D}$ with $d_1 \geq d_2 \geq \dots \geq d_{r_k} \geq 0$). Different from the classification rule of
184 NSM in (12), the rule in (13) indicates that the classification rule of SIMCA
185 provides equal weights to the p dimensions in the linear combination of the
186 the original features $\mathbf{x}_{(c)}^{k,\text{new}} (\mathbf{x}_{(c)}^{k,\text{new}})^T$, while providing different weights to the
187 r_k dimensions in the linear combination of the scores $\mathbf{t}^{k,\text{new}} (\mathbf{t}^{k,\text{new}})^T$.

188 2.3. Learning distance to subspace

189 We define the general formulation (10) as the distance from $\mathbf{x}_{\text{new}}$ to the
190 k th class subspace. Hence we assign $\mathbf{x}_{\text{new}}$ to the nearest class subspace based
191 on the distance to subspace defined in (10).

192 The distance to subspace for the k th class defined in (10) depends on
193 two matrices: $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$. It can be treated as the difference between two
194 squared distances: $\mathbf{x}_{(c)}^{k,\text{new}} \mathbf{M}_1^k (\mathbf{x}_{(c)}^{k,\text{new}})^T$ is the squared distance from $\mathbf{x}_{(c)}^{k,\text{new}}$
195 to the centre of the class subspace $\mathcal{L}_k$, and $\mathbf{t}^{k,\text{new}} \mathbf{M}_2^k (\mathbf{t}^{k,\text{new}})^T$ is the squared
196 distance from the projection of $\mathbf{x}_{(c)}^{k,\text{new}}$ to $\mathcal{L}_k$ to the centre of $\mathcal{L}_k$.

197 The matrices $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ are of great importance for classification.
198 Instead of determining $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ 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 $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$, 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 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ in $\mathbb{R}^{1 \times p}$ with a set of labels $\{y_1, y_2, \dots, y_N\}$,

the distance metric $d(\mathbf{x}_i, \mathbf{x}_j)$ between two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ should satisfy the following properties:

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

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

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

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

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

2.3.2. Distance to subspace

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

$$d^2(\mathbf{x}, \mathcal{L}_k) = \mathbf{x}_{(c)}^k \mathbf{M}_1^k (\mathbf{x}_{(c)}^k)^T - \mathbf{t}^k \mathbf{M}_2^k (\mathbf{t}^k)^T, \quad (15)$$

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

252 2.3.3. Learned distance to subspace

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

$$258 \quad \mathbf{S} = \{(\mathbf{x}_i, \mathcal{L}_k) \mid \mathbf{x}_i \text{ belongs to class } k\}, \text{ and}$$

$$259 \quad \mathbf{D} = \{(\mathbf{x}_i, \mathcal{L}_k) \mid \mathbf{x}_i \text{ does not belong to class } k\}.$$

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

$$265 \quad \mathbf{S} = \{(\mathbf{x}_{1(i)}, \mathcal{L}_1), (\mathbf{x}_{2(j)}, \mathcal{L}_2) \mid i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_2\}, \text{ and}$$

$$266 \quad \mathbf{D} = \{(\mathbf{x}_{1(i)}, \mathcal{L}_2), (\mathbf{x}_{2(j)}, \mathcal{L}_1) \mid i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_2\}.$$

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 $\mathbf{S}$, while maximise the sum of those that fall into the dissimilarity set $\mathbf{D}$. 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:

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

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:

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

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

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

$$\mathbf{M}_1^k \succeq 0 \text{ and } \mathbf{M}_2^k \succeq 0, \quad (20)$$

where $\mathbf{M}_1^k \succeq 0$ and $\mathbf{M}_2^k \succeq 0$ denote that $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ are positive semidefi-

