Reinforcing Soft Independent Modelling of Class Analogy (SIMCA)

Rui Zhu

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Statistical Science

University College London

May 3, 2017

I, Rui Zhu, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Soft independent modelling of class analogy (SIMCA) is a widely used subspace-based classification technique for spectral data analysis. The principal component (PC) subspace is built for each class separately through principal components analysis (PCA). The squared orthogonal distance (OD^2) between the test sample and the class subspace of each class, and the squared score distance (SD^2) between the projection of the test sample to the class subspace and the centre of the class subspace, are usually used in the classification rule of SIMCA to classify the test sample.

Although it is commonly used to classify high-dimensional spectral data, SIMCA suffers from several drawbacks and some misleading calculations in literature. First, modelling classes separately makes the discriminative between-class information neglected. Second, the literature of SIMCA fail to explore the potential benefit of using geometric convex class models, whose superior classification performance has been demonstrated in face recognition. Third, based on our experiments on several real datasets, calculating OD^2 using the formulae in a highly-cited SIMCA paper (De Maesschalck et al., 1999) results in worse classification performance than using those in the original SIMCA paper (Wold, 1976) for some high-dimensional data and provides misleading classification results. Fourth, the distance metrics used in the classification rule of SIMCA are predetermined, which are not adapted to different data.

Hence the research objectives of my PhD work are to reinforce SIMCA from the following four perspectives: O1) to make its feature space more discriminative; O2) to use geometric convex models as class models in SIMCA for spectral data classification and to study the classification mechanism of classification using different class models; O3) to investigate the equality and inequality of the calculations of OD^2 in De Maesschalck et al. (1999) and Wold (1976) for low-dimensional and high-dimensional scenarios; and O4) to make its distance metric adaptively learned from data. In this thesis, we present four contributions to achieve the above four objectives, respectively:

First, to achieve O1), we propose to first project the original data to a more discriminative subspace before applying SIMCA. To build such discriminative subspace, we propose the discriminatively ordered subspace (DOS) method, which selects the eigenvectors of the generating matrix with high discriminative ability between classes to span DOS. A paper of this work, “Building a discriminatively ordered subspace on the generating matrix to classify high-dimensional spectral data”, has been recently published by the journal of “Information Sciences”.

Second, to achieve O2), we use the geometric convex models, convex hull and convex cone, as class models in SIMCA to classify spectral data. We study the dual of classification methods using three class models: the PC subspace, convex hull and convex cone, to investigate their classification mechanism. We provide theoretical results of the dual analysis, establish a separating hyperplane classification (SHC) framework and provide a new data exploration scheme to analyse the properties of a dataset and why such properties make one or more of the methods suitable for the data.

Third, to achieve O3), we compare the calculations of OD^2 in De Maesschalck et al. (1999) and Wold (1976). We show that the corresponding formulae in the two papers are equivalent, only when the training data of one class have more samples than features. When the training data of one class have more features than samples (i.e. high-dimensional), the formulae in De Maesschalck et al. (1999) are not precise and affect the classification results. Hence we suggest to use the formulae in Wold (1976) to calculate OD^2, to get correct classification results of SIMCA for high-dimensional data.

Fourth, to achieve O4), we learn the distance metrics in SIMCA based on the derivation of a general formulation of the classification rules used in literature. We define the general formulation as the distance metric from a sample to a class subspace. We propose the method of learning distance to subspace to learn this distance metric by making the samples to be closer to their correct class subspaces while be farther away from their wrong class subspaces.

Lastly, at the end of this thesis we append two pieces of work on hyperspectral image analysis. First, the joint paper with Mr Mingzhi Dong and Dr Jing-Hao Xue, "Spectral Nonlocal Restoration of Hyperspectral Images with Low-Rank Property", has been published by the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. Second, the joint paper with Dr Fei Zhou and Dr Jing-Hao Xue, "MvSSIM: A Quality Assessment Index for Hyperspectral Images", has been in revision for Neurocomputing. As these two papers do not focus on the research objectives of this thesis, they are appended as some additional work during my PhD study.

Acknowledgements

I would like to express my deepest and sincere gratitude to my supervisor Dr Jing-Hao Xue, for all his support and encouragement during my PhD study. Without his guidance this PhD thesis would not have been possible. I feel very lucky to have worked with him in the past three and a half years. I learned a tremendous amount from him, not only from his knowledge and expertise in statistical machine learning, but also from his attention to detail and his way of working with others.

I would also like to thank my second supervisor Professor Tom Fearn. His topic on "classification of NIR spectral data" for my MSc project first brought me into the area of statistical machine learning. My sincere thanks also goes to Professor Kazuhiro Fukui, for his insightful comments on my work with him and his warm host of my stay in Japan. I also greatly appreciate the helpful discussions with my other collaborators, Ziyu Wang, Mingzhi Dong and Dr Fei Zhou.

I gratefully acknowledge the funding sources that supported my PhD study. I was funded by the Teaching Assistant Scholarship from the Department of Statistical Science and the Dean's prize from the Faculty of Mathematical & Physical Sciences, UCL.

I thank my friends and flatmates, Ying Zhang, Xiaoke Liu, Xiaoyu Liu and Ken Liang, for the happy time we have lived together. Special thanks to Ran Wang for all her support and the great time we have shared.

Lastly, I would like to thank my dearest parents for all their love, understanding, support and encouragement throughout my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 SIMCA

High-dimensional spectral data, such as near infrared (NIR) spectroscopic data and mass spectrometry (MS) data, are widely used in a variety of fields, for example chemometrics, bioinformatics and hyperspectral image analysis. In the analysis of spectral data, classification is an omnipresent task (Downey, 1994; Pan et al., 2003; Berrueta et al., 2007; Roggo et al., 2007; Zhang et al., 2012; Holloway et al., 2014), which enables us to distinguish different species, identify the geographical origins of the products, or predict molecular substructure, to name a few.

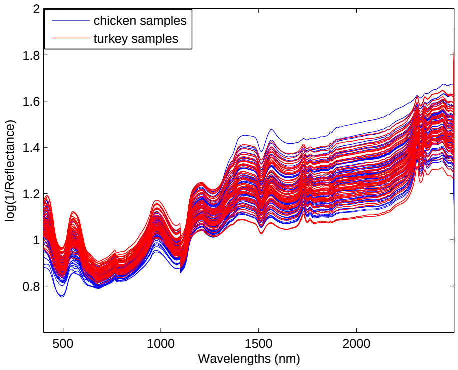


Figure 1.1: Spectra of meat samples from two classes: chicken and turkey.

Figure 1.1 shows an example for NIR spectroscopic data of two classes, the chicken meat samples and the turkey meat samples, which is further analysed in Chapter 2. Each curve depicts the spectrum of a sample, which is usually represented by a high-dimensional feature vector. A classification task is to classify the spectra of new samples into the two classes based on the information provided by some labelled training spectra. In this thesis, we focus on two-class classification for high-dimensional spectral data. Based on the two-class classification results, multi-class classification can be readily obtained by using the one-vs-one or one-vs-all strategy (Bishop, 2006).

Soft independent modelling of class analogy (SIMCA) (Wold, 1976) is a subspace-based classification method that is widely used in the classification of high-dimensional spectral data in chemometrics (Downey, 1994; Berrueta et al., 2007; Roggo et al., 2007; Fujimoto and Tsuchikawa, 2010; Li et al., 2014; Davis et al., 2015; Jaiswal et al., 2016; Li et al., 2016; Márquez et al., 2016; Srivastava et al., 2016; Wang et al., 2016; Basri et al., 2017). Fujimoto and Tsuchikawa (2010) studied the identification of dead and sound knots based on SIMCA. Li et al. (2014) applied SIMCA for Chinese liquor discrimination. Davis et al. (2015) applied SIMCA for textile classification. Jaiswal et al. (2016) used SIMCA to detect aflatoxin M1 in milk. Li et al. (2016) applied SIMCA for the identification of pummelo cultivars. Márquez et al. (2016) applied SIMCA for a hazelnut adulteration problem to classify unadulterated and adulterated with almond classes. Srivastava et al. (2016) discriminated between dextrose and substitutes by using SIMCA. Wang et al. (2016) discriminated bamboo species by using SIMCA. Basri et al. (2017) used SIMCA to classify pure and adulterated palm oil.

SIMCA consists of two phases when it is used for two-class classification. First, in the training phase, two class models are built for the two classes separately. The class models in SIMCA are represented using principal component (PC) class subspaces through using principal component analysis (PCA). Second, in the test phase, a new sample is classified using a classification rule based on its distances to the two class subspaces. Usually two distances are used in the classification rule: the squared orthogonal distance (OD^2) between the new sample and the class subspace and the squared score distance (SD^2) between the projection of the new sample to the class subspace and the centre of that class subspace. When Wold (1976) first proposed SIMCA, only OD^2 was used in the classification rule. Recently, a linear combination of OD^2 and SD^2 has been widely adopted as the classification rule (Berrueta et al., 2007).

SIMCA is designed for a 'soft' assignment of a new sample, which means that a new sample can be assigned to one of the known classes, both of the known classes and none of the classes. Thus SIMCA can be used as a classifier as well as an outlier detector. In this thesis, we treat SIMCA as a simple classification method that classifies a new sample to only one of the known classes, to obtain non ambiguous classification results as suggested in Berrueta et al. (2007).

1.2 Limitations of SIMCA

In spite of its wide use, SIMCA suffers from the following four limitations. The first two limitations are related to the class models built in the training phase of SIMCA, and the last two limitations are related to the distances used in the classification rule in the test phase of SIMCA.

Limitation 1 Since the PC class subspaces are built independently in SIMCA, the discriminative between-class information is neglected during this process. Therefore the classification rule calculated independently for each class may not be discriminative enough to classify a new sample.

Limitation 2 Besides PC subspaces, geometric convex models, such as convex hull and convex cone, have also been used as class models and have shown better classification results compared with PC subspaces for other classification tasks, such as face recognition. However, to the best of our knowledge, the literature of SIMCA have barely explored such potentially beneficial changing of class models for better classification of spectral data.

Limitation 3 We notice that the calculation of OD^2 in the highly-cited SIMCA paper (De Maesschalck et al., 1999) is different from that in the original SIMCA paper (Wold, 1976). Using the formulae in De Maesschalck et al. (1999) results in worse classification performance than using those in Wold (1976) for some high-dimensional data and provides misleading classification results.

Limitation 4 The distance metrics used in the classification rule of SIMCA are predefined: the Euclidean distance metric is used in OD^2 and the Mahalanobis distance metric is used in SD^2. However, different data usually prefer different distance metrics and the predefined distance metrics in SIMCA should be adapted to different data.

1.3 Our contributions to SIMCA

In this thesis, we present our contributions to overcome the above four limitations in two parts: Part I presents two contributions to the class models used in SIMCA to overcome Limitation 1 and Limitation 2, respectively, and Part II presents two contributions to the distances used in SIMCA to overcome Limitation 3 and Limitation 4, respectively. We introduce the four contributions briefly as follows.

Contribution 1: Building a discriminatively ordered subspace on the generating matrix To overcome Limitation 1, we aim to make the feature space more discriminative. An appealing remedy is to first project the original data to a more discriminative subspace before applying SIMCA. For this, generalised difference subspace (GDS) that explores the information between class subspaces in the generating matrix can be a strong candidate. However, due to the difference between a class subspace (of infinite scale) and a class (of finite scale), the eigenvectors selected by GDS may not also be discriminative for classifying samples of classes. Therefore in this contribution, we propose a discriminatively ordered subspace (DOS): different from GDS, our DOS selects the eigenvectors with high discriminative ability between classes rather than between class subspaces. The experiments on three real spectral datasets demonstrate that applying DOS before SIMCA outperform its counterparts.

Contribution 2: Dual of nearest-class-model methods: a separating hyperplane classification framework To overcome Limitation 2, we use two geometric convex models, convex hull and convex cone, as class models in SIMCA to classify spectral data. We also aim to investigate the classification mechanism for the classification methods with three class models: the PC subspace, convex hull and convex cone. To make the investigation straightforward, we use OD as the classification rule. Also, to avoid confusion with SIMCA, we name the classification methods studied in this contribution as nearest class-model-based methods. We first propose the nearest convex cone method (NCCM) to fill the gap between two existing methods, the nearest subspace method (NSM) and the nearest convex hull method (NCHM). NSM is equivalent to SIMCA using OD as the classification rule (SIMCA-OD); NCCM is equivalent to SIMCA-OD using convex cones as class models; and NCHM is equivalent to SIMCA-OD using convex hulls as class models. Then we investigate NSM, NCHM and NCCM both theoretically and empirically, to understand deeply their underlying classification mechanism and analyse their data-dependent classification performances. Theoretically, we provide results of the dual analysis of NSM, NCHM and NCCM, and establish a separating hyperplane classification (SHC) framework for the nearest-class-model methods. Empirically, we provide a new data exploration scheme to analyse the properties of a dataset and why such properties make one or more of the methods suitable for the data.

Contribution 3: On the orthogonal distance of SIMCA for high-dimensional data To overcome Limitation 3, we investigate the equality and inequality of the calculations of OD^2 in De Maesschalck et al. (1999) and Wold (1976) for low-dimensional and high-dimensional scenarios. We show that only when the training data of a class have more samples than features, the corresponding formulae in the two papers are equivalent. When the training data of a class are of high dimension (i.e., when the number of features is larger than the number of samples), the formulae in De Maesschalck et al. (1999) are not precise. Hence, we suggest that the calculation of OD^2 should follow the original definition in Wold (1976), in order to obtain a correct decision of SIMCA for classification of high-dimensional data, which are now common in practice.

Contribution 4: Learning distance to subspace To overcome Limitation 4, we aim to find good distance metrics for the classification rule of SIMCA to improve its classification performance using distance metric learning methods. However, different from those in distance metric learning methods that measure the distances between samples, the distance metrics in SIMCA measure the distances between samples and class subspaces. To adapt the distance metric learning methods to learn the distance metrics in SIMCA, we first derive a general formulation for the classification rules of SIMCA used in literature and define it as the distance to subspace. We show that the distance to subspace is dependent on two parameterisation matrices and propose a method of learning distance to subspace to learn those matrices. We term the learned distance metrics as "learned distance to subspace (LD2S)". LD2S is based on the following set of similarity/dissimilarity constraints: the samples are similar to their correct class subspaces while are dissimilar to the wrong class subspaces. LD2S aims to make the samples to be closer to their correct class subspaces while being farther away from their wrong class subspaces. The superior classification performance of using LD2S in the classification rule on one real spectral dataset has demonstrated the effectiveness of LD2S.

1.4 The structure of the thesis

The thesis is organised as follows. In Chapter 2, we present Contribution 1 to make the feature space more discriminative. In Chapter 3, we present Contribution 2 to analyse the classification mechanism of nearest-class-model methods. In Chapter 4, we present Contribution 3 to investigate the difference of calculating OD^2 using the formulae in De Maesschalck et al. (1999) and those in Wold (1976). In Chapter 5, we present Contribution 4 to learn good distance metrics for the classification rule used in SIMCA. In Chapter 6 we present some concluding remarks and future work for reinforcing SIMCA. The structure of the thesis is illustrated in Figure 1.2.

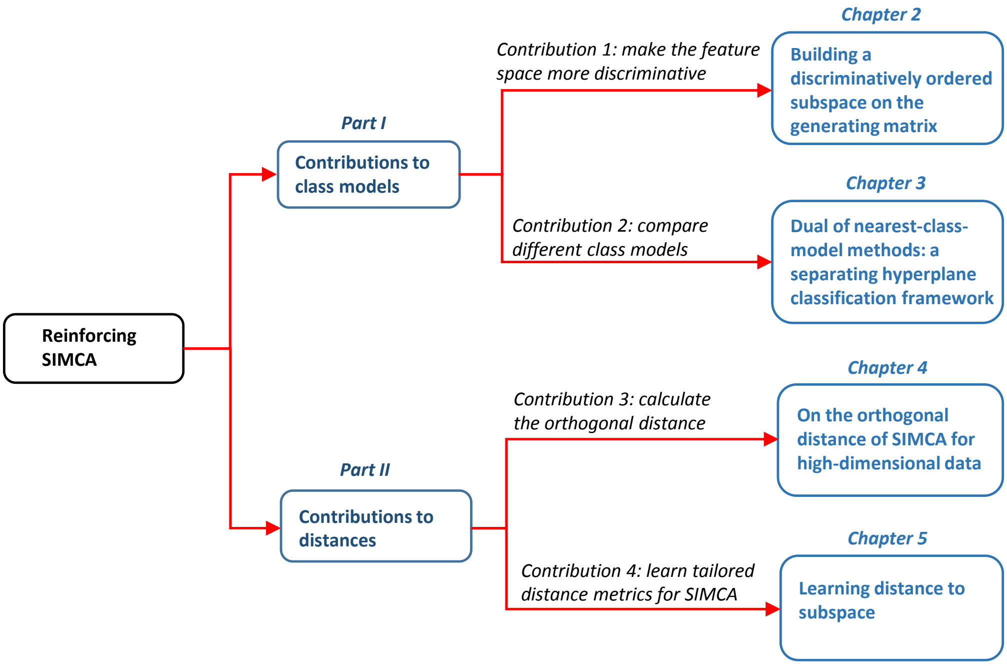

Reinforcing SIMCA
Part I
Contributions to class models
Part II
Contributions to distances
Contribution 1: make the feature space more discriminative
Contribution 2: compare different class models
Contribution 3: calculate the orthogonal distance
Contribution 4: learn tailored distance metrics for SIMCA
Chapter 2
Building a discriminatively ordered subspace on the generating matrix
Chapter 3
Dual of nearest-class-model methods: a separating hyperplane classification framework
Chapter 4
On the orthogonal distance of SIMCA for high-dimensional data
Chapter 5
Learning distance to subspace

Figure 1.2: The structure of the thesis.

Part I

Contributions to the class models used in SIMCA

Part I presents two of our contributions to the class models used in SIMCA. In this part, we focus on studying the class models used in the training phase of SIMCA and use the classification rules based on OD^2, such as the classification rule in Wold (1976). The classification results of using other classification rules based on both OD^2 and SD^2 can be easily obtained by replacing the classification rules used in this chapter with those required.

We present the two contributions in Chapter 2 and Chapter 3, respectively. First, in Chapter 3 we aim to solve the problem of ignoring the discriminative between-class information when building the class models by making the feature space more discriminative. We propose to first project the original data to a more discriminative subspace, the discriminatively ordered subspace (DOS), before applying SIMCA. The content of Chapter 2 is based on our recently published paper (Zhu et al., 2017). Second, in Chapter 3 we aim to use geometric models as class models in SIMCA and to study the classification mechanism and the data-dependant classification performances of using different class models. We propose the separating hyperplane classification framework for the classification methods with different class models based on the dual analysis.

Chapter 2

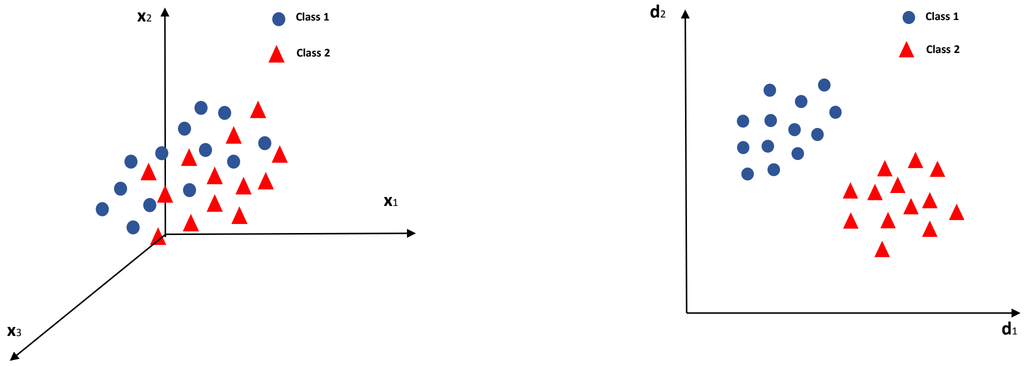
Building a discriminatively ordered subspace on the generating matrix

When SIMCA is used for two-class classification, firstly two class subspaces are built for the two classes separately through using principal component analysis (PCA). Then a classification rule based on OD^2 and/or SD^2 is used to determine the class membership of the new sample. In this Chapter, we use the F-value proposed in Wold (1976) as the classification rule, which is based on OD^2.

Although it has been widely used for the classification of high-dimensional spectral data, SIMCA suffers from the problem that the class subspaces are built independently without considering between-class information. Therefore the F-value calculated independently for each class may not be discriminative enough to classify a new sample.

An appealing solution to this problem is to find a more discriminative subspace than the original feature space and project the data to this subspace before applying SIMCA. The projections of the samples to this discriminative subspace are expected to be more separated and can be more easily classified than those in the original feature space, as illustrated in Figure 2.1. Also, as the new subspace contains more discriminative information for classification, the F-value calculated in this subspace is expected to be more discriminative. It is therefore the objective of our work in this chapter to find such a discriminative subspace.

Recently, Fukui and Maki (2015) proposed the generalised difference subspace



(a) Original feature space.

(b) Discriminative subspace.

Figure 2.1: (a) Two classes of samples are mixed together in the original 3-dimensional feature space. (b) The same groups of samples can be well separated when they are projected to a discriminative 2-dimensional subspace.

(GDS) projection as a preprocessing method to improve a popular subspace-based classifier called mutual subspace method (MSM) in image set-based object recognition. GDS aims to tackle an issue of MSM: the class subspaces are independently generated by PCA in a class-by-class manner, and thus may not be strongly discriminative for classification. This issue is actually the same as that of SIMCA. Hence, we believe the GDS projection can also be utilised as a preprocessing method for SIMCA to improve its classification performance.

GDS is a subspace containing the information about the difference between class subspaces, and thus is supposed to be more discriminative than the original feature space. GDS is generated on the basis of a generating matrix $\boldsymbol{G}_D$, which is calculated as the sum of the projection matrices of the two class subspaces and can provide between-class information. Fukui and Maki (2015) show that the eigenvectors of $\boldsymbol{G}_D$ with small eigenvalues contain the information of difference between class subspaces while those with large eigenvalues contain the information about similarity between class subspaces. The GDS projection thus keeps only the last few eigenvectors with small eigenvalues and discards the first few eigenvectors with large eigenvalues, in order to make use of the difference information.

The GDS projection shows superior performance on face recognition and hand shape recognition problems. However, there is a limitation of the GDS. The GDS projection discards the eigenvectors of $\boldsymbol{G}_D$ with large eigenvalues because they contain similarity information between class subspaces and thus are assumed ineffective for classification. This assumption is, however, not always valid due to the conceptual difference between a class subspace (of infinite scale) and a class (of finite scale). For example, two separable classes may span the same subspace. More technically, this assumption defines similarity information by using the eigenvector directions only, without considering the distribution of the projected samples in these directions. If the projected samples of different classes in the directions of similarity (i.e. the directions with large eigenvalues of $\boldsymbol{G}_D$) are still class separable, then these directions can also be discriminative in separating classes (although not discriminative in separating class subspaces), and thus discarding them can be harmful for classification of samples.

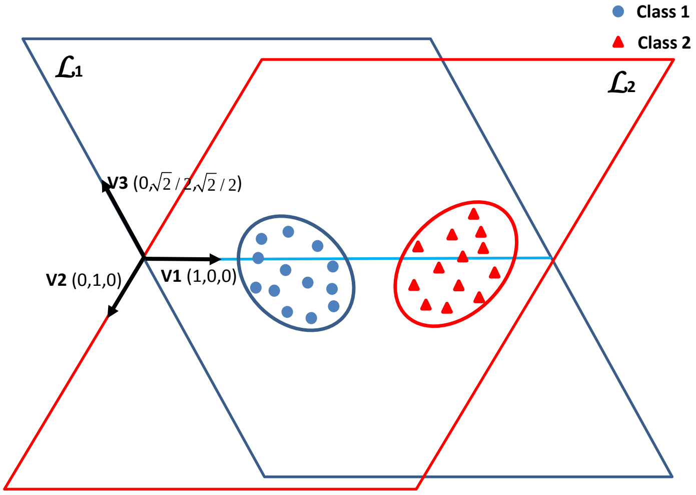
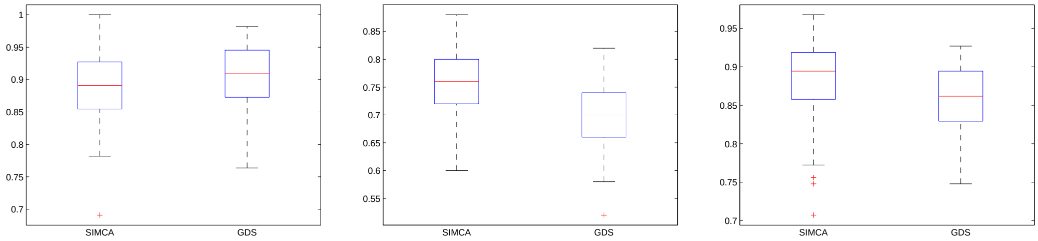


Figure 2.2: An illustrative example of the difference between a class subspace (of infinite scale) and a class (of finite scale).

To illustrate the difference between a class subspace and a class, we show an intuitive example in Figure 2.2. The infinite scale subspace of class 1, $\mathscr{L}_1$, is spanned by $\boldsymbol{v}_1$ and $\boldsymbol{v}_3$, and the infinite scale subspace of class 2, $\mathscr{L}_2$, is spanned by $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. The samples of the two classes lie in the two ellipses with finite scales in $\mathscr{L}_1$ and $\mathscr{L}_2$, respectively. It is obvious that $\boldsymbol{v}_1$ is the intersection of $\mathscr{L}_1$ and $\mathscr{L}_2$, which represents the same direction, i.e. the similarity information, between class subspaces. The GDS projection discards $\boldsymbol{v}_1$ because it is the eigenvector of $\boldsymbol{G}_D$ with the largest eigenvalue and contains similarity information between class subspaces. However, the samples of the two classes are class separable on the direction of $\boldsymbol{v}_1$, which suggests that $\boldsymbol{v}_1$ contains discriminative information between classes. (We shall demonstrate another motivating example for this issue in Section 2.1.3.1 using a real spectral dataset.)



(a) meat **(b)** Phenyl **(c)** fat

Figure 2.3: Classification accuracies of SIMCA and the GDS-preprocessed SIMCA on three real spectral datasets: meat, Phenyl and fat. In each panel, the left-hand boxplot is for SIMCA, and the right-hand boxplot is for the GDS-preprocessed SIMCA.

Moreover, here we illustrate that discarding the eigenvectors of $\boldsymbol{G}_D$ with large eigenvalues can be harmful for classification using three real spectral datasets: meat, Phenyl and fat. In Figure 2.3, we plot the classification accuracies of SIMCA and the GDS-preprocessed SIMCA on the three datasets. We can clearly observe that a preprocessing step of SIMCA by GDS does not necessarily benefit the classification performance of SIMCA; it actually has an negative effect (lowering classification accuracy) on SIMCA for the Phenyl dataset and the fat dataset. Detailed discussion on this will be provided in Section 2.2.

To make use of the between-class information in $\boldsymbol{G}_D$ and to overcome the above limitation of the GDS projection, we propose a discriminatively ordered subspace (DOS): our DOS is spanned by the most discriminative eigenvectors of $\boldsymbol{G}_D$ instead of the eigenvectors with small eigenvalues and extracts the most discriminative information from the data. That is, we sort the eigenvectors in terms of their discriminative ability and select the top-ranked eigenvectors with high discriminative abilities to generate the DOS projection. This discriminatively ordering procedure during the generation of the subspace is where the term 'discriminatively ordered' came from in DOS. As our objective is to develop DOS to tackle the issue of SIMCA, the discriminative ability of an eigenvector is measured by the classification accuracy of SIMCA on the samples projected to this eigenvector. The higher the classification accuracy, the higher the discriminative ability. We choose this filter-type of eigenvector selection scheme for high-dimensional spectral data, taking into consideration its simplicity and efficiency, as well as the uncorrelatedness and orthogonality of the candidate eigenvectors. The effectiveness of the DOS-preprocessed SIMCA will be demonstrated in Section 2.2.

The rest of this chapter is organised as follows. In Section 2.1, a discussion of the GDS projection and a detailed description of the DOS projection are provided. In Section 2.2, GDS and DOS are compared with respect to the improvement of classification performance of SIMCA on real spectral datasets. Section 2.3 presents some concluding remarks.

2.1 Methodology

2.1.1 SIMCA

In the training phase of SIMCA, suppose $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times p}$ is the training set of class k ($k = 1,2$), in which there are n_k training instances and each instance is represented by a p-dimensional data vector (i.e. in the original p-dimensional feature space). To build the principal component (PC) subspace for each class, we apply eigendecomposition to the covariance matrix of the kth class:

$$\mathrm{Cov}(\boldsymbol{X}_k) = \frac{1}{n_k - 1}(\boldsymbol{X}_{k(c)})^T \boldsymbol{X}_{k(c)} = \boldsymbol{V}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^T \ , \tag{2.1}$$

where $\boldsymbol{X}_{k(c)}$ is the column-centred $\boldsymbol{X}_k$; the columns of $\boldsymbol{V}_k \in \mathbb{R}^{p \times q_k}$ ($q_k = \mathrm{rank}(\mathrm{Cov}(\boldsymbol{X}_k))$) denote the normalised eigenvectors, and $\boldsymbol{\Sigma}_k$ is a diagonal matrix with eigenvalues $\{\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{q_k}\}$. We select the first r_k ($r_k \leq q_k$) columns of $\boldsymbol{V}_k$ as the basis vectors $\boldsymbol{W}_k$ that spans the kth class subspace $\mathscr{P}_k$, which is

r_k-dimensional.

It follows that the projection matrix $\boldsymbol{P}_k \in \mathbb{R}^{p \times p}$ of $\mathscr{P}_k$ can be written as

$$\boldsymbol{P}_k = \boldsymbol{W}_k \boldsymbol{W}_k^T \ . \tag{2.2}$$

In the test phase, a new sample $\boldsymbol{x}_{new}$ is assigned based on the following two residuals. First, the residual of the kth class in the training set:

$$\boldsymbol{E}_k = \boldsymbol{X}_{k(c)} - \boldsymbol{X}_{k(c)} \boldsymbol{P}_k \ . \tag{2.3}$$

Second, the residual of $\boldsymbol{x}_{new}$ when it is projected to the kth class subspace:

$$\boldsymbol{e}^{k,new} = \boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new} \boldsymbol{P}_k \ , \tag{2.4}$$

where $\boldsymbol{x}_{(c)}^{k,new}$ is centred by the mean vector of $\boldsymbol{X}_k$. Then $\boldsymbol{x}_{new}$ is assigned to the class with the smallest F-value (Mertens et al., 1994), where the F-value is defined as

$$F = \frac{||\boldsymbol{e}^{k,new}||_2^2}{||\boldsymbol{E}_k||_2^2/(n_k - r_k - 1)} \ , \tag{2.5}$$

in which $||\cdot||_2$ denotes the Frobenius norm and $||\boldsymbol{e}^{k,new}||_2^2$ is OD^2.

2.1.2 Generalised difference subspace

Since the class subspaces in SIMCA are built independently, the between-class information is not considered by SIMCA and thus the classification performance is limited. To improve the performance of SIMCA, we aim to find a subspace more discriminative than the original feature space. Applying SIMCA to the projections of the samples in this discriminative subspace is expected to have better performance because the samples are expected to be more separated in this subspace. The process of seeking and projecting to such a discriminative subspace can be treated as a preprocessing step of SIMCA.

Mutual subspace method (MSM) is a commonly used subspace-based method for image set-based object classification, which has a similar problem as SIMCA:

MSM builds the class subspace by using PCA for each class separately. The generated class subspace of an image set of an unknown object is compared with the known class subspaces of reference objects and classified to the class with the smallest canonical angle.

When the image set of an unknown object contains only one image, the image is represented by a feature vector and the canonical angles are calculated between the vector and the class subspaces. In this case, MSM is reduced to the commonly-used subspace method (SM) in image classification. The only difference between SM and SIMCA is the criterion for assigning new samples: SM assigns the new sample to the class with the smallest canonical angle between the sample and the class subspace, while SIMCA assigns the new sample to the class with the smallest F-value calculated in (2.5).

MSM suffers from the problem that the class subspaces generated by PCA may not be sufficiently discriminative for classification. Hence recently Fukui and Maki (2015) proposed to project the data onto a generalised difference subspace (GDS) as a preprocessing step of MSM, so as to improve the classification performance of MSM. GDS contains difference information between two class subspaces and is more discriminative to separate the two class subspaces than the original feature space. Thus the projections of the samples to GDS are expected to be more separated and can be better classified. Since SIMCA and MSM suffer from similar problems, we believe the GDS projection can also be used as a preprocessing method of SIMCA to improve the classification performance of the latter.

2.1.2.1 GDS

The GDS projection is proposed on the basis of the properties of the difference subspace (DS) of two class subspaces. The DS, denoted by $\mathscr{D}$, is calculated by using the sum matrix $\boldsymbol{G}_D \in \mathbb{R}^{p \times p}$, which is defined as

$$\boldsymbol{G}_D = \sum_{k=1}^{K} \boldsymbol{P}_k \, , \tag{2.6}$$

where $K = 2$. Applying eigendecomposition to $\boldsymbol{G}_D$, we obtain

$$\boldsymbol{G}_D = \boldsymbol{V}_D \boldsymbol{\Sigma}_D \boldsymbol{V}_D^T \,, \tag{2.7}$$

where the columns in $\boldsymbol{V}_D = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_{r_D}] \in \mathbb{R}^{p \times r_D}$ are the normalised eigenvectors of $\boldsymbol{G}_D$, and $\boldsymbol{\Sigma}_D$ denotes the diagonal matrix with corresponding eigenvalues $\{\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r_D}\}$ in descending order, where $r_D = \text{rank}(\boldsymbol{G}_D)$.

The DS is defined as the subspace spanned by the eigenvectors $\boldsymbol{v}_i$ in $\boldsymbol{V}_D$ with corresponding eigenvalues λ_i less than one. As shown by Fukui and Maki (2015), these eigenvectors are proportional to the difference between the canonical vector pairs of the two class subspaces, and hence they contain the difference information between the two class subspaces.

In addition to DS, Fukui and Maki (2015) also define the principal component subspace (PCS), denoted by $\mathscr{M}$, which is spanned by the eigenvectors $\boldsymbol{v}_i$ in $\boldsymbol{V}_D$ with corresponding eigenvalues λ_i larger than one. They point out that $\mathscr{M}$ contains the similarity information between class subspaces, because the eigenvectors are proportional to the sum of the canonical vector pairs.

Based on the properties of the DS, Fukui and Maki (2015) propose the generalised DS (GDS) projection for K ($K \geq 2$) classes. The GDS projection discards the first few eigenvectors of $\boldsymbol{G}_D$ with large eigenvalues and keeps only the last few eigenvectors of $\boldsymbol{G}_D$ with small eigenvalues. In this way, the GDS spanned by the last few eigenvectors contains difference information between class subspaces. The projections of the samples onto GDS are expected to be more separated and can be better classified. The dimension of GDS is determined by maximising the mean canonical angles between class subspaces, as suggested in Fukui and Maki (2015).

2.1.2.2 The generating matrix

To further investigate the properties of the sum matrix $\boldsymbol{G}_D$ and the GDS, we introduce the generating matrix proposed in Therrien (1975). The generating matrix is defined as the linear combination of the projection matrices of the two class subspaces (Therrien, 1975). Therrien (1975) shows that the generating matrix can be used to find the intersection of the class subspaces.

For two classes, the generating matrix $\boldsymbol{G} \in \mathbb{R}^{p \times p}$ can be written as

$$\boldsymbol{G} = \sum_{k=1}^{K} \alpha_k \boldsymbol{P}_k \ , \tag{2.8}$$

where $K = 2$, $\alpha_k \in (0,1)$, and $\sum_{k=1}^{K} \alpha_k = 1$. Applying eigendecomposition to $\boldsymbol{G}$, we can obtain

$$\boldsymbol{G} = \boldsymbol{V}_G \boldsymbol{\Sigma}_G \boldsymbol{V}_G^T \ , \tag{2.9}$$

where the columns of $\boldsymbol{V}_G \in \mathbb{R}^{p \times r_G}$ denote the normalised eigenvectors of $\boldsymbol{G}$, and $\boldsymbol{\Sigma}_G$ denotes the diagonal matrix with eigenvalues $\{\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r_G}\}$, where $r_G = \operatorname{rank}(\boldsymbol{G})$.

Therrien (1975) shows three important properties of $\boldsymbol{G}$. First, the eigenvalues of $\boldsymbol{G}$ are in the interval $[0,1]$. Second, the eigenvectors with the corresponding eigenvalues of one span the intersection of the two subspaces $\bigcap_{k=1}^{2} \mathscr{P}_k$. Third, the eigenvectors with nonzero eigenvalues span the sum subspace of the two classes, and the eigenvectors with eigenvalues of zeros span the complement of this sum subspace.