nite. The constraints in (18) and (19) can be rewritten as

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

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

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

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

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

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

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

277 Considering the nature of spectral data, i.e. high-dimensional feature and

278 small sample size, learning the full matrices, $\mathbf{M}_1^k$ with $p(p+1)/2$ parameters
 279 and $\mathbf{M}_2^k$ with $r_k(r_k+1)/2$ parameters, could easily suffer from the overfitting
 280 problem. In (12) and (13), $\mathbf{M}_{1(NSC)}^k = \mathbf{I}_p$ and $\mathbf{M}_{1(S)}^k = \frac{1}{h_1}\mathbf{I}_p$ are identity
 281 matrices with common coefficients 1 and $1/h_1$ for all dimensions, respectively.
 282 Therefore, in this paper, we learn $\mathbf{M}_1^k = c_k\mathbf{I}_p$ (with $c_k \geq 0$) and $\mathbf{M}_2^k =$
 283 $\text{diag}(m_{21}^k, m_{22}^k, \dots, m_{2r_k}^k)$ (with each element nonnegative), as natural and
 284 practically-interpretable extensions of those used in (12) and (13).

285 3. Experiments

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

293 3.1. Datasets

294 The number of samples in each class and the number of features for the
 295 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

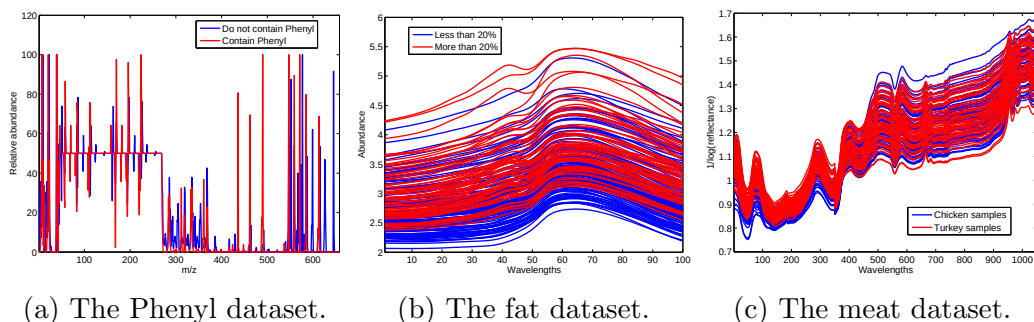


Figure 3: The plots of the spectra of the three datasets.

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

337 mean classification error is chosen as the number of PCs.

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

342 In NSM-LD2S, the optimisation problem (21) is solved by ‘cvx’ in MAT-
343 LAB.

344 In SVM, the radial basis function (RBF) kernel is adopted. The scale
345 parameter of the RBF kernel and the penalty factor C are tune by 5-fold
346 cross-validation. The values of the two parameters to be chosen are set to
347 10, 10^2 and 10^3 . In k NN, the number of nearest neighbours is tuned by 5-
348 fold cross-validation. The values to be chosen are set to 3, 5 and 7. In NB,
349 the prior probability of each class is set as the proportion of the number of
350 training samples of that class over the total number of training samples.