Since the vectors in $\bigcap_{k=1}^{2} \mathscr{P}_k$ are in both $\mathscr{P}_1$ and $\mathscr{P}_2$, $\bigcap_{k=1}^{2} \mathscr{P}_k$ denotes the subspace that contains the most similar directions of the two class subspaces. In other words, the most similar directions of the two class subspaces are extracted by the eigenvectors of $\boldsymbol{G}$ with eigenvalues of one. In contrast, the eigenvectors with eigenvalues of zeros are the complements of the sum subspace which contain information that is irrelevant to the two class subspaces. The larger the eigenvalue, the more similarity information the corresponding eigenvector contains.

The generation of GDS is closely related to the generating matrix: $\boldsymbol{G}_D$ and $\boldsymbol{G}$ are both linear combinations of $\boldsymbol{P}_k$ although with different coefficients. The linear coefficients of $\boldsymbol{G}_D$ are all one, i.e. $\alpha_k = 1 \ \forall \ k$, while those of $\boldsymbol{G}$ are constrained by $\alpha_k \in (0,1)$ and $\sum_{k=1}^{K} \alpha_k = 1$. Although $\boldsymbol{G}_D$ and $\boldsymbol{G}$ are slightly different, we can derive similar properties of $\boldsymbol{G}_D$ as those of $\boldsymbol{G}$ by following the proofs in Therrien (1975). First, the eigenvalues of $\boldsymbol{G}_D$ are in the interval $[0,2]$. Second, the eigenvectors with the corresponding eigenvalues of two span the intersection of the two subspaces $\bigcap_{k=1}^{2} \mathscr{P}_k$. Third, the eigenvectors with the corresponding eigenvalues that are nonzero span the sum subspace of the two subspace and those with zero eigenvalues span the complement of the sum subspace. Hence, with some abuse of notation, we also call the sum matrix $\boldsymbol{G_D}$ a generating matrix.

The eigenvectors of $\boldsymbol{G_D}$ with eigenvalues in $(1,2]$ span the PCS $\mathscr{M}$ which contains similarity information between the two class subspaces. This argument seems to be consistent with the property of $\boldsymbol{G_D}$, based on the assumption that the eigenvectors closed to the intersection directions contain large amount of similarity information. Since the eigenvectors with eigenvalues of two span the intersections subspace, the eigenvectors with eigenvalues close to two could be close to the intersection directions. On the other hand, the eigenvectors with eigenvalues far from two, i.e. eigenvalues in $[0,1)$, are far from the intersection directions. Therefore, the GDS projection aims to discard the eigenvectors that are close to the intersection directions, so as to provide a discriminative subspace.

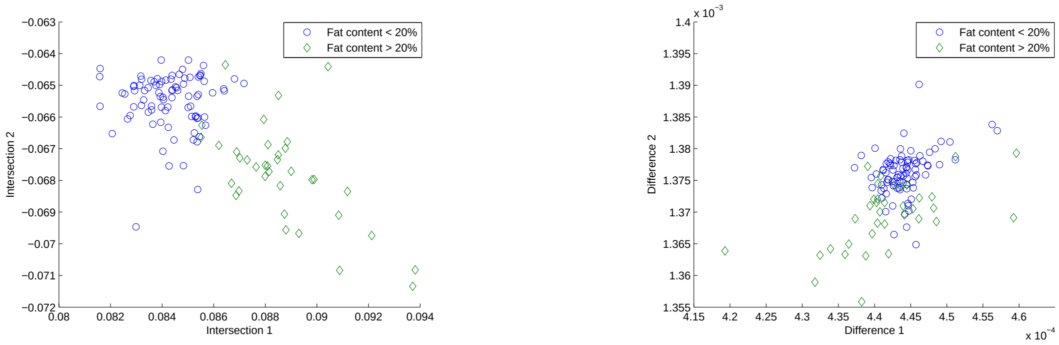
2.1.3 Discriminatively ordered subspace

The GDS projection is based on the assumption that, because the first few eigenvectors with large eigenvalues close to the intersection directions contain similarity information between the class subspaces, they are not important for classification. However, this assumption is not always true, as a class subspace (of infinite scale) and a class (of finite scale) are different, and hence the ability to discriminate two class subspaces are not necessarily in line with the ability to discriminate samples of two classes. In the extreme case, two separable classes may span the same class subspace. More technically, the similarity information in the GDS assumption only considers the directions, while the scores or the projection values on the directions should also be considered. The eigenvectors of $\boldsymbol{G_D}$ that are close to the intersection directions between the two class subspaces can be discriminative when the scores on these eigenvectors are largely separable between classes. In the following section, we show a motivating real-data example that even the directions in the intersection subspace of the two classes can be discriminative.

2.1.3.1 Intersection and discriminative ability: a motivating example

The fat dataset contains 193 spectra of finely chopped meat measured at 100 wavelengths, in which 122 samples contain less than 20% fat and 71 samples contain more than 20% fat. Detailed description of this dataset can be found in Section 2.2.1. We split the dataset into a training set and a test set: 35 samples with fat content less than 20% and 35 samples with fat content more than 20% are randomly sampled into the training set; the rest samples form the test set.

The projection matrix $\boldsymbol{P}_k$ is calculated by using all the 34 available eigenvectors of each class. There are 68 eigenvectors that can be obtained from the eigendecomposition of $\boldsymbol{G}_D$, in which the first seven eigenvectors have eigenvalues of two and the last 34 eigenvectors have eigenvalues less than one. Thus the first seven eigenvectors span the intersection of the two class subspaces and the last 34 eigenvectors span the DS.



(a) **(b)**

Figure 2.4: (a) Projections of the test samples onto two directions of the intersection. (b) Projections of the test samples onto two directions of the DS.

Figure 2.4 shows two scatter plots of the test samples. Figure 2.4a shows the projections of the test samples onto two intersection directions, and Figure 2.4b shows the projections of the test samples onto the first two DS directions. It is clear that the test samples can be well separated when projected onto the two directions in the intersection subspace, whereas the projections of the test samples onto the two directions of DS show slight separation with a mixture in the central region.

In other words, this indicates that the two eigenvectors in the intersection subspace are more discriminative than those in DS. Therefore, it is better to keep the two eigenvectors in the intersection subspace instead of those in the DS.

This counter-example demonstrates that the eigenvectors of $\boldsymbol{G}_D$ in the intersection directions can be discriminative and the assumption in the GDS method is not valid in this case.

2.1.3.2 Discriminatively ordered subspace

As shown in Section 2.1.2.2, the eigenvectors of the generating matrix $\boldsymbol{G}_D$ contain between-class information. Thus we are able to select discriminative eigenvectors of $\boldsymbol{G}_D$ to generate a discriminative subspace for better classification. In the GDS projection, the eigenvectors of $\boldsymbol{G}_D$ are sorted by the eigenvalues in descending order, and the last few eigenvectors with small eigenvalues are selected to generate the GDS. However, as we have shown, the eigenvectors with large eigenvalues are possible to be more discriminative than those with small eigenvalues, and discarding the eigenvectors with large eigenvalues that are discriminative may be harmful for classification.

Therefore, instead of using the GDS projection, we aim to select the most discriminative eigenvectors of $\boldsymbol{G}_D$ to generate a discriminative subspace. We propose a discriminatively ordered subspace (DOS), which uses the discriminative ability (rather than eigenvalues) to sort the eigenvectors in ascending order and select the last few eigenvectors with high discriminative ability to generate the discriminative subspace. In our case for improving SIMCA, the discriminative ability of an eigenvector is measured by the classification accuracy of SIMCA on the samples projected to this eigenvector. For each eigenvector, if the projections of the samples of the two classes are more separated, then the classification accuracy of SIMCA will be high. This simple eigenvector-by-eigenvector selection scheme is appropriate for high-dimensional spectral data, given that the candidate eigenvectors are uncorrelated. In the end we choose a set of eigenvectors with high discriminative abilities to span a subspace that can make the samples of the two classes more separated and improve the performance of SIMCA.

Specifically, given the generating matrix $\boldsymbol{G}_D$ in (2.6) and its eigendecomposition in (2.7), the eigenvectors $\boldsymbol{v}_i$ $(i = 1, \dots, r_D)$ are sorted using their discriminative abilities d_i, which are calculated using leave-one-out cross-validation (LOOCV) on the training set as follows.

The training set is denoted as $\boldsymbol{X}_{train}^T = [\boldsymbol{X}_1^T, \boldsymbol{X}_2^T] = [\boldsymbol{x}_1^T, \dots, \boldsymbol{x}_{N_1+N_2}^T] \in \mathbb{R}^{p\times(N_1+N_2)}$, where $\boldsymbol{X}_1^T = [\boldsymbol{x}_1^T, \dots, \boldsymbol{x}_{N_1}^T] \in \mathbb{R}^{p\times N_1}$ and $\boldsymbol{X}_2^T = [\boldsymbol{x}_{N_1+1}^T, \dots, \boldsymbol{x}_{N_1+N_2}^T] \in \mathbb{R}^{p\times N_2}$ are the training sets for the two classes and $\boldsymbol{x}_m \in \mathbb{R}^{1\times p}$ is the mth $(m = 1, \dots, N_1 + N_2$) training sample.

Firstly, we project all the training samples in $\boldsymbol{X}_{train}$ to each eigenvector $\boldsymbol{v}_i \in \mathbb{R}^{p\times 1}$ and obtain the projections $\hat{\boldsymbol{X}}_{train,i} = \boldsymbol{X}_{train}\boldsymbol{v}_i \in \mathbb{R}^{(N_1+N_2)\times 1}$. For the mth validation, the mth projection, $\hat{\boldsymbol{x}}_{m,i} = \boldsymbol{x}_m\boldsymbol{v}_i \in \mathbb{R}^{1\times 1}$, is used as the validation sample and the rest projections are used as the training samples.

Secondly, we apply SIMCA to each validation by setting the dimensions of the two class subspaces to zeros, i.e. $r_1 = r_2 = 0$. Based on (2.3), (2.4), and (2.5), we observe that the F-value is dependent on the distance from the projected validation sample to the projected class centre. We assign the validation sample to the class with the smallest F-value.

Thirdly, for each eigenvector $\boldsymbol{v}_i$, we obtain $N_1 + N_2$ predictions from LOOCV. The classification accuracy d_i is calculated as

$$d_i = \frac{N_c}{N_1 + N_2}\,, \tag{2.10}$$

where N_c is the number of correctly classified test samples.

Fourthly, after obtaining $d_i's$ for $i = 1, \dots, r_D$, we sort the eigenvectors $\boldsymbol{v}_i's$ in ascending order of $d_i's$ and obtain the matrix of the sorted eigenvectors $\boldsymbol{V}_{sort} = [\boldsymbol{v}_{(1)}, \boldsymbol{v}_{(2)}, \dots, \boldsymbol{v}_{(r_D)}]$, where the discriminative ability $d_{(1)} < d_{(2)} < \cdots < d_{(r_D)}$. The last few eigenvectors in $\boldsymbol{V}_{sort}$ are selected to span the discriminative subspace $\mathscr{D}_s$, which we term discriminatively sorted subspace (DOS).

Finally, we project the samples to DOS and apply SIMCA to the projections of the samples. The dimension of $\mathscr{D}_s$ and the dimensions of the two class subspaces in $\mathscr{D}_s$ can be tuned by cross-validation through minimising the classification error of the training set.

2.2 Experiments

In the following experiments, we compare the performances of the original SIMCA without preprocessing, the SIMCA preprocessed by the linear discriminative analysis (LDA) projection, the SIMCA preprocessed by the GDS projection, and the SIMCA preprocessed by the DOS projection. The LDA-preprocessed SIMCA is also compared since LDA is a commonly used method to find a discriminative subspace. Three real datasets are used in the experiments: the fat dataset, the meat dataset, and the Phenyl dataset. In the illustrations presented in this section, the DOS-preprocessed SIMCA is denoted by 'DOS', the GDS-preprocessed SIMCA is denoted by 'GDS', the LDA-preprocessed SIMCA is denoted by 'LDA' and the original SIMCA is denoted by 'SIMCA'.

2.2.1 Datasets

2.2.1.1 The meat dataset

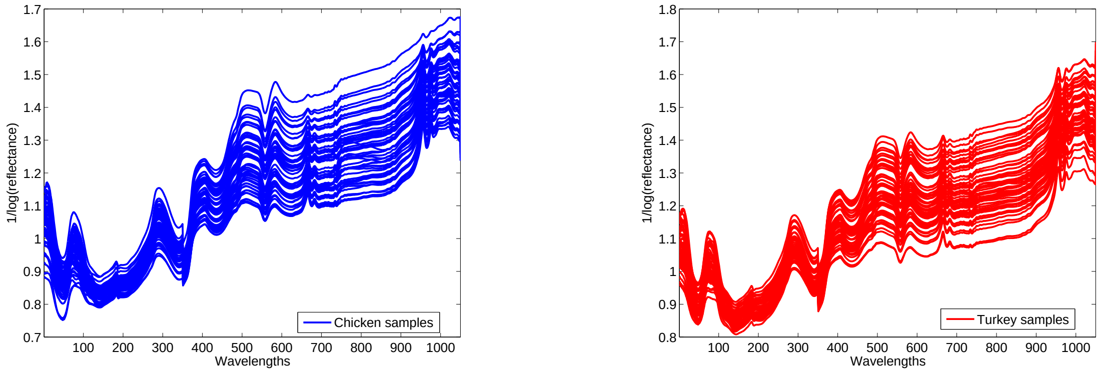


Figure 2.5: The spectra of the two classes in the meat dataset.

The meat dataset (Arnalds et al., 2004) contains beef, pork, lamb, chicken and turkey meat samples measured at 1051 wavelengths. Only the 55 chicken and 54 turkey samples in the dataset are used in our experiments since the two groups are difficult to classify. The first 350 wavelengths in the meat dataset are used because the experiments in Arnalds et al. (2004) suggest that the first 350 wavelengths ranging from 400 to 1100 nm perform the best. The spectra of the meat dataset are illustrated in Figure 2.5.

During the training-test split, the total of 55 chicken samples and 54 turkey samples are randomly partitioned into a training set (27 chicken samples and 27 turkey samples) and a test set (28 chicken samples and 27 turkey samples).

2.2.1.2 The Phenyl dataset

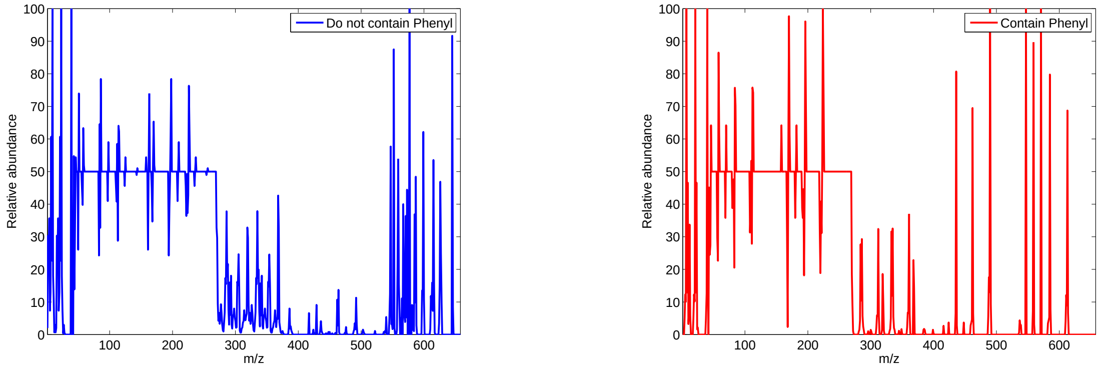


Figure 2.6: The spectra of the two classes in the Phenyl dataset.

The Phenyl dataset is provided in the R package, ‘chemometrics’. The dataset consists of 600 mass spectra of chemical components, with 300 compounds contain the phenyl substructure and 300 compounds do not contain the substructure. Each spectrum contains 658 mass spectral features. Since a plot of the spectra of all samples is confusing, we only show the spectra of two instances in the Phenyl dataset, one for each class, in Figure 2.6.

We randomly select 100 samples from the Phenyl dataset for our experiments, with 50 contain the phenyl substructure and 50 do not contain the structure. These 100 instances are randomly partitioned into two equal subsets: a training set containing 50 samples (25 contain the phenyl substructure and 25 do not contain the substructure), and a test set containing 50 samples (25 contain the phenyl substructure and 25 do not contain the substructure).

2.2.1.3 The fat dataset

The fat content dataset (Ferraty and Vieu, 2006) contains 193 spectra of finely chopped meat measured at 100 wavelengths, in which 122 meat samples contain less than 20% fat and 71 samples contain more than 20% fat. The spectra of the

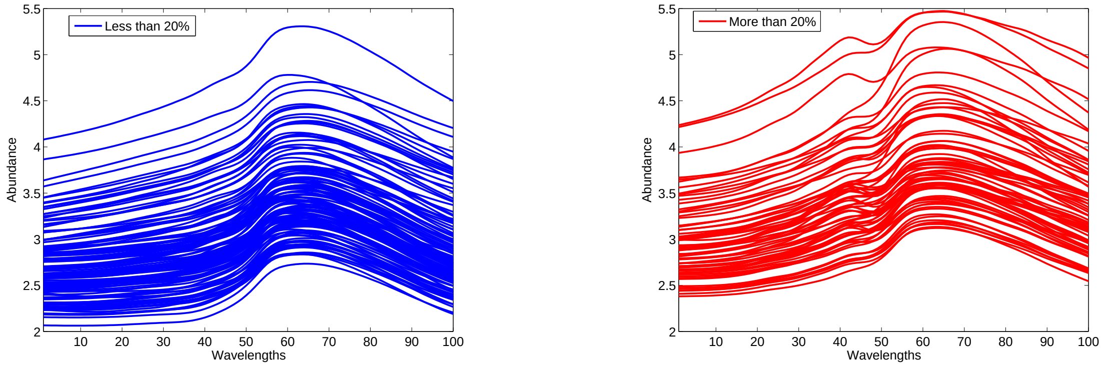


Figure 2.7: The spectra of two classes in the fat content dataset.

data of the two classes are shown in Figure 2.7.

For this dataset, 100 samples are selected as a training set (50 samples with the fat content less than 20% and 50 samples with the fat content larger than 20%) and the remaining samples are selected as a test set.