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

354 3.3. Results

355 3.3.1. The Phenyl dataset

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

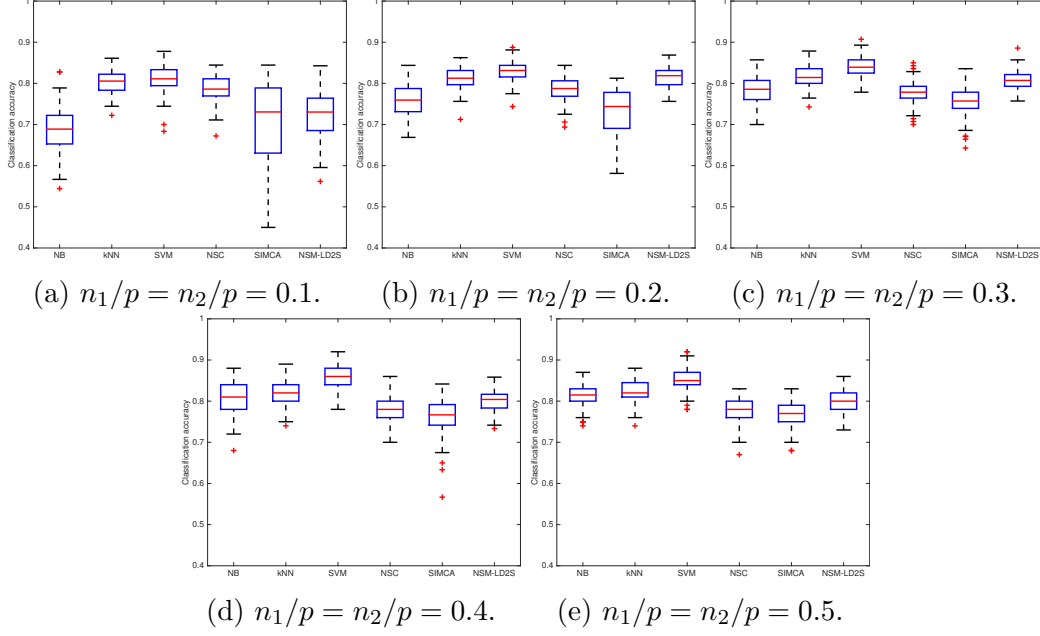


Figure 4: Classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the Phenyl dataset.

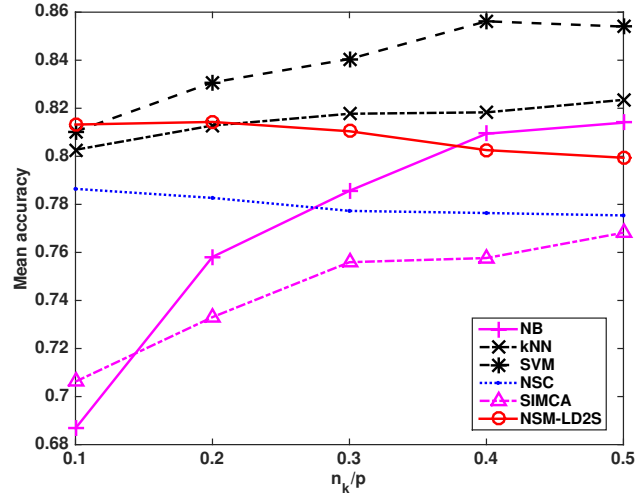


Figure 5: Mean classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the Phenyl dataset.

362 However, it is conceivable that, for certain other datasets, the classifica-
 363 tion performance of NSM-LD2S cannot always be better than those of NSC
 364 and SIMCA, in particular under small n_k/p ratios. In the following two
 365 sections, we show two examples that NSM-LD2S performs worse than NSC
 366 and SIMCA for small n_k/p ratios but better for large n_k/p ratios. This is
 367 because there are more parameters in NSM-LD2S to be learned than in NSC
 368 and SIMCA, and NSM-LD2S needs more training samples to achieve good
 369 classification performance for some data. In addition, the classification per-
 370 formances of NB, k NN and SVM are also not always better than the nearest
 371 subspace methods. The following two examples can also demonstrate this
 372 argument.

373 3.3.2. *The fat dataset*

374 In the fat dataset, the classification performance of NSM-LD2S and SIMCA
 375 are worse than NSC when $n_k/p = 0.1$ and are better than NSC when
 376 $n_k/p \geq 0.2$, as shown in Figure 6 and Figure 7. NSM-LD2S provides the
 377 best classification performance when $n_k/p \geq 0.2$.

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

382 3.3.3. *The meat dataset*

383 Compared with the fat dataset, the classification accuracies of the three
 384 methods for the meat dataset show a stronger effect of the n_k/p ratios. When
 385 $n_k/p < 0.4$, NSM-LD2S performs much worse than NSC and SIMCA, espe-