2.2.2 Experiment settings

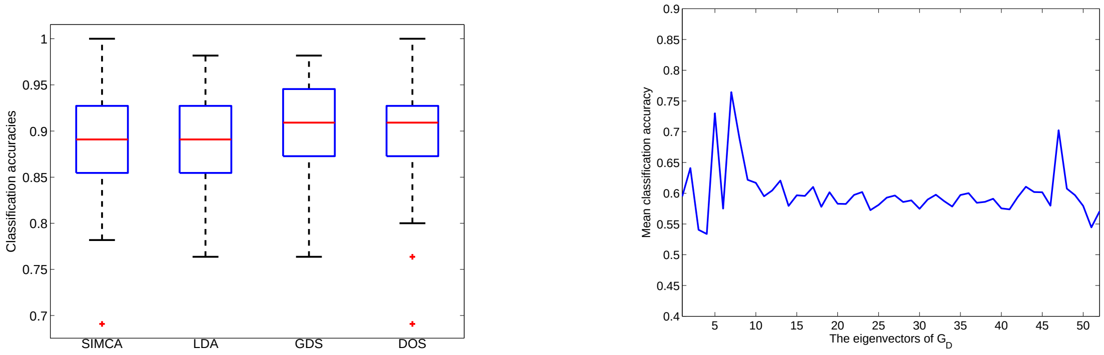
The performances of the original SIMCA, the LDA-preprocessed SIMCA, the GDS-preprocessed SIMCA, and the DOS-preprocessed SIMCA are compared.

In SIMCA, the dimensions of the two class subspaces are tuned by 10-fold cross-validation. Before applying LDA, the high-dimensional spectral data are projected to the PC subspace of all available PCs. Then in LDA-preprocessed SIMCA, the dimensions of the two class subspaces are set to zeros because only one discriminative direction can be found for two classes by LDA and this direction should be used for classification. In GDS and DOS, all the available PCs of each class subspace are used to obtain the generating matrix $\boldsymbol{G}_D$. In GDS, the dimension of GDS and the dimensions of the two class subspaces are also tuned by 10-fold cross-validation. The dimensions are chosen to minimise the classification error. In DOS, the discriminative order of the eigenvectors of $\boldsymbol{G}_D$ is determined by using the training set. Leave-one-out cross-validation (LOOCV) is used to obtain the classification accuracy of each eigenvector. The dimension of $\mathscr{D}_s$ and the dimensions of the two class subspaces are also tuned by 10-fold cross-validation. The dimensions are chosen to minimise the classification error, same as those for SIMCA and GDS.

All the experiments are repeated 100 times and the classification accuracies of all the experiments are recorded and depicted in boxplots.

2.2.3 Results

2.2.3.1 The meat dataset



(a) Classification accuracies.

(b) Discriminative abilities.

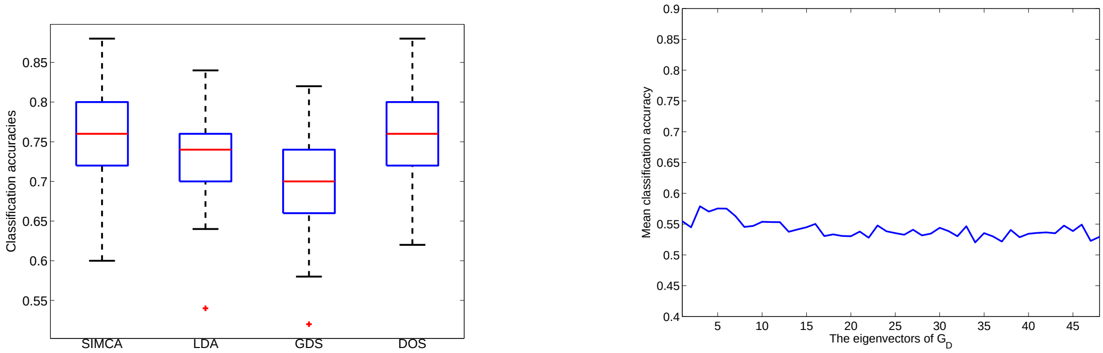
Figure 2.8: For the meat dataset: (a) classification accuracies of SIMCA, LDA, GDS and DOS; (b) discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G}_D$.

Figure 2.8a shows the boxplots of the classification accuracies of the four methods for the meat dataset, from which we can observe that LDA performs similar to SIMCA while GDS and DOS both perform on average better than SIMCA.

Figure 2.8b shows the discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G}_D$ versus the descending order of eigenvalues, which explains the good performance of GDS. That is, in Figure 2.8b, the horizontal axis shows the eigenvectors of $\boldsymbol{G}_D$ with eigenvalues in descending order and the vertical axis shows the corresponding average classification accuracies of SIMCA using the projected samples onto each of the eigenvectors. Since the first few eigenvectors of $\boldsymbol{G}_D$ do not have high discriminative abilities, discarding them, as done by GDS, can benefit classification, and thus GDS can provide good classification results.

In short, Figure 2.8 suggests that GDS performs well when the deletion of the first few eigenvectors (in terms of large eigenvalues) is beneficial for classification. In addition, DOS can achieve similarly good classification performance as GDS in this situation, as the first few eigenvectors are also not selected by DOS due to their low discriminative abilities.

2.2.3.2 The Phenyl dataset



(a) Classification accuracies.

(b) Discriminative abilities.

Figure 2.9: For the Phenyl dataset: (a) classification accuracies of SIMCA, LDA, GDS and DOS; (b) discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G}_D$.

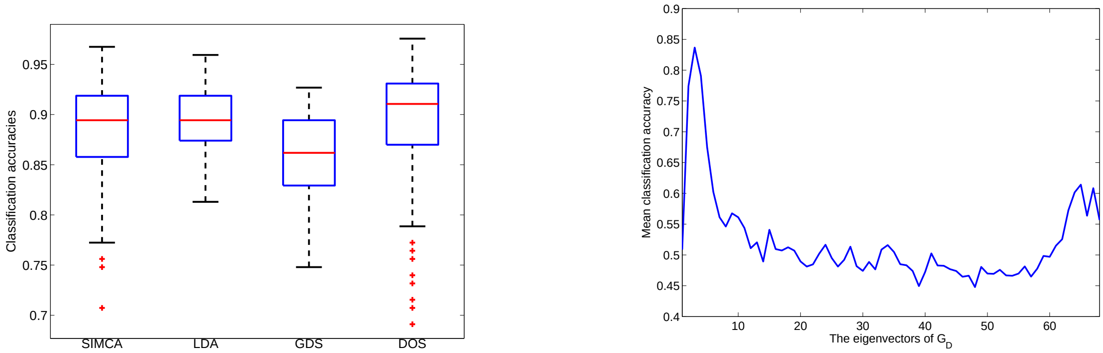
As we have seen in Figure 2.3, GDS may fail to provide good classification results in the cases of the Phenyl and fat datasets. Now we shall see that DOS may provide good classification results even when GDS fails in these cases.

Figure 2.9a shows that GDS performs worse than SIMCA, which indicates that the GDS projection is not a good preprocessing method for the Phenyl dataset. LDA performs better than GDS, but worse than SIMCA. In contrast, DOS performs better than GDS and LDA, although only providing similar classification accuracies as SIMCA in this case.

To explain this result, we can check Figure 2.9b, which shows the discriminative abilities of the eigenvectors of $\boldsymbol{G}_D$ for the Phenyl dataset. On the one hand, we observe that the first few eigenvectors with large eigenvalues have higher discriminative abilities than the remaining ones. Thus deleting the first few eigenvectors is harmful to classification. This explains why GDS cannot provide good classification results. On the other hand, we also observe that the discriminative abilities of the eigenvectors are ranged from 0.52 to 0.58, which suggests that the discriminative abilities of the eigenvectors are similar to each other. Since the eigenvectors are similarly important to classification in this case, it is hard to achieve better classification by selecting from these eigenvectors. This explains why DOS performs similarly to SIMCA.

In summary, Figure 2.9 indicates that GDS fails to provide good classification results in the situation where the first few eigenvectors (in terms of large eigenvalues) of $\boldsymbol{G}_D$ are important for classification. DOS can provide better classification results than GDS in this situation. However, the classification results of DOS do not show noticeable improvement compared with those of SIMCA for this dataset, because the eigenvectors of $\boldsymbol{G}_D$ have similar discriminative abilities.

2.2.3.3 The fat dataset



(a) Classification accuracies.

(b) Discriminative abilities.

Figure 2.10: For the fat dataset: (a) classification accuracies of SIMCA, LDA, GDS and DOS; (b) discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G}_D$.

Here we shall demonstrate that DOS can achieve better classification accuracies than SIMCA when the discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G}_D$ have a large variation. In this situation, DOS can select the most discriminative eigenvectors to make the samples more separate and is a good preprocessing method for classification.

As shown in Figure 2.10a for the fat dataset, GDS performs worse than SIMCA and LDA, but DOS can achieve better performance than SIMCA and LDA.

Once again, let us use Figure 2.10b to explain the above results. On the one hand, because the discriminative abilities of the first few eigenvectors are higher than the remaining ones, GDS deletes the first few eigenvectors of $\boldsymbol{G}_D$ that are actually discriminative for classification, leading to a poor performance. On the other hand, Figure 2.10b shows that the discriminative abilities range from 0.45

to 0.85, which indicate a large difference in discriminative abilities between the eigenvectors. Hence DOS can select the most discriminative eigenvectors of $\boldsymbol{G_D}$ and provide better classification results than SIMCA.

To sum up, Figure 2.10 suggests that DOS performs well when there is a large difference in the discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G_D}$. The good performance of DOS demonstrates that selecting the eigenvectors of $\boldsymbol{G_D}$ by using the discriminative ability instead of using eigenvalues can be effective, when GDS fails to provide improvement in classification.

2.2.3.4 Summary of experiments

We would like to convey two messages through our experiments.

Firstly, from Figure 2.8b, Figure 2.9b and Figure 2.10b, we can observe that there is no negative correlation between eigenvalues and discriminative abilities of the eigenvectors of the generating matrix $\boldsymbol{G_D}$. The eigenvectors with large eigenvalues, although close to the intersection of two class subspaces, may have high discriminative abilities and can largely benefit classification of the samples of the two classes.

Secondly, from Figure 2.8a, Figure 2.9a and Figure 2.10a, we can observe that DOS can provide superior or at least comparable classification performance to SIMCA, LDA and GDS. The classification results suggest that it is appropriate to use high discriminative ability, instead of using low eigenvalues (or being away from the intersection of class subspaces), to select the eigenvectors of $\boldsymbol{G_D}$ to span a discriminative subspace for classification.

2.2.4 Discussion

2.2.4.1 Intersection of two class subspaces and its discriminative ability

In Section 2.1.3.1, we have shown a motivating example that the intersection of two class subspaces can be discriminative for the fat dataset. In this section, we further investigate the relationship between the intersection and its discriminative ability for all the three datasets.

To check whether an eigenvector $\boldsymbol{v}_i$ is the intersection between class subspaces, we define $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$ to measure the Euclidean distances from $\boldsymbol{v}_i$ to its projections in the two class subspaces, respectively. When $\boldsymbol{v}_i$ is in both class subspaces, it is the intersection of the two class subspaces. To be more specific, the Euclidean distances from $\boldsymbol{v}_i$ to its projections in the two class subspaces are zeros when $\boldsymbol{v}_i$ is the intersection. The larger the Euclidean distances, the farther $\boldsymbol{v}_i$ away from the two class subspaces.

Suppose the two class subspaces, $S(\boldsymbol{P}_1)$ and $S(\boldsymbol{P}_2)$, are defined by two projection matrices $\boldsymbol{P}_1 \in \mathbb{R}^{p\times p}$ and $\boldsymbol{P}_2 \in \mathbb{R}^{p\times p}$, respectively. The Euclidean distances from $\boldsymbol{v}_i$ to its projections in the two subspaces can be calculated as

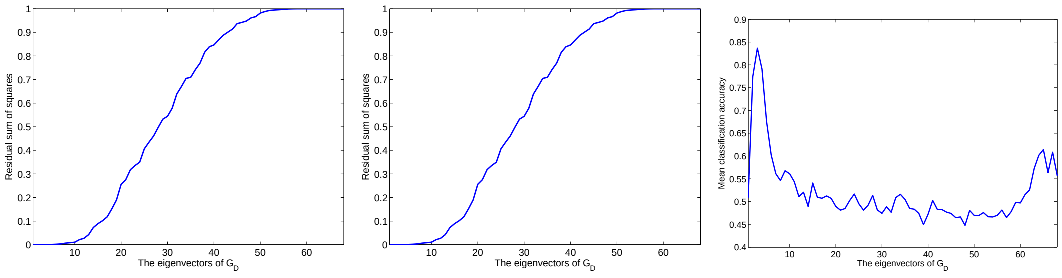
$$||\boldsymbol{e}_1||_2^2 = ||\boldsymbol{P}_1\boldsymbol{v}_i - \boldsymbol{v}_i||_2^2 \tag{2.11}$$

and

$$||\boldsymbol{e}_2||_2^2 = ||\boldsymbol{P}_2\boldsymbol{v}_i - \boldsymbol{v}_i||_2^2, \tag{2.12}$$

respectively. As $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$ decrease, $\boldsymbol{v}_i$ goes closer to the two class subspaces and to the intersection. If $||\boldsymbol{e}_1||_2^2 = 0$ and $||\boldsymbol{e}_2||_2^2 = 0$, then $\boldsymbol{v}_i$ is the intersection of the two class subspaces, because $\boldsymbol{v}_i$ is in both subspaces, i.e. $\boldsymbol{P}_1\boldsymbol{v}_i = \boldsymbol{v}_i$ and $\boldsymbol{P}_2\boldsymbol{v}_i = \boldsymbol{v}_i$.

In the following part of this section, we discuss the relationship between the subspace intersection and its discriminative ability based on the values of $||\boldsymbol{e}_1||_2^2$, $||\boldsymbol{e}_2||_2^2$, and the corresponding discriminative abilities of the eigenvectors of $\boldsymbol{G}_D$.



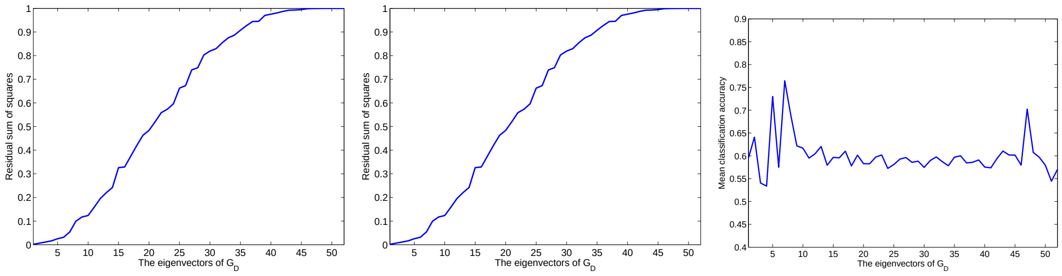
(a) $||\boldsymbol{e}_1||_2^2$ **(b)** $||\boldsymbol{e}_2||_2^2$ **(c)** Discriminative ability

Figure 2.11: For the eigenvectors of $\boldsymbol{G}_D$ of the fat dataset: their distances ($||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$) to the two class subspaces, and their discriminative abilities.

As an extension of the motivating example in Section 2.1.3.1 for the fat dataset, we present three plots in Figure 2.11 illustrating the relationship between the intersection of the two class subspaces and its discriminative ability.

Figure 2.11a and Figure 2.11b plot $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$ against the descending order of eigenvalues, respectively. More specifically, in Figure 2.11a and Figure 2.11b, the horizontal axis lists the eigenvectors of $\boldsymbol{G}_D$ in the order of descending eigenvalues, and the vertical axis shows their values of $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$. Figure 2.11c depicts the discriminative abilities of the eigenvectors, which is the same as Figure 2.10b.

We can clearly observe that the first few eigenvectors with the largest eigenvalues span the intersection of the two class subspaces of the fat dataset, because $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$ of these eigenvectors are all zeros. However, we can also find that the corresponding discriminative abilities of these eigenvectors are higher compared with other eigenvectors, as shown in Figure 2.11c. That is, for the fat dataset, the intersection between the two class subspaces has high discriminative ability.



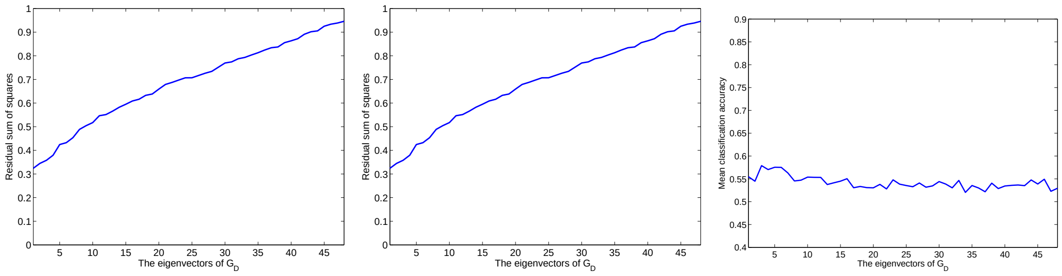
(a) $||\boldsymbol{e}_1||_2^2$ **(b)** $||\boldsymbol{e}_2||_2^2$ **(c)** Discriminative ability

Figure 2.12: For the eigenvectors of $\boldsymbol{G}_D$ of the meat dataset: their distances ($||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$) to the two class subspaces, and their discriminative abilities.

In contrast to the relationship observed in the fat dataset, here we shall see that the intersection can also have low discriminative ability.

The first eigenvector of the meat dataset is the intersection between the two class subspaces, as shown in Figure 2.12a and Figure 2.12b. The discriminative ability of this eigenvector is 0.6, which is low compared with many other eigenvectors. In other words, for the meat dataset, the intersection of the two class subspaces has low discriminative ability.

Despite the two datasets discussed above that there exists intersection between



(a) $||\boldsymbol{e}_1||_2^2$ **(b)** $||\boldsymbol{e}_2||_2^2$ **(c)** Discriminative ability

Figure 2.13: For the eigenvectors of $\boldsymbol{G}_D$ of the Phenyl dataset: their distances ($||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$) to the two class subspaces, and their discriminative abilities.

class subspaces, now we show another dataset, the Phenyl dataset, that it is also possible that there is no intersection between two class subspaces.

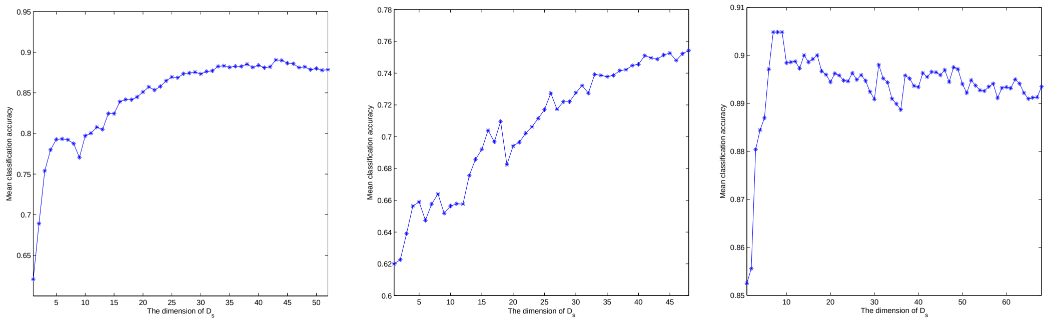
We can observe from Figure 2.13a and Figure 2.13b that $||\boldsymbol{e}_1||_2^2$ and $||\boldsymbol{e}_2||_2^2$ of the first eigenvector are far from zeros. Thus there seems to be no intersection between the two class subspaces for the Phenyl dataset.

Therefore, we can draw two conclusions based on the observations from Figure 2.11, Figure 2.12, and Figure 2.13. First, the intersection between class subspaces does not always exist in all datasets. Second, even when the intersection exists, there is no definitely negative correlation between the intersection and its discriminative ability; that is, the discriminative ability of the intersection of two class subspaces is data-dependent, not necessarily low.

The second conclusion above supports our argument that there is difference between a class subspace and a class. The intersection represents the same directions that two class subspaces can take, which can be discarded if we aim to classify two class subspaces. However, the intersection can be discriminative, and thus is important and cannot be simply discarded when we aim to classify the samples of two classes, which is actually the task of classification in practice.

2.2.4.2 Cross-validation of the dimension of the discriminatively ordered subspace

In the DOS projection, the dimension of DOS $\mathscr{D}_s$ is an important parameter we need to tune. In this section, we discuss the effectiveness of using cross-validation to determine it.



(a) meat **(b)** Phenyl **(c)** fat

Figure 2.14: Effect of the dimension of $\mathscr{D}_s$.

Figure 2.14 plots the effect of the dimension of $\mathscr{D}_s$ on the classification accuracy on the test sets of the three real datasets, where the dimension changes from one to the total number of eigenvectors in $\boldsymbol{V}_{sort}$. One hundred experiments of DOS are repeated for each dimension and the mean classification accuracies are plotted.

For the meat dataset, the dimension of $\mathscr{D}_s$ determined by 10-fold cross-validation in Section 2.2.3, which uses the training set only, ranges from 41 to 47 in the repeated experiments. Figure 2.14a shows a small peak of the mean classification accuracy of the test set around the dimension of 43, which is in line with the dimension determined by the training set-based 10-fold cross-validation.

For the fat dataset, the same effectiveness can be observed: the peak of the mean classification accuracy of the test set is around seven, as shown in Figure 2.14c, which is roughly consistent with the dimension (which is from two to seven) determined by using 10-fold cross-validation on the training set.

For the Phenyl dataset, Figure 2.14b does not show an obvious peak, and the mean classification accuracy of the test set seems to increase with the dimension and become stable when the dimension is larger than 41. The dimension determined by 10-fold cross-validation using the training set ranges from 38 to 43, which also conforms with the dimension of 41 in the test set.

In short, Figure 2.14 implies that the dimension of $\mathscr{D}_s$ determined by cross-validation using the training set is roughly consistent with the dimension with the largest mean classification accuracy of the test set. Thus cross-validation is an effective way to determine the dimension of $\mathscr{D}_s$ for the DOS projection.

2.3 Conclusion

SIMCA is a widely-used subspace method for classifying two-class high-dimensional spectral datasets. It suffers from the problem that the class subspaces are built independently without considering between-class information. This problem can be tackled by projecting the data to a subspace more discriminative than the original feature space before applying SIMCA. We have proposed a new method, the DOS projection, to generate such a discriminative subspace, by considering the between-class information and the discriminative ability of each basis vector of the subspace. The experiments on three real-world spectral datasets have demonstrated the effectiveness of the DOS projection.

Chapter 3

Dual of nearest-class-model methods: a separating hyperplane classification framework

SIMCA is one famous example of a category of popular classification methods: the subspace-based classifiers, also known as the class modelling methods in the chemometrics community or the subspace methods in the machine learning and pattern recognition communities. In the subspace-based classifiers, each class is modelled by a subspace generated from the training samples of that class, independently of other classes; a test sample is assigned to the class with the highest similarity between the sample and the class model.

Principal component (PC) subspace is a widely-used class subspace. The PC subspace of a class is built through principal component analysis (PCA) of the training samples of that class, such that a class is represented by a low-dimensional linear subspace spanned by a small number of selected PCs. The leading PCs, constructed by the linear combinations of the original features, extract the most variable information in the class and remove a large amount of redundant information in the original features. Hence the PC subspace has been widely used as a class representation, especially for classification of high-dimensional data.

SIMCA (Wold, 1976) is one representation of PC-subspace-based classifiers. In SIMCA, the dissimilarity measure is the reweighted Euclidean distance from a test sample to a PC subspace; a test sample is assigned to the nearest PC subspace based on this distance.

It is, however, not necessary to use subspaces to represent classes. The geometric convex model representation is another popular class representation approach for classification tasks. The geometric convex model for a class is constructed by a linear combination of class samples, with certain constraints on the linear combination coefficients.

The convex hull representation (Nalbantov et al., 2006; Cevikalp and Triggs, 2010; Cevikalp et al., 2008; Zhou and Shi, 2009) is one geometric model that attracts a lot of attention recently. Nalbantov et al. (2006) propose the nearest convex hull classification, which uses a convex hull model to represent a class and classifies a test sample to the class with the nearest convex hull. The convex hull model of a class is constructed by the convex combination, i.e. the linear combination with nonnegative and sum-to-one constraints on the coefficients, of the training samples of that class. The dissimilarity measure is the Euclidean orthogonal distance from a test sample to a convex hull (Nalbantov et al., 2006).

The convex cone model has also been used as class representation for face recognition (Kobayashi and Otsu, 2008). A convex cone model is constructed by the conic combinations of the class samples, i.e. the linear combinations with nonnegative coefficients. Kobayashi and Otsu (2008) propose the cone-restricted subspace method, using the angle between a test sample and a convex cone for classification.

The PC subspace is a set of vectors that are linear combinations of the PCs with no constraints on the coefficients. Thus the PC subspace covers an infinite area that has weak constraints on the location of a class within its class subspace, which is considered as a loose representation of the class. In contrast, the geometric convex model provides a restricted area to represent the class by setting constraints on the linear combination coefficients. The restricted area is bounded by the class samples that are used to construct the convex model. In addition, the coefficients of the convex models usually have physical meanings in real-world applications, such as the abundances of the endmembers in hyperspectral image unmixing and the compositions of chemical compounds in chemometrics.

The convex hull model adopts the convex constraints on the linear combination coefficients. However, the convex constraint is often too tight in the sense that the classes often extend well beyond the convex hulls (Cevikalp et al., 2008). Considering the tightness of a model, a convex cone model lies in between a linear subspace model and a convex hull model. A convex cone is more restricted than a linear subspace because of the nonnegative constraints on the coefficients, while is looser than a convex hull because the conic combination constraint is looser than the convex combination constraint.

The geometric convex-model-based classification methods have shown superior classification performances to the PC subspace classifiers (Nalbantov et al., 2006; Kobayashi and Otsu, 2008). However, the literature of SIMCA have barely explored the potentially beneficial changing of class models for better classification of spectral data. In addition, the reason why the classification performance of the geometric convex model is better for certain datasets is also barely explored in literature.

In this chapter, we aim to use geometric convex class representation models, the convex hull and the convex cone, in SIMCA instead of the PC class subspace, for spectral data classification. We also aim to investigate and compare the classification schemes based on the three class representation models to assist the understanding of their classification performances for certain datasets.

To make the investigation more straightforward, we use the orthogonal distance (OD) from a test sample to a class model as the dissimilarity measure (or classification rule) for classification, i.e. a test sample is assigned to the class with the shortest OD from that sample to the class model. We use OD instead of OD^2 in this chapter because OD can provide the same classification results as OD^2 and is more convenient for the investigation, especially the dual analysis. To avoid confusions with SIMCA, we name the classification methods using different class models with the dissimilarity measure OD as nearest class-model-based classification methods.

In this fashion, the PC subspace representation leads to the nearest subspace method (NSM), which is equivalent to SIMCA using OD as the classification rule (SIMCA-OD); the convex hull model leads to the nearest convex hull method (NCHM) (Nalbantov et al., 2006), which is equivalent to SIMCA-OD using convex hulls as class models; and for the convex cone model, we propose the nearest convex cone method (NCCM), which is equivalent to SIMCA-OD using convex cones as class models. Note that NCCM is different from the method in Kobayashi and Otsu (2008), since the dissimilarity measure is now distance instead of angle.

Since the models are built in different ways, a direct comparison of the three methods is hard. To solve this problem, we shall seek a common platform for the comparison of the three methods. We achieve this by noticing the link between the geometric convex models and the separating hyperplanes for classification in SVM (Bennett and Bredensteiner, 2000; Zhou et al., 2002). Bennett and Bredensteiner (2000) show that determining the best separating hyperplane in SVM is equivalent to looking for the nearest points of the convex hulls of the training samples of two classes, through the dual analysis for SVM.

In this chapter, we find the equivalent hyperplane-based classifiers for the three methods through the dual analysis of their minimum distance problems. We show that the minimum distance from a test sample to a class model is equivalent to the maximum distance from that sample to a hyperplane. Thus for each class model, we can find one separating hyperplane that separates the test sample from the class training samples. The test sample is then classified to the class with the nearest hyperplane. We show from a pure geometric view the theoretical results for the dual analysis of the minimum distance problems in linear vector spaces with arbitrary norms.

In this way, comparing the three different class-model-based classification methods is transformed to comparing the separating hyperplanes found in the dual analysis. The latter comparison is simpler than the former one because the hyperplanes could be compared simply based on their parameters, i.e. normal vectors and biases. In addition, the separating hyperplanes could assist the understanding of the classification schemes of the class-model-based methods.

Furthermore, we establish a separating hyperplane classification (SHC) framework which generalises the class-model-based methods to a framework, based on the separating hyperplanes found in the dual analysis. The SHC framework describes a category of classification methods that classify a test sample based on its pair of separating hyperplanes. The test sample is assigned to the class with the nearest hyperplane, based on the arbitrary-norm-measured distance. We show that the normal vectors of the separating hyperplanes are of great importance to classification: the more discriminative the normal vectors, the better the classification.

It is worth noting that the SHC framework is different from the extensions of SVM based on a pair of separating hyperplanes in one-sided best fitting hyperplane classifier (1S-BFHC) or two-sided best fitting hyperplane classifier (2S-BFHC) (Cevikalp, 2016), generalised eigenvalue proximal support vector machine (GEPSVM) (Mangasarian and Wild, 2006) or twin support vector machine (TSVM) (Jayadeva et al., 2007). In Cevikalp (2016), Mangasarian and Wild (2006) and Jayadeva et al. (2007), the pair of separating hyperplanes are found for the pair of class models and are fixed for all the test samples, making the classification boundary linear for linear kernels. In contrast, the pair of separating hyperplanes in our SHC framework vary with test samples, making the classification boundary nonlinear.

By linking the class-model-based methods with the hyperplane-based classification through the SHC framework, we could design complicated classifiers under the framework, inspired by the well-studied SVM and their extensions based on hyperplanes. For example, the kernel tricks could be easily induced; and the optimisation problems could also be solved by the sequential minimal optimisation (SMO) algorithm used in SVM.

Empirically, we apply NSM, NCHM and NCCM to three real spectroscopic datasets and show that the classification performances of the three methods are data-dependant. We are enabled to explain why one class model is better than others for a specific dataset, based on the comparison of the normal vectors of the separating hyperplanes. Moreover, we propose a novel data exploration scheme to analyse the properties of a dataset to understand why such properties can make a class model suitable for the data.

In summary, the contributions of this chapter are fivefold.

1. In Section 3.1.2.2, we propose NCCM to fill the gap between NSM and NCHM for nearest class-model-based methods, considering the model tightness.

2. We present the dual analysis of NSM, NCHM and NCCM in Section 3.2.3. We also prove the theoretical results for NCCM based on the relationship between a convex cone and its polar cone.

3. In Section 3.2.4, we establish a separating hyperplane classification (SHC) framework for the nearest class-model-based methods on arbitrary norms. The normal vectors of the separating hyperplanes are shown vital to classification. The SHC framework could improve the understanding of the nearest class-model-based methods and provide easy comparison of NSM, NCHM and NCCM.

4. We propose a data exploration scheme in Section 3.3.5, to analyse the properties of datasets and explain why such properties make a class model suitable for the data.

5. Throughout the chapter, we provide geometric intuitions to assist the understanding of the methods, the theoretical analysis and the empirical analysis.

Overall this chapter is organised as follows. In Section 3.1, we discuss NSM, NCHM and NCCM. In Section 3.2, we show the dual analysis of NSM, NCHM and NCCM. In Section 3.3, NSM, NCHM and NCCM are compared on three real datasets. Section 3.4 presents some concluding remarks.

3.1 Methodology

3.1.1 PC Subspace representation: nearest subspace method (NSM)

We first define subspace as follows.

Definition 3.1.1. *Subspace.* Suppose $S = \{\boldsymbol{x}_i\}_{i=1}^{N}$ is a subset of $\mathbb{R}^p$. The set $\mathscr{L}(S) = \{\boldsymbol{v} : \boldsymbol{v} = \sum\limits_{i=1}^{N} \alpha_i \boldsymbol{x}_i \mid \boldsymbol{x}_i \in S, \alpha_i \in \mathbb{R}\}$, called the subspace generated by S, consists of all vectors in $\mathbb{R}^p$ which are linear combinations of vectors in S. We also say that the vectors in S span the subspace $\mathscr{L}(S)$.

In the training phase, NSM builds class subspaces for the classes separately using PCA. We denote $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times p}$ as the training set of class k ($k = 1, 2$ for two-class classification), where n_k is the number of training samples and each row of $\boldsymbol{X}_k$ represents a p-dimensional training sample. The PC subspace for the kth class can be obtained from applying the reduced singular value decomposition to the column-centred $\boldsymbol{X}_k$:

$$\boldsymbol{X}_{k(c)} = \boldsymbol{U}_k \boldsymbol{\Lambda}_k \boldsymbol{V}_k^T, \tag{3.1}$$

where the rows of $\boldsymbol{U}_k \in \mathbb{R}^{n_k \times q_k}$ denote the normalised PC scores; the columns of $\boldsymbol{V}_k \in \mathbb{R}^{p \times q_k}$ denote the PCs; and $\boldsymbol{\Lambda}_k$ is a diagonal matrix of singular values $\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{q_k}\}$. The r_k-dimensional ($r_k \leq q_k$) PC subspace $\mathscr{L}(\boldsymbol{W}_k)$ is spanned by the first r_k PCs $\boldsymbol{W}_k \in \mathbb{R}^{p \times r_k}$.

In the test phase, a new sample $\boldsymbol{x}_{new} \in \mathbb{R}^{1 \times p}$ is assigned according to the distance from $\boldsymbol{x}_c^{k,new}$ to the class subspace $\mathscr{L}(\boldsymbol{W}_k)$, where $\boldsymbol{x}_c^{k,new}$ is centred by the mean vector of $\boldsymbol{X}_k$. The distance is defined as the minimum distance from $\boldsymbol{x}_c^{k,new}$ to the vectors in $\mathscr{L}(\boldsymbol{W}_k)$:

$$d_k^{\mathscr{L}} = \min_{\boldsymbol{\alpha}_k^{\mathscr{L}}} ||\boldsymbol{x}_c^{k,new} - (\boldsymbol{W}_k \boldsymbol{\alpha}_k^{\mathscr{L}})^T||_2, \tag{3.2}$$

where $\boldsymbol{\alpha}_k^{\mathscr{L}} \in \mathbb{R}^{r_k \times 1}$ contains r_k coefficients associated with the r_k PCs in $\boldsymbol{W}_k$. The minimisation problem (3.2) has a closed-form solution of $\boldsymbol{\alpha}_k^{\mathscr{L}*} = (\boldsymbol{x}_c^{k,new} \boldsymbol{W}_k)^T$.

Thus the distance can be written as

$$d_k^{\mathscr{L}} = ||\boldsymbol{x}_c^{k,new} - \boldsymbol{x}_c^{k,new}\boldsymbol{P}_k||_2, \tag{3.3}$$

where $\boldsymbol{P}_k = \boldsymbol{W}_k\boldsymbol{W}_k^T$ is the projection matrix of the subspace $\mathscr{L}(\boldsymbol{W}_k)$; $\boldsymbol{x}_c^{k,new}\boldsymbol{P}_k$ is the projection of $\boldsymbol{x}_c^{k,new}$ on $\mathscr{L}(\boldsymbol{W}_k)$. NSM assigns $\boldsymbol{x}_{new}$ to the class with the smallest $d_k^{\mathscr{L}}$:

$$\hat{y}^{\mathscr{L}} = \underset{k}{\operatorname{argmin}}\, d_k^{\mathscr{L}}, \tag{3.4}$$

where $\hat{y}^{\mathscr{L}}$ denotes the predicted label for $\boldsymbol{x}_{new}$ by NSM. NSM can be considered as SIMCA using OD as the classification rule.

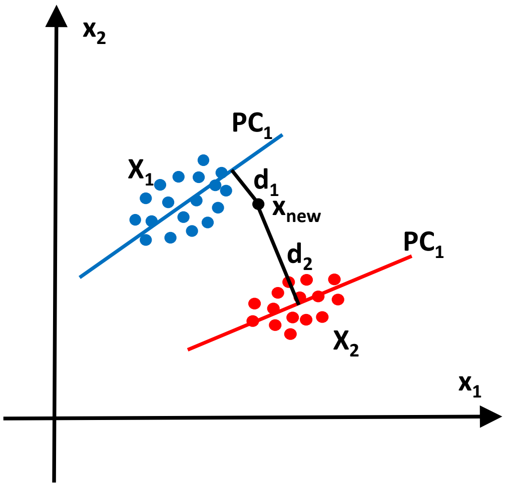


Figure 3.1: An illustrative example of NSM in a 2D space.

An illustrative example of a PC subspace and NSM is shown in a 2D space in Figure 3.1. The blue and red straight lines indicate the first PCs of the two classes, respectively. If we set $r_1 = r_2 = 1$, then the distances from $\boldsymbol{x}_{new}$ to the two class subspaces are shown as d_1 and d_2, respectively. In this example, we assign $\boldsymbol{x}_{new}$ to class 1 since $d_1 < d_2$.