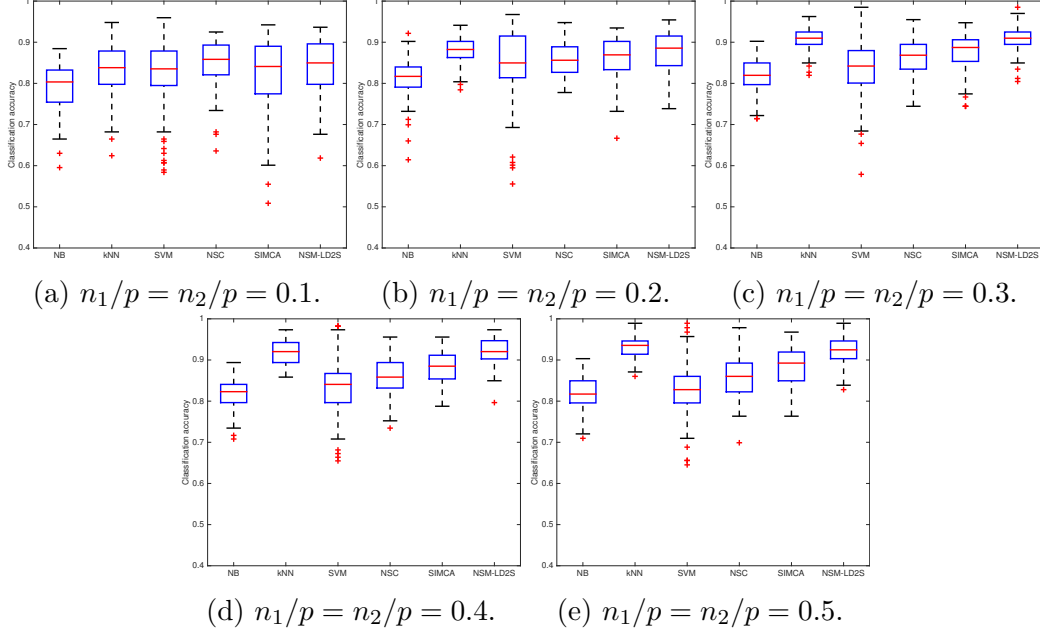


Figure 6: Classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the fat dataset.

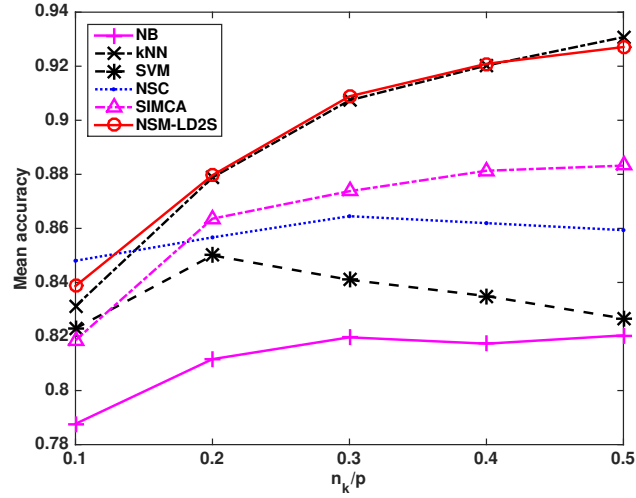


Figure 7: Mean classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the fat dataset.