3.1.2 Geometric convex model representation

There are two major differences between the PC subspace representation and the geometric convex model representation. First, the PC subspace is spanned by PCs which are the linear combinations of the original features, while the geometric convex model is constructed by the linear combinations of the class samples. To be more specific, the PC subspace is spanned by a set of vectors in $\boldsymbol{W}_k$, which are linear combinations of the original features in $\boldsymbol{X}_k$, i.e. the columns of $\boldsymbol{X}_k$. In contrast, the geometric convex model is for the linear combinations of the rows of $\boldsymbol{X}_k$.

Second, since there is no constraints on the linear combination, the PC subspace representation has weak information about the location of the class samples. However, the geometric convex model representation imposes constraints on the linear combination of the training samples, providing more restricted areas for class representation.

Here we introduce the nearest convex hull method (NCHM) and the nearest convex cone method (NCCM), both based on the geometric convex model representation.

3.1.2.1 Nearest convex hull method (NCHM)

Nalbantov et al. (2006) propose the NCHM. We define convex set and convex hull as follows.

Definition 3.1.2. *Convex set.* A set K in a linear vector space is said to be convex if, given $\boldsymbol{x}_1, \boldsymbol{x}_2 \in K$, all points of the form $\alpha \boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$ with $0 \leq \alpha \leq 1$ are in K.

Definition 3.1.3. *Convex hull.* Let $S = \{\boldsymbol{x}_i\}_{i=1}^N$ be an arbitrary set in a linear vector space. The convex hull, $ch(S) = \{\boldsymbol{z} \,:\, \boldsymbol{z} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \mid \boldsymbol{x}_i \in S,\ 0 \leq \alpha_i \leq 1,\ \sum_{i=1}^{N} \alpha_i = 1\}$, is the smallest convex set containing S. In other words, $ch(S)$ is the intersection of all convex sets containing S.

Given the training samples $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times p}$ of class k, the convex hull built by $\boldsymbol{X}_k$ is the set of vectors $\boldsymbol{z} \in \mathbb{R}^p$:

$$ch(\boldsymbol{X}_k) = \{\boldsymbol{z} \,:\, \boldsymbol{z} = \boldsymbol{X}_k^T \boldsymbol{\alpha}_k^{CH} \mid 0 \leq \boldsymbol{\alpha}_k^{CH} \leq 1,\ \mathbf{1}^T \boldsymbol{\alpha}_k^{CH} = 1\}, \tag{3.5}$$

where $\boldsymbol{\alpha}_k^{CH} \in \mathbb{R}^{n_k \times 1}$ is a vector containing the coefficients associated with the n_k training samples in $\boldsymbol{X}_k$, $0 \leq \boldsymbol{\alpha}_k^{CH} \leq 1$ means each element are in $[0,1]$, and $\mathbf{1} \in \mathbb{R}^{n_k \times 1}$ has all elements of one.

Given a new sample $\boldsymbol{x}_{new} \in \mathbb{R}^{1\times p}$, the distance from $\boldsymbol{x}_{new}$ to the convex hull $ch(\boldsymbol{X}_k)$ of the kth class is

$$
\begin{aligned}
d_k^{CH} &= \min_{\boldsymbol{\alpha}_k^{CH}} \ ||\boldsymbol{x}_{new} - (\boldsymbol{X}_k^T \boldsymbol{\alpha}_k^{CH})^T||_2, \\
\text{s.t. } & 0 \leq \boldsymbol{\alpha}_k^{CH} \leq 1, \ \mathbf{1}^T \boldsymbol{\alpha}_k^{CH} = 1.
\end{aligned}
\tag{3.6}
$$

Then $\boldsymbol{x}_{new}$ is assigned to the class with the smallest d_k^{CH}:

$$
\hat{y}^{CH} = \operatorname*{argmin}_k d_k^{CH}, \tag{3.7}
$$

where $\hat{y}^{CH}$ denotes the predicted label for $\boldsymbol{x}_{new}$ by NCHM.

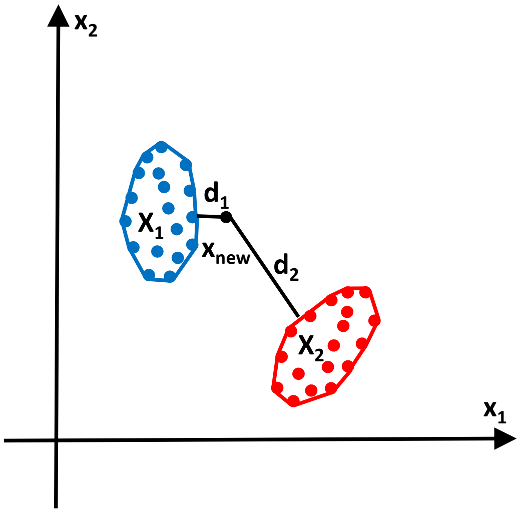


Figure 3.2: An illustrative example of NCHM in a 2D space.

An illustrative example of NCHM is shown in a 2D space in Figure 3.2. The convex hulls of the two classes are shown as the blue and red polygons, respectively. Since $d_1 < d_2$, we assign $\boldsymbol{x}_{new}$ to class 1 in this example.

3.1.2.2 Nearest convex cone method (NCCM)

In NCCM, we define cone, convex cone and convex polyhedral cone as follows.

Definition 3.1.4. *Cone.* A set C in a linear vector space is said to be a cone with vertex at the origin if $\boldsymbol{x}$ in C implies that $\alpha\boldsymbol{x} \in C$ for all $\alpha \geq 0$.

Definition 3.1.5. *Convex polyhedral cone.* A set C is a convex cone if it is a cone and is convex. A convex polyhedral cone is a convex cone that is generated by a finite number of generators. Let $S = \{\boldsymbol{x}_i\}_{i=1}^{N}$ be an arbitrary set in a linear vector space. The set, $cc(S) = \{\boldsymbol{z} \ : \ \boldsymbol{z} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \mid \boldsymbol{x}_i \in S, \alpha_i \geq 0\}$, is the convex polyhedral cone generated by S.

Given the training samples $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times p}$ of class k, the convex polyhedral cone built by $\boldsymbol{X}_k$ is defined as a set of vectors $\boldsymbol{z} \in \mathbb{R}^p$:

$$cc(\boldsymbol{X}_k) = \{\boldsymbol{z} : \boldsymbol{z} = \boldsymbol{X}_k^T \boldsymbol{\alpha}_k^{CC} \mid \boldsymbol{\alpha}_k^{CC} \geq 0\}, \tag{3.8}$$

where $\boldsymbol{\alpha}_k^{CC} \in \mathbb{R}^{n_k \times 1}$ and $\boldsymbol{\alpha}_k^{CC} \geq 0$ means each element in $\boldsymbol{\alpha}_k^{CC}$ is nonnegative. Thus each vector in $cc(\boldsymbol{X}_k)$ is a conical combination of the samples in $\boldsymbol{X}_k$.

To assign a new sample $\boldsymbol{x}_{new} \in \mathbb{R}^{1 \times p}$ to one of the classes, we calculate the distance from $\boldsymbol{x}_{new}$ to $cc(\boldsymbol{X}_k)$:

$$d_k^{CC} = \min_{\boldsymbol{\alpha}_k^{CC}} ||\boldsymbol{x}_{new} - (\boldsymbol{X}_k^T \boldsymbol{\alpha}_k^{CC})^T||_2, \ \text{ s.t. } \ \boldsymbol{\alpha}_k^{CC} \geq 0. \tag{3.9}$$

Then $\boldsymbol{x}_{new}$ is assigned to the class with the minimum d_k^{CC}:

$$\hat{y}^{CC} = \operatorname*{argmin}_k d_k^{CC}, \tag{3.10}$$

where $\hat{y}^{CC}$ denotes the predicted label for $\boldsymbol{x}_{new}$ by NCCM.

An illustrative example of NCCM is shown in a 2D space in Figure 3.3. The convex cones for the two classes are shown as the blue and red triangular area, respectively. Since $d_1 < d_2$, we assign $\boldsymbol{x}_{new}$ to class 1 in this example.

3.2 Dual analysis of the minimum distance problems

The minimum distance problems (3.2), (3.6) and (3.9) play key roles in NSM, NCHM and NCCM. However, the underlying classification mechanism of the minimum distance problems are barely explored theoretically in literature, which makes it difficult to explain their classification performances on certain datasets. To make

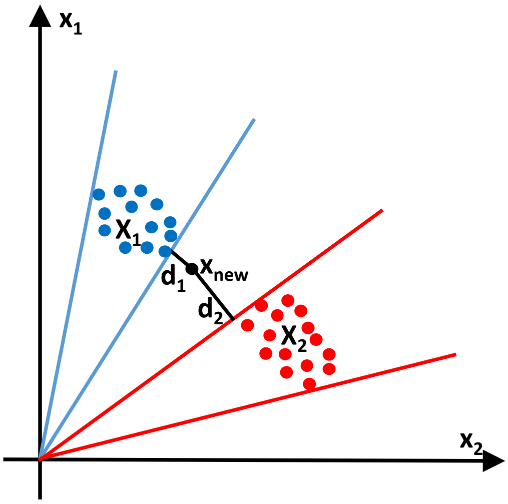


Figure 3.3: An illustrative example of NCCM in a 2D space.

the analysis and comparison of NSM, NCHM and NCCM easier, we aim to find the sets of separating hyperplanes associated with each methods. The separating hyperplanes could largely assist the understanding of the classification methods.

Dual analysis of the minimum distance problems enables us to find the separating hyperplanes, such that finding the minimum distance from a sample to a class model is equivalent to find the maximum distance from that sample to a separating hyperplane. Different from the Euclidean space settings used in the previous section, we discuss more general cases in the linear vector space with arbitrary norm in this section. Examples and illustrations for the Hilbert space are also discussed for better geometric understanding.

We first introduce some essential definitions related to the dual analysis and define the hyperplane properly. Then we show the dual analysis for the three minimum distance problems (3.2), (3.6) and (3.9). The dual analysis of minimum distance to the subspace and the convex hull could be found in Luenberger (1969) and we only show their results here. We show a detailed proof of the duality theorem of minimum distance to the convex cone based on an observation of the relationship between a convex cone and its polar cone.

3.2.1 Preliminary

Definition 3.2.1. *Normed linear vector space.* A normed linear vector space is a vector space $\mathscr{X}$, on which a real-valued function is defined to map each element $\boldsymbol{x}$ in $\mathscr{X}$ into a real number $||\boldsymbol{x}||$ called the norm of $\boldsymbol{x}$. The norm satisfies the following axioms:

1. $||\boldsymbol{x}|| \geq 0$ for all $\boldsymbol{x} \in \mathscr{X}$, $||\boldsymbol{x}|| = 0$ if and only if $\boldsymbol{x} = 0$.

2. $||\boldsymbol{x}+\boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$ for each $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}$.

3. $||\alpha \boldsymbol{x}|| = |\alpha| ||\boldsymbol{x}||$ for all scalar α and each $\boldsymbol{x} \in \mathscr{X}$.

Definition 3.2.2. *Linear functional.* A transformation from a vector space $\mathscr{X}$ into the space of real scalars is said to be a functional on $\mathscr{X}$. A functional f on a vector space $\mathscr{X}$ is linear if for any two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}$ and any two scalars α, β there holds $f(\alpha\boldsymbol{x}+\beta\boldsymbol{y}) = \alpha f(\boldsymbol{x}) + \beta f(\boldsymbol{y})$.

Definition 3.2.3. *The normed dual space.* Let $\mathscr{X}$ be a normed linear vector space. The space of all bounded linear functionals on $\mathscr{X}$ is called the normed dual of $\mathscr{X}$ and is denoted by $\mathscr{X}^*$. The norm of an element $f \in \mathscr{X}^*$ is $||f|| = \sup_{||x|| \leq 1} \; |f(x)|$.

Following Luenberger (1969), we use $\boldsymbol{x}^*$ to denote the linear functionals and write $\langle \boldsymbol{x}, \boldsymbol{x}^* \rangle$ to denote $f(\boldsymbol{x})$.

Definition 3.2.4. *Real inner space.* A real inner space is a real linear vector space $\mathscr{X}$ together with an inner product, which is a map from $\mathscr{X} \times \mathscr{X}$ to $\mathscr{R}$ and denoted by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ where $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}$. The inner product satisfies the following axioms:

1. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$.

2. $\langle \boldsymbol{x}+\boldsymbol{y}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \langle \boldsymbol{y}, \boldsymbol{z} \rangle$.

3. $\langle \lambda \boldsymbol{x}, \boldsymbol{y} \rangle = \lambda \langle \boldsymbol{x}, \boldsymbol{y} \rangle$, where λ is a constant.

4. $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$; $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x}$ is the origin.

Definition 3.2.5. *Hilbert space.* A complete real inner space is called a real Hilbert space.

A Hilbert space has the following nice property.

Theorem 3.2.1 (Luenberger (1969))**.** *If $\boldsymbol{x}^*$ is a bounded linear functional on a Hilbert space $\mathscr{H}$, there exists a unique vector $\mathbf{w} \in \mathscr{H}$ such that for all $\boldsymbol{x} \in \mathscr{H}$, $\langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = \langle \boldsymbol{x}, \boldsymbol{w} \rangle$. Moreover, we have $||\boldsymbol{x}^*|| = ||\boldsymbol{w}||$ and every $\mathbf{w}$ determines a unique bounded linear functional in this way.*

3.2.2 Hyperplane

Based on the above definitions, we define a hyperplane as follows and show some properties of a hyperplane that relates the primal problem with the dual problem.

Definition 3.2.6. *Hyperplane.* The translation of a subspace is said to be a linear variety. A hyperplane H in a linear vector space $\mathscr{X}$ is a maximal proper linear variety, that is, a linear variety H such that $H \neq \mathscr{X}$, and if V is any linear variety containing H, then either $V = \mathscr{X}$ or $V = H$.

Proposition 1. *Let H be a hyperplane in a linear vector space $\mathscr{X}$. Then there is a linear functional f on $\mathscr{X}$ and a constant c such that $H = \{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = c\}$. Conversely, if f is a nonzero linear functional on $\mathscr{X}$, the set $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = c\}$ is a hyperplane in $\mathscr{X}$. H is closed for every c if and only if f is continuous.*

As stated in Proposition 1, hyperplanes have a close relationship with linear functionals. Thus the primal problem can be related with the dual problem by using the hyperplane as a media.

For a closed hyperplane H, we define two closed half-spaces: the negative half-space $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \leq c\}$ and the positive half-space $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \geq c\}$. The distance from a point to a hyperplane is of great importance in dual analysis, thus we introduce it in Theorem 3.2.2.

Theorem 3.2.2 ((Zhou et al., 2002))**.** *Let $\boldsymbol{x}_e$ be an element in a real normed linear space $\mathscr{X}$ and let d denote its distance from the hyperplane H: $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = c\}$. Then,*

$$d = \inf_{\boldsymbol{h} \in H} ||\boldsymbol{x}_e - \boldsymbol{h}|| = \frac{|\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle - c|}{||\boldsymbol{x}^*||}.$$

3.2.3 Dual analysis for NSM, NCHM and NCCM

3.2.3.1 Dual analysis of the minimum distance problem (3.2)

Theorem 3.2.3 (Luenberger (1969))**.** *Let $\boldsymbol{x}_e$ be an element in a real normed linear space $\mathscr{X}$ and let d denote its distance from the subspace $\mathscr{M}$. Suppose the orthogonal complement of $\mathscr{M}$ is $\mathscr{M}^{\perp}$. Then,*

$$d = \inf_{\boldsymbol{m}\in\mathscr{M}} ||\boldsymbol{x}_e - \boldsymbol{m}|| = \max_{||\boldsymbol{x}^*||\leq 1, \boldsymbol{x}^*\in\mathscr{M}^{\perp}} \langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle, \tag{3.11}$$

where the maximum on the right is achieved for some $\boldsymbol{x}_0^* \in \mathscr{M}^{\perp}$.

If the infimum on the left is achieved for some $\boldsymbol{m}_0 \in \mathscr{M}$, then $\boldsymbol{x}_0^*$ is aligned with $\boldsymbol{x}_e - \boldsymbol{m}_0$, i.e. $\langle \boldsymbol{x}_e - \boldsymbol{m}_0, \boldsymbol{x}_0^* \rangle = ||\boldsymbol{x}_e - \boldsymbol{m}_0||||\boldsymbol{x}_0^*||$.

Based on Theorem 3.2.2, the right-hand side of (3.11) can be explained as the maximum distance from $\boldsymbol{x}_e$ to the hyperplane $H_{sub} = \{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0 \mid \boldsymbol{x}^* \in \mathscr{M}^{\perp}\}$, since the maximum is achieved when $||\boldsymbol{x}^*|| = 1$. Thus Theorem 3.2.3 could be understood as: The minimum distance from a point $\boldsymbol{x}_e$ to the subspace $\mathscr{M}$ is equivalent to the maximum distance from $\boldsymbol{x}_e$ to the hyperplane H_{sub}.

For a better geometric understanding, we discuss Theorem 3.2.3 in the Hilbert space. Based on Theorem 3.2.1, the dual space of a Hilbert space is itself. For each $\boldsymbol{x}^*$, we could find a unique $\boldsymbol{w} \in \mathscr{H}$ which is the normal vector of H_{sub}. Replace $\boldsymbol{x}^*$ by $\boldsymbol{w}$, the right-hand side of (3.11), i.e. $\langle \boldsymbol{x}_e, \boldsymbol{w} \rangle$, still denotes the distance from $\boldsymbol{x}_e$ to H_{sub} since the maximum is achieved for $||\boldsymbol{w}|| = ||\boldsymbol{x}_e|| = 1$. We also have $\langle \boldsymbol{x}_e - \boldsymbol{m}_0, \boldsymbol{w}_0 \rangle = ||\boldsymbol{x}_e - \boldsymbol{m}_0||||\boldsymbol{w}_0||$, thus $\boldsymbol{x}_e - \boldsymbol{m}_0 = \mu \boldsymbol{w}_0$ ($\mu > 0$). For any vector $\boldsymbol{m} \in \mathscr{M}$, $\langle \boldsymbol{x}_e - \boldsymbol{m}_0, \boldsymbol{m} \rangle = \langle \mu \boldsymbol{w}_0, \boldsymbol{m} \rangle = \mu \langle \boldsymbol{w}_0, \boldsymbol{m} \rangle = 0$, as $\boldsymbol{w}_0 \in \mathscr{M}^{\perp}$. This indicates that $\boldsymbol{x}_e - \boldsymbol{m}_0$ has the same direction as $\boldsymbol{w}_0$ and $\boldsymbol{x}_e - \boldsymbol{m}_0$ is perpendicular to $\mathscr{M}$.

Figure 3.4 shows an illustrative example of Theorem 3.2.3. Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are the orthogonal bases for $\mathscr{R}^3$. Assume $\boldsymbol{x}_e$ lies in the subspace spanned by $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ and $\mathscr{M}$ is the subspace spanned by $\boldsymbol{x}_2$. Thus $\mathscr{M}^{\perp}$ is the subspace spanned by $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$. Then the minimum distance from $\boldsymbol{x}_e$ to $\mathscr{M}$ is achieved at the point $\boldsymbol{m}_0$; and the maximum distance from $\boldsymbol{x}_e$ to H_{sub} with normal vectors in $\mathscr{M}^{\perp}$ is attained when $\boldsymbol{w}_0$ has the same direction as $\boldsymbol{x}_3$. We can find that these two distances are the

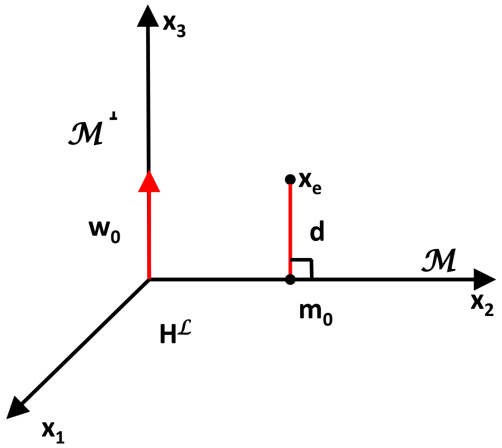


Figure 3.4: An illustrative example Theorem 3.2.3.

same, both equal to d. The hyperplane $H^{\mathscr{L}}$ with the normal vector $\boldsymbol{w}_0$ is actually the subspace spanned by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The vector $\boldsymbol{x}_e - \boldsymbol{m}_0$ has the same direction as $\boldsymbol{w}_0$. This result is clear with simple geometry, if we treat $\boldsymbol{m}_0$ as the orthogonal projection of $\boldsymbol{x}_e$ on the subspace $\mathscr{M}$.

3.2.3.2 Dual analysis of the minimum distance problem (3.6)

Theorem 3.2.4 (Luenberger (1969)). *Let $\boldsymbol{x}_e$ be a point in a real normed vector space X and let $d > 0$ denote its distance from the convex set K having support functional h, i.e. $h(\boldsymbol{x}^*) = \sup_{\boldsymbol{k} \in K} \langle \boldsymbol{k}, \boldsymbol{x}^* \rangle$. Then*

$$d = \inf_{\boldsymbol{k} \in K} ||\boldsymbol{x}_e - \boldsymbol{k}|| = \max_{||\boldsymbol{x}^*|| \leq 1} [\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle - h(\boldsymbol{x}^*)], \tag{3.12}$$

where the maximum on the right is achieved by some $\boldsymbol{x}_0^* \in \mathscr{X}^*$.

If the infimum on the left is achieved by some $\boldsymbol{k}_0 \in K$, then $\boldsymbol{x}_0^*$ is aligned with $\boldsymbol{x}_e - \boldsymbol{k}_0$, i.e. $\langle \boldsymbol{x}_e - \boldsymbol{k}_0, \boldsymbol{x}_0^* \rangle = ||\boldsymbol{x}_e - \boldsymbol{k}_0|| ||\boldsymbol{x}_0^*||$.

The right-hand side of (3.12) can be understood as the maximum distance from $\boldsymbol{x}_e$ to the hyperplane $H^{CH} = \{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = h(\boldsymbol{x}^*)\}$. Thus Theorem 3.2.4 indicates that the minimum distance from $\boldsymbol{x}_e$ to the convex hull is equivalent to the maximum distance from $\boldsymbol{x}_e$ to the hyperplane H^{CH}.

In the Hilbert space, we could find a unique $\boldsymbol{w}_0 \in \mathscr{H}$ for $\boldsymbol{x}_0^*$. Since $\boldsymbol{x}_0^*$ is aligned with $\boldsymbol{x}_e - \boldsymbol{k}_0$, $\boldsymbol{x}_e - \boldsymbol{k}_0 = \mu \boldsymbol{w}_0$ ($\mu > 0$) and $\boldsymbol{x}_e - \boldsymbol{k}_0$ has the same direction as $\boldsymbol{w}_0$.

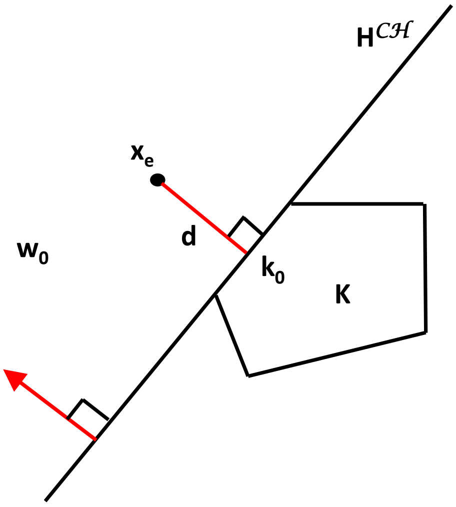


Figure 3.5: An illustrative example of Theorem 3.2.4.

Figure 3.5 shows an intuitive example of Theorem 3.2.4 in $\mathbb{R}^2$. The minimum distance from $\boldsymbol{x}_e$ to K is achieved at point $\boldsymbol{k}_0$, which lies on the nearest face of K to $\boldsymbol{x}_e$. The maximum distance between $\boldsymbol{x}_e$ and H^{CH} that separates $\boldsymbol{x}_e$ and K is achieved when the nearest face of K to $\boldsymbol{x}_e$ is in H^{CH}. The normal vector $\boldsymbol{w}_0$ is perpendicular to H^{CH} and has the same direction as $\boldsymbol{x}_e - \boldsymbol{k}_0$.

3.2.3.3 Dual analysis of the minimum distance problem (3.9)

Inspired by the relationship between $\mathscr{M}$ and $\mathscr{M}^{\perp}$ used in Theorem 3.2.3, we apply the relationship between a convex cone and its polar cone to the dual analysis of (3.9) and obtain Theorem 3.2.5. We first introduce the definition of a polar cone and then show Theorem 3.2.5 and its proof.

Definition 3.2.7. *Polar cone.* Given a convex polyhedral cone C in a normed space $\mathscr{X}$, the set $C^p = \{\boldsymbol{x}^* \in \mathscr{X}^* : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \leq 0, \, \forall \boldsymbol{x} \in C\}$ is called the polar cone of C.

If $\boldsymbol{x}_e$ is an interior point of C, then $d = 0$, which is a trivial case. Thus in the following theorem, we discuss the case when $\boldsymbol{x}_e$ is not an interior point of C with $d > 0$.

Theorem 3.2.5. *Let $\boldsymbol{x}_e$ be an element in a real normed linear space $\mathscr{X}$. Let $d > 0$*

denote the distance from $\boldsymbol{x}_e$ to the convex cone C. Then,

$$d = \inf_{\boldsymbol{c}\in C} ||\boldsymbol{x}_e - \boldsymbol{c}|| = \max_{||\boldsymbol{x}^*||\leq 1, \boldsymbol{x}^*\in C^p} \langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle,$$

where the maximum on the right is achieved for some $\boldsymbol{x}_0^* \in C^p$.

If the infimum on the left is achieved for some $\boldsymbol{c}_0 \in C$, then $\boldsymbol{x}_0^*$ is aligned with $\boldsymbol{x}_e - \boldsymbol{c}_0$, i.e. $\langle \boldsymbol{x}_e - \boldsymbol{c}_0, \boldsymbol{x}_0^* \rangle = ||\boldsymbol{x}_e - \boldsymbol{c}_0|| ||\boldsymbol{x}_0^*||$.

Proof. We first show that there exist some $\boldsymbol{x}^* \in C^p$ with the hyperplane $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0\}$ being able to separate $\boldsymbol{x}_e$ and C. The two closed half-spaces associated with the hyperplane $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0\}$ are $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \geq 0\}$ and $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle \leq 0\}$. When $\boldsymbol{x}^* \in C^p$, $\langle \boldsymbol{c}, \boldsymbol{x}^* \rangle \leq 0$ for $\boldsymbol{c} \in C$, and C is in the negative half-space. Since $\boldsymbol{x}_e$ is not an interior point of C, we could find some $\boldsymbol{x}^* \in C^p$ such that $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle \geq 0$ and $\boldsymbol{x}_e$ is in the positive half-space. Thus $\boldsymbol{x}_e$ and C lie in opposite half-spaces determined by the hyperplane $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0\}$ with $\boldsymbol{x}^* \in C^p$.

Let $S(\varepsilon)$ be the sphere centred at $\boldsymbol{x}_e$ of radius ε. For $\boldsymbol{x}^* \in C^p$ having $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle \geq 0$ and $||\boldsymbol{x}^*|| = 1$, let ε^* be the supremum of the ε's for which the hyperplane $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0\}$ separates C and $S(\varepsilon)$. It is clear that $0 \leq \varepsilon^* \leq d$. Also $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle = \varepsilon^*$ when $||\boldsymbol{x}^*|| = 1$. Thus, for every $\boldsymbol{x}^* \in C^p$ having $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle \geq 0$ and $||\boldsymbol{x}^*|| = 1$, we have $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle \leq d$.

On the other hand, since C contains no interior point of $S(d)$, there is a hyperplane separating C and $S(d)$, and thus an $\boldsymbol{x}_0^* \in C^p$ such that $\langle \boldsymbol{x}_e, \boldsymbol{x}^* \rangle = d$.

To prove the alignment statement, suppose $\boldsymbol{c}_0 \in C$ and $||\boldsymbol{x}_e - \boldsymbol{c}_0|| = d$. Since $\boldsymbol{c}_0 \in C$, $\langle \boldsymbol{c}_0, \boldsymbol{x}_0^* \rangle \leq 0$ and $\langle \boldsymbol{x}_e - \boldsymbol{c}_0, \boldsymbol{x}_0^* \rangle \geq \langle \boldsymbol{x}_e, \boldsymbol{x}_0^* \rangle = d$. However, according to the Cauchy-Schwarz inequality, $\langle \boldsymbol{x}_e - \boldsymbol{c}_0, \boldsymbol{x}_0^* \rangle \leq ||\boldsymbol{x}_e - \boldsymbol{c}_0|| ||\boldsymbol{x}_0^*|| = d$. Thus $\langle \boldsymbol{x}_e - \boldsymbol{c}_0, \boldsymbol{x}_0^* \rangle = ||\boldsymbol{x}_e - \boldsymbol{c}_0|| ||\boldsymbol{x}_0^*|| = d$ and $\boldsymbol{x}_0^*$ is aligned with $\boldsymbol{x}_e - \boldsymbol{c}_0$. □

Theorem 3.2.5 indicates that the minimum distance between $\boldsymbol{x}_e$ and C is equivalent to the maximum distance between $\boldsymbol{x}_e$ and the hyperplane $H^{CC} = \{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}^* \rangle = 0 \mid \boldsymbol{x}^* \in C^p, ||\boldsymbol{x}^*|| = 1\}$ that separates $\boldsymbol{x}_e$ and C.

In the Hilbert space, we could find a unique $\boldsymbol{w}_0 \in \mathscr{H}$ for $\boldsymbol{x}_0^*$. Substituting $\boldsymbol{w}_0$ with $\boldsymbol{x}_0^*$, we could get $\langle \boldsymbol{x}_e, \boldsymbol{w}_0 \rangle = d$. Also $\langle \boldsymbol{x}_e - \boldsymbol{c}_0, \boldsymbol{w}_0 \rangle = ||\boldsymbol{x}_e - \boldsymbol{c}_0|| ||\boldsymbol{w}_0|| = d$. The equality holds when $\boldsymbol{x}_e - \boldsymbol{c}_0 = \mu \boldsymbol{w}_0$ $(\mu > 0)$. Thus we could get the following two conclusions. First, $\langle \boldsymbol{c}_0, \boldsymbol{w}_0 \rangle = 0$, which indicates that $\boldsymbol{c}_0$ and $\boldsymbol{w}_0$ are orthogonal. Second, $\boldsymbol{x}_e = \boldsymbol{c}_0 + \mu \boldsymbol{w}_0$, which indicates that $\boldsymbol{x}_e$ could be decomposed to $\boldsymbol{c}_0 \in C$ and $\mu \boldsymbol{w} \in C^p$. These two conclusions indicates that the orthogonal decompositions of $\boldsymbol{x}_e$ to C and C^p are $\boldsymbol{c}_0$ and $\mu \boldsymbol{w}_0$, respectively. Based on the Moreau's theorem in the Hilbert space stated below, $\boldsymbol{c}_0$ and $\mu \boldsymbol{w} \in C^p$ are the projections of $\boldsymbol{x}_e$ on C and C^p, respectively.

Theorem 3.2.6 (Moreau (1962))**.** *Let C be a nonempty closed convex cone in $\mathscr{H}$, and let $\boldsymbol{x} \in \mathscr{H}$. Then the following statements are equivalent:*

1. $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{z}$, $\boldsymbol{y} \in C$, $\boldsymbol{z} \in C^p$ *and* $\langle \boldsymbol{y}, \boldsymbol{z} \rangle = 0$,

2. $\boldsymbol{y} = \mathscr{P}_C \boldsymbol{x}$ *and* $\boldsymbol{z} = \mathscr{P}_{C^p} \boldsymbol{x}$,

where $\mathscr{P}_C$ and $\mathscr{P}_{C^p}$ denote the projection operators onto C and C^p, respectively.

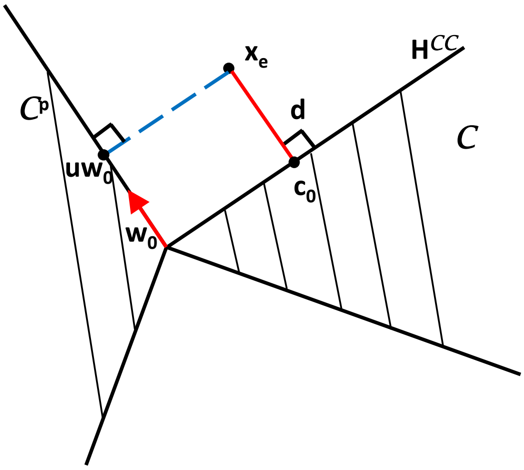


Figure 3.6: An illustrative example of Theorem 3.2.5.

Figure 3.6 illustrates Theorem 3.2.5 in $\mathbb{R}^2$. The minimum distance d from $\boldsymbol{x}_e$ to C is achieved by $\boldsymbol{c}_0$, which is the orthogonal projection of $\boldsymbol{x}_e$ to the nearest face of C to $\boldsymbol{x}_e$. The maximum distance from $\boldsymbol{x}_e$ to H^{CC} is achieved when H^{CC} contains the nearest face of C to $\boldsymbol{x}_e$. It is obvious that the distance from $\boldsymbol{x}_e$ to this H^{CC} is also d. The normal vector associated with this hyperplane is $\boldsymbol{w}_0$, which has the same direction as $\boldsymbol{x}_e - \boldsymbol{c}_0$; the point $\mu \boldsymbol{w}_0$ is the orthogonal projection of $\boldsymbol{x}_e$ to C^p.

3.2.4 A separating hyperplane classification (SHC) framework

The dual analysis enables us to explain the classification schemes of NSM, NCCM and NCHM from the separating hyperplane point of view. Theorems 3.2.3, 3.2.4 and 3.2.5 indicate that the three methods all classify a test sample by using a pair of separating hyperplanes in two-class classification. Note that in this chapter we focus on two-class classification; multi-class classification could be obtained without difficulty on the basis of two-class classification through applying the one-vs-one or one-vs-all strategy (Bishop, 2006).

Suppose $\boldsymbol{X}_k$, $H_k = \{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}_k^* \rangle = c_k\}$ denote the training set and the separating hyperplane for the kth class respectively. Hyperplane H_k separates the new sample $\boldsymbol{x}_{new}$ and the training set $\boldsymbol{X}_k$. The two separating hyperplanes, H_1 and H_2, divide the original feature space into four parts: 1) $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}_1^* \rangle \leq c_1$ and $\langle \boldsymbol{x}, \boldsymbol{x}_2^* \rangle \leq c_2\}$, 2) $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}_1^* \rangle \geq c_1$ and $\langle \boldsymbol{x}, \boldsymbol{x}_2^* \rangle \leq c_2\}$, 3) $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}_1^* \rangle \leq c_1$ and $\langle \boldsymbol{x}, \boldsymbol{x}_2^* \rangle \geq c_2\}$ and 4) $\{\boldsymbol{x} : \langle \boldsymbol{x}, \boldsymbol{x}_1^* \rangle \geq c_1$ and $\langle \boldsymbol{x}, \boldsymbol{x}_2^* \rangle \geq c_2\}$. A new sample $\boldsymbol{x}_{new}$ falls into one of the four parts. Figure 3.7 shows a simple example of the locations of $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ and $\boldsymbol{x}_{new}$ in the space divided by H_1 and H_2.

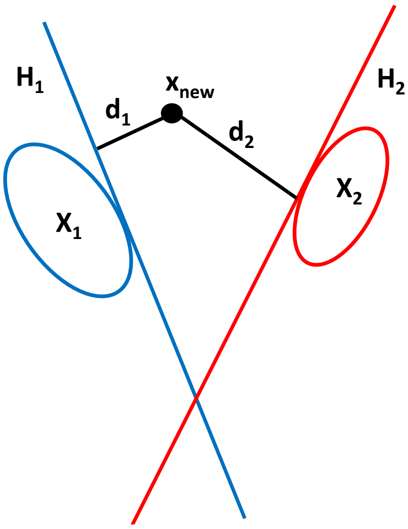


Figure 3.7: The separating hyperplane classification framework.

Based on the two separating hyperplanes, we could derive a separating hyperplane classification (SHC) framework for different class representation models and distances with arbitrary norms: First, for the kth class, we obtain

$$\begin{aligned} \max_{c_k, ||\boldsymbol{x}_k^*||=1} \quad & d_k = \langle \boldsymbol{x}_{new}, \boldsymbol{x}_k^* \rangle - c_k \\ \text{s.t. } & \text{constraint}(\boldsymbol{x}_k^*, c_k), \end{aligned} \tag{3.13}$$

where constraint$(\boldsymbol{x}_k^*, c_k)$ denotes constraints on $\boldsymbol{x}_k^*$ and c_k. Then, $\boldsymbol{x}_{new}$ is assigned to the class k with the minimum d_k.

This SHC framework for two-class classification can be explained as follows. For each test sample, we find a pair of separating hyperplanes that separate the test sample and the two class models, respectively. The test sample is then assigned to the class with the minimum distance from that sample to the corresponding hyperplane.