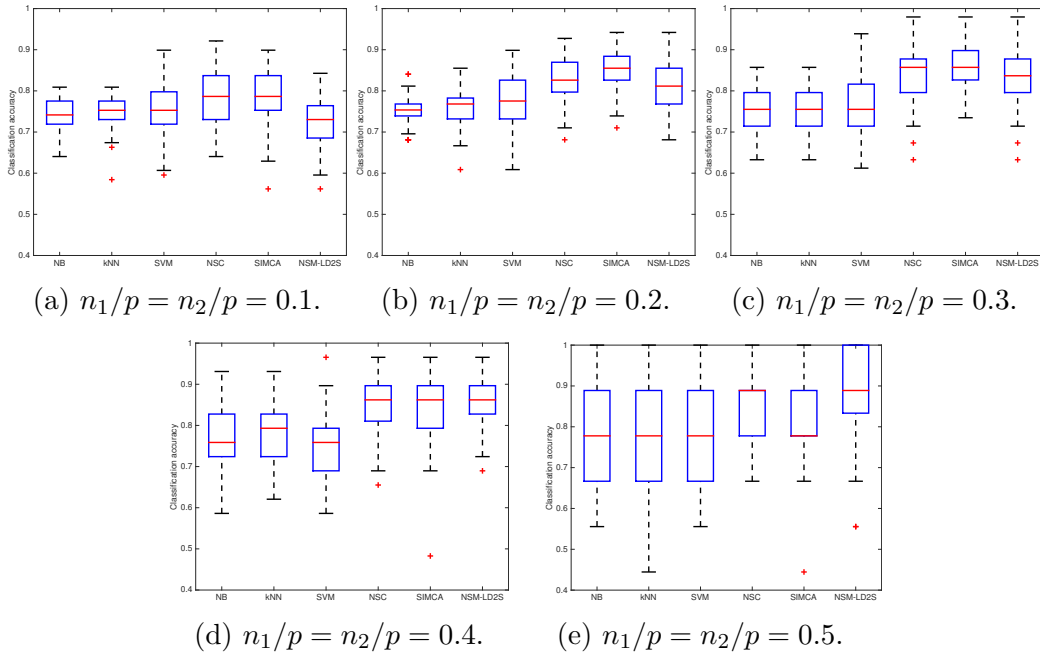


Figure 8: Classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the meat dataset.

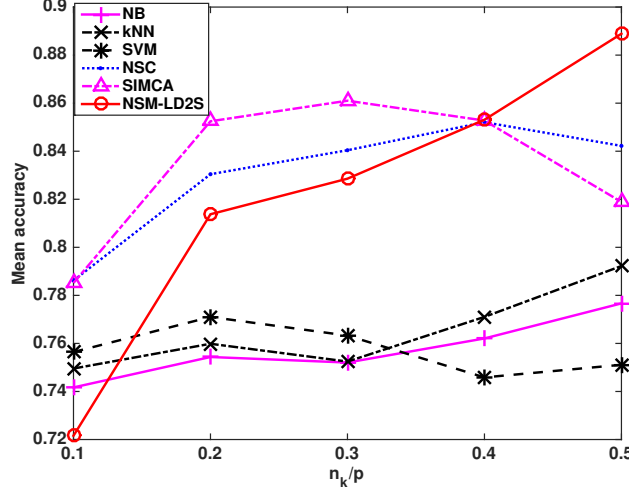


Figure 9: Mean classification accuracies of NB, k NN, SVM, NSC, SIMCA and NSM-LD2S for the meat dataset.

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. k NN 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

400 all the parameters in LD2S.

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

412 4. Conclusion

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

422 Acknowledgement

423 The authors thank the reviewers for their constructive comments.

424 References

- 425 [1] T. Arnalds, J. McElhinney, T. Fearn, G. Downey, A hierarchical dis-
426 criminant analysis for species identification in raw meat by visible and
427 near infrared spectroscopy, *Journal of Near Infrared Spectroscopy* 12 (3)
428 (2004) 183–188.
- 429 [2] K. V. Branden, M. Hubert, Robust classification in high dimensions
430 based on the SIMCA method, *Chemometrics and Intelligent Laboratory*
431 *Systems* 79 (1) (2005) 10–21.
- 432 [3] Y. Chi, Nearest subspace classification with missing data, in: *Signals,*
433 *Systems and Computers, 2013 Asilomar Conference on, IEEE, 2013, pp.*
434 *1667–1671.*
- 435 [4] Y. Chi, F. Porikli, Connecting the dots in multi-class classification:
436 From nearest subspace to collaborative representation, in: *Computer*
437 *Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on,*
438 *IEEE, 2012, pp. 3602–3609.*
- 439 [5] C. Durante, R. Bro, M. Cocchi, A classification tool for N-way array
440 based on SIMCA methodology, *Chemometrics and Intelligent Labora-*
441 *tory Systems* 106 (1) (2011) 73–85.
- 442 [6] F. Ferraty, P. Vieu, *Nonparametric Functional Data Analysis: Theory*
443 *and Practice, Springer Science & Business Media, 2006.*

- 444 [7] K. Fukui, A. Maki, Difference subspace and its generalization for
445 subspace-based methods, *IEEE Transactions on Pattern Analysis and*
446 *Machine Intelligence* 37 (11) (2015) 2164–2177.
- 447 [8] P. Hall, D. M. Titterington, J.-H. Xue, Median-based classifiers for
448 high-dimensional data, *Journal of the American Statistical Association*
449 104 (488) (2009) 1597–1608.
- 450 [9] P. Hall, J.-H. Xue, Incorporating prior probabilities into high-
451 dimensional classifiers, *Biometrika* 97 (1) (2010) 31–48.
- 452 [10] P. Hall, J.-H. Xue, On selecting interacting features from high-
453 dimensional data, *Computational Statistics & Data Analysis* 71 (2014)
454 694–708.
- 455 [11] K.-C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face
456 recognition under variable lighting, *IEEE Transactions on Pattern Anal-*
457 *ysis and Machine Intelligence* 27 (5) (2005) 684–698.
- 458 [12] C. Mees, F. Souard, C. Delporte, E. Deconinck, P. Stoffelen, C. Stévigny,
459 J.-M. Kauffmann, K. De Braekeleer, Identification of coffee leaves using
460 FT-NIR spectroscopy and SIMCA, *Talanta* 177 (2018) 4–11.
- 461 [13] J.-X. Mi, D.-S. Huang, B. Wang, X. Zhu, The nearest-farthest subspace
462 classification for face recognition, *Neurocomputing* 113 (2013) 241–250.
- 463 [14] B. Mnassri, B. Ananou, M. Ouladsine, et al., Fault detection and di-
464 agnosis based on PCA and a new contribution plot, *IFAC Proceedings*
465 *Volumes* 42 (8) (2009) 834–839.

- 466 [15] B. Mnassri, M. Ouladsine, et al., Reconstruction-based contribution ap-
 467 proaches for improved fault diagnosis using principal component analy-
 468 sis, *Journal of Process Control* 33 (2015) 60–76.
- 469 [16] I. Nejadgholi, M. Bolic, A comparative study of PCA, SIMCA and Cole
 470 model for classification of bioimpedance spectroscopy measurements,
 471 *Computers in biology and medicine* 63 (2015) 42–51.
- 472 [17] M. Rafferty, X. Liu, D. M. Lavery, S. McLoone, Real-time multi-
 473 ple event detection and classification using moving window pca, *IEEE*
 474 *Transactions on Smart Grid* 7 (5) (2016) 2537–2548.
- 475 [18] A. Sgarbossa, C. Costa, P. Menesatti, F. Antonucci, F. Pallottino,
 476 M. Zanetti, S. Grigolato, R. Cavalli, A multivariate SIMCA index as
 477 discriminant in wood pellet quality assessment, *Renewable Energy* 76
 478 (2015) 258–263.
- 479 [19] Q. Tian, S. Chen, L. Qiao, Ordinal margin metric learning and its exten-
 480 sion for cross-distribution image data, *Information Sciences* 349 (2016)
 481 50–64.
- 482 [20] P. Van den Kerkhof, J. Vanlaer, G. Gins, J. F. Van Impe, Analysis
 483 of smearing-out in contribution plot based fault isolation for statistical
 484 process control, *Chemical Engineering Science* 104 (2013) 285–293.
- 485 [21] K. Q. Weinberger, L. K. Saul, Distance metric learning for large margin
 486 nearest neighbor classification, *Journal of Machine Learning Research*
 487 10 (2009) 207–244.

- 488 [22] S. Wold, Pattern recognition by means of disjoint principal components
489 models, *Pattern Recognition* 8 (3) (1976) 127–139.
- 490 [23] E. P. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning
491 with application to clustering with side-information, *Advances in Neural
492 Information Processing Systems* 15 (2003) 505–512.
- 493 [24] J. Yu, D. Tao, J. Li, J. Cheng, Semantic preserving distance metric
494 learning and applications, *Information Sciences* 281 (2014) 674–686.
- 495 [25] L. Zhang, W.-D. Zhou, B. Liu, Nonlinear nearest subspace classifier, in:
496 *International Conference on Neural Information Processing*, Springer,
497 2011, pp. 638–645.
- 498 [26] P. Zhu, Q. Hu, W. Zuo, M. Yang, Multi-granularity distance metric
499 learning via neighborhood granule margin maximization, *Information
500 Sciences* 282 (2014) 321–331.
- 501 [27] R. Zhu, K. Fukui, J.-H. Xue, Building a discriminatively ordered sub-
502 space on the generating matrix to classify high-dimensional spectral
503 data, *Information Sciences* 382 (2017) 1–14.
- 504 [28] R. Zhu, J.-H. Xue, On the orthogonal distance to class subspaces for
505 high-dimensional data classification, *Information Sciences* 417 (2017)
506 262–273.