In NSM, NCHM and NCCM, the Euclidean norm $||\cdot||_2$ is used. We summarise constraint$(\boldsymbol{x}_k^*, c_k)$ for NSM, NCHM and NCCM in Table 3.1. Note that $\boldsymbol{x}_k^*$ is replaced by $\boldsymbol{w}_k$.

Table 3.1: constraint$(\boldsymbol{x}_k^*, c_k)$ for NSM, NCHM and NCCM.

NSM	NCHM	NCCM
$\langle \boldsymbol{x}_i^k \boldsymbol{P}_k, \boldsymbol{w}_k \rangle = 0$ $c_k = 0$	$\langle \boldsymbol{x}_{new}, \boldsymbol{w}_k \rangle \leq c_k$ $\langle \boldsymbol{x}_i^k, \boldsymbol{w}_k \rangle \geq c_k$	$\langle \boldsymbol{x}_{new}, \boldsymbol{w}_k \rangle \leq 0$ $\langle \boldsymbol{x}_i^k, \boldsymbol{w}_k \rangle \geq 0$ $c_k = 0$
$\boldsymbol{P}_k$ denotes the projection matrix for class k. $\boldsymbol{x}_i^k \in \mathbb{R}^{1\times p}$ denotes the ith row in $\boldsymbol{X}_k$.		

Besides the constraints listed in Table 3.1, other constraints could also be specified based on the properties of the dataset and the requirements from the user, to extend further.

In the SHC framework, the normal vectors of the separating hyperplanes plays important roles in classification. Theorems 3.2.3, 3.2.4 and 3.2.5 suggest that the dual function $\boldsymbol{x}_0^*$ that determines the separating hyperplane is aligned with the vector $\boldsymbol{x}_{new} - \boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is the nearest point to $\boldsymbol{x}_{new}$ in the class model. In the Hilbert space, this means that the normal vector of the separating hyperplane is parallel with $\boldsymbol{x}_{new} - \boldsymbol{x}_0$. The norm of $\boldsymbol{x}_{new} - \boldsymbol{x}_0$ is defined as the distance from $\boldsymbol{x}_{new}$ to the class model. Thus the discriminative information contained in the direction of $\boldsymbol{x}_{new} - \boldsymbol{x}_0$, which is also the direction of the associated normal vector of the hyperplane, is vital to classification. The more the discriminative information contained in the normal vector, the higher the classification accuracy. In other words, to get better classification, constraints should be specified to make the normal vector contain more discriminative information.

The SHC framework is not only restricted to the standard nearest-class-model methods. In the nearest-class-model methods, the between-class information is not used in classification since the class models are built independently. To further improve the classification performance, the discriminative between-class information could be imposed as constraints to get separating hyperplanes better for classification. In this way, we can actually build class models with information from all classes and make the class models more discriminative as desired. For a simple example, to find the hyperplane of class k for $\boldsymbol{x}_{new}$, we could add constraints into the optimisation problem (3.13) to force the training samples of class k and those of the other class to lie on the opposite sides of the hyperplane. With such additional constraints, the information from the other class can also help to find the hyperplane for class k.

Note that the SHC framework is different from SVM and its extensions that are based on a pair of separating hyperplanes, i.e. 1S-BFHC and 2S-BFHC (Cevikalp, 2016), GEPSVM (Mangasarian and Wild, 2006) and TSVM (Jayadeva et al., 2007). In SVM, only one separating hyperplane is determined for all test samples based on the information from two classes together. In 1S-BFHC, 2S-BFHC, GEPSVM and TSVM, one hyperplane is found for each class such that it is closer to the samples of one class while far from the samples from the other class; but as in SVM, the pair of hyperplanes are fixed for all test samples. Thus in SVM and its extensions with linear kernels, the classification boundary is linear. However, in our case, different test samples are associated with different pairs of hyperplanes, which makes the classification boundary nonlinear.

Although there are differences between the SHC framework and SVM and its extensions, the well studied strategies for finding better separating hyperplanes in SVM and its extensions could also be introduced to the SHC framework to get better classifiers. For example, the kernel tricks could be used to introduce nonlinearity and the SMO algorithm could be applied to solve the optimisation problems.

3.3 Experiments

In the following experiments, we show the classification and analysis of NSM, NCHM and NCCM on three real datasets: the fat dataset, the meat dataset and the Phenyl dataset.

3.3.1 Datasets

The fat dataset, the meat dataset and the Phenyl dataset are used in the experiments. Detailed descriptions of the three datasets can be found in Chapter 2 Section 2.2.1.

For the fat dataset, a training set contains 100 randomly selected samples, with 35 samples of less than 20% fat and 35 samples of more than 20% fat, and a test set contains the remaining samples.

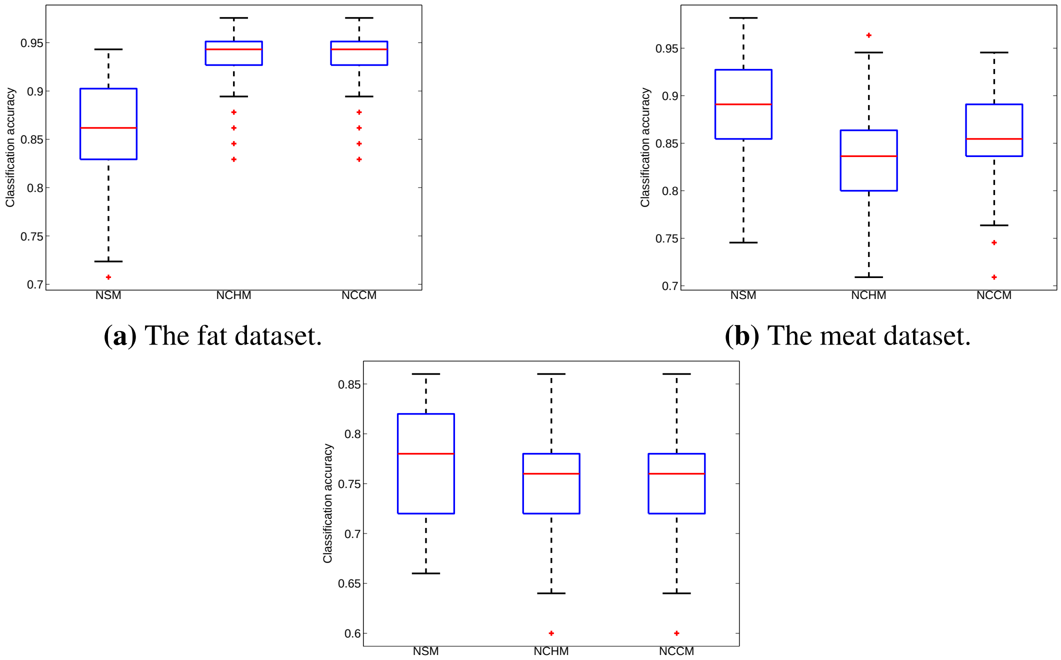
For the meat dataset, a training set contains 27 chicken samples and 27 turkey samples, and a test set contains 28 chicken samples and 27 turkey samples.

For the Phenyl dataset, 100 samples are randomly selected and used in the experiments. In the 100 samples, 50 samples contain the phenyl substructure and 50 samples do not contain the structure. A training set consists of 25 samples with the phenyl substructure and 25 without the substructure, and a test set consists of 25 with the phenyl substructure and 25 without the substructure.

3.3.2 Experiment settings

In NSM, the dimensions of the two class subspaces are tuned by 10-fold cross-validation on the training set. The dimensions are chosen to minimise the classification error. In NCHM, the optimisation problem (3.6) is solved using the 'cvx' package in MATLAB. In NCCM, the optimisation problem (3.9) is solved using the 'lsqnonneg' function in MATLAB. All the experiments are repeated 100 times and the classification accuracies of all the experiments are recorded and depicted in boxplots.

3.3.3 Classification Results



(a) The fat dataset.

(b) The meat dataset.

(c) The Phenyl dataset.

Figure 3.8: The classification accuracies of NSM, NCHM and NCCM on the three datasets.

The classification accuracies of NSM, NCHM and NCCM for the three datasets are shown in Figure 3.8. It is clear that their relative performances are different for different datasets.

For the fat dataset, it is clear that the geometric convex model representations (NCHM and NCCM) are better than the PC subspace representation (NSM) in classification, as shown in Figure 3.8a. However, for the meat and Phenyl datasets, the geometric convex models are worse than the PC model, as shown in Figure 3.8b and Figure 3.8c, respectively.

Two summaries could be drawn from Figure 3.8. First, the classification performances of NSM, NCHM and NCCM are data-dependant. Second, the performance of NCCM is between that of NSM and NCHM for all three datasets. This makes sense since the convex cone model is tighter than the PC subspace while looser than the convex hull model.

3.3.4 Analysis of classification results

Section 3.3.3 shows that the classification performances of NSM, NCHM and NCCM are data-dependant. To understand this pattern, we compare the normal vectors of the pairs of separating hyperplanes of the three methods. As discussed in Section 3.2.4, the more discriminative the normal vectors are, the higher the classification accuracy. However, it is hard to determine the discriminative ability of the normal vectors directly. In this section, we show the discriminative ability of the normal vectors through their relationships with PCs, whose discriminative ability could be readily determined by the classification performance (Zhu et al., 2017). If the direction of the normal vector is similar to those of the discriminative PCs, then the normal vector contains discriminative information.

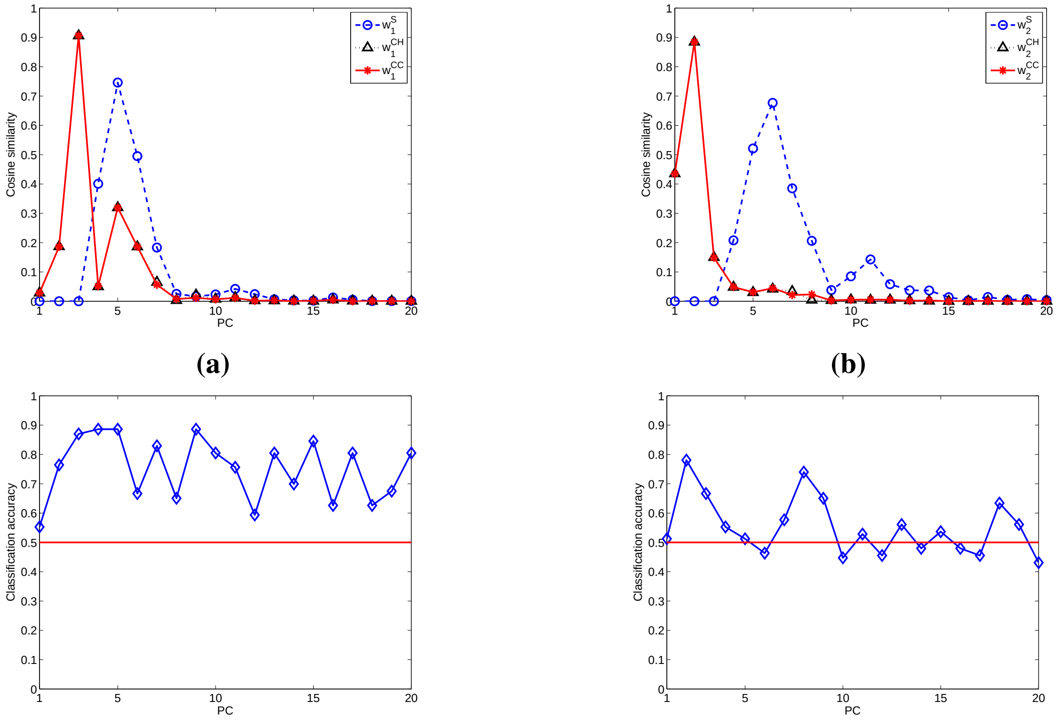
Here the discriminative ability of a PC is assessed by the classification accuracy of linear discriminative analysis (LDA) of the samples projected to that PC. The relationship between a normal vector $\boldsymbol{w} \in \mathbb{R}^{p\times 1}$ and a PC $\boldsymbol{v} \in \mathbb{R}^{p\times 1}$ is measured by their absolute cosine similarity:

$$\mathrm{sim}(\boldsymbol{w}, \boldsymbol{v}) = \frac{|\boldsymbol{w}^T \boldsymbol{v}|}{||\boldsymbol{w}||_2 ||\boldsymbol{v}||_2}.$$

3.3.4.1 The fat dataset

On the fat dataset, NCHM and NCCM provide better results than NSM (Figure 3.8a), which indicates that the separating hyperplanes found by NCHM and NCCM are better for classifying the fat dataset than those found by NSM. To further investigate and illustrate this, here we show an exemplar sample from the "less than 20%" class that is correctly classified by NCHM and NCCM, while wrongly classified by NSM. Each classification method is associated with a pair of separating hyperplanes with normal vectors $\boldsymbol{w}_k^{\mathrm{method}}$, where the superscript denotes the classification method and the subscript denotes the class.

To measure the relationship between the normal vectors and the PCs, we plot their cosine similarities against the first 20 PCs in Figure 3.9a and Figure 3.9b for the "less than 20%" subspace and the "more than 20%" subspace, respectively. The



(a) **(b)** **(c)** **(d)**

Figure 3.9: The discriminative ability of the normal vectors of NSM, NCHM and NCCM for the fat dataset. (a) and (b): the cosine similarities between the normal vectors and the PCs for the "less than 20%" subspace and the "more than 20%" subspace, respectively. (c) and (d): the discriminative ability of the PCs of the two subspaces.

higher the cosine similarity, the closer the directions of the normal vector and the PC. The overlapping curves of $\boldsymbol{w}_1^{\mathrm{CH}}$ and $\boldsymbol{w}_1^{\mathrm{CC}}$ (and those of $\boldsymbol{w}_2^{\mathrm{CH}}$ and $\boldsymbol{w}_2^{\mathrm{CC}}$) suggest that the normal vectors of the pair of separating hyperplanes in NCHM and NCCM to classify this test sample are the same. In contrast, the normal vectors of NSM are different from those of NCHM and NCCM, indicated by the blue dashed curve being quite different from the black and red curves.

Each curve in Figure 3.9a and Figure 3.9b shows a unique peak, i.e. the PC with the direction most similar to the normal vector, which can be used to assess the discriminative ability of the normal vectors. If the most similar PC is discriminative, then the normal vector is believed to be discriminative. The assessment of discriminative ability of the first 20 PCs is depicted in Figure 3.9c and Figure 3.9d, where the red horizontal lines indicates the classification accuracy of 0.5. The PCs with the classification accuracies above the red line are believed to be discriminative.

For the "less than 20%" subspace, by comparing Figure 3.9a and Figure 3.9c, we could observe that the normal vectors of the three methods are all discriminative; $\boldsymbol{w}_1^{\mathrm{CH}}$ and $\boldsymbol{w}_1^{\mathrm{CC}}$ have the most similar directions as the third PC (PC3), as shown in Figure 3.9a, and PC3 is highly discriminative, as shown in Figure 3.9c. Similarly, $\boldsymbol{w}_1^{\mathrm{S}}$ has the most similar direction as PC5, which is also highly discriminative.

However, for the "more than 20%" subspace, Figure 3.9b and Figure 3.9d show that the normal vectors of NSM are not discriminative, although the normal vectors of NCHM and NCCM remains discriminative: $\boldsymbol{w}_2^{\mathrm{S}}$ has the most similar direction to PC6, which has a classification accuracy less than 0.5 and is not discriminative.

Hence, for the fat dataset, we can suggest that the normal vectors of NCHM and NCCM are more discriminative than those of NSM, which explains to some extent why NSM performs worse than NCHM and NCCM (Figure 3.8a).

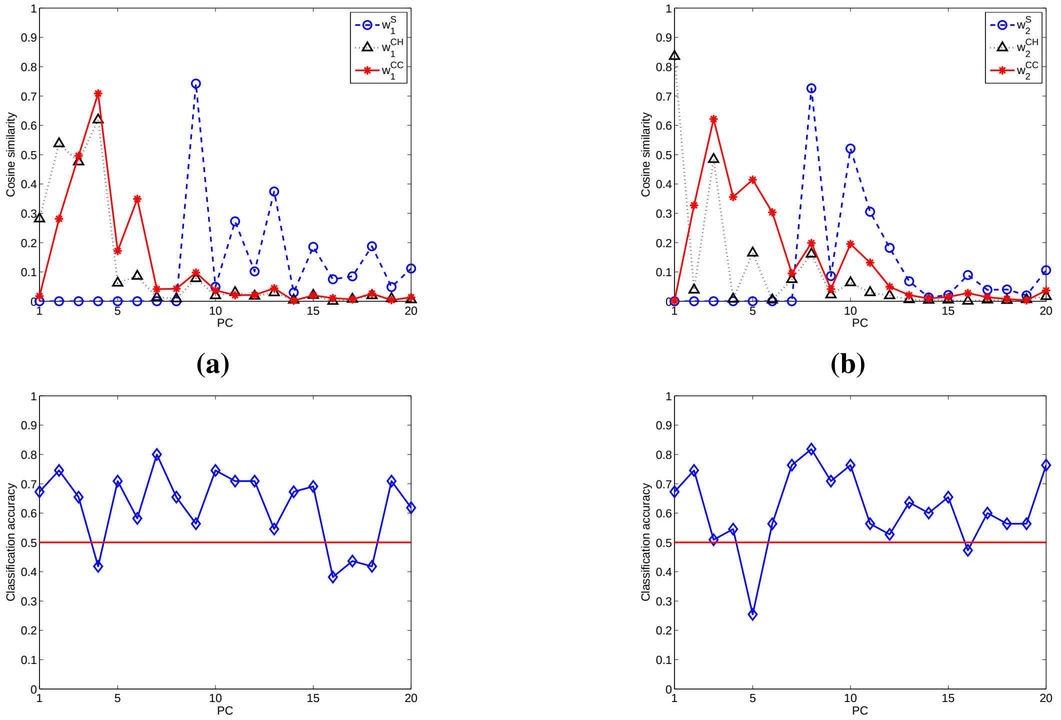
3.3.4.2 The meat dataset

The classification performance of NSM is better than those of NCHM and NCCM for the meat dataset (Figure 3.8b). As with the analysis in Section 3.3.4.1, Figure 3.10 shows the results for one meat sample that is correctly classified by NSM while wrongly classified by NCHM and NCCM.

For the chicken subspace, it is clear that $\boldsymbol{w}_1^{\mathrm{CH}}$ and $\boldsymbol{w}_1^{\mathrm{CC}}$ have the most similar directions to PC4, which is not discriminative, as shown in Figure 3.10a and Figure 3.10c. However, $\boldsymbol{w}_1^{\mathrm{S}}$ has the most similar direction to PC9, which is relatively discriminative compared with PC4.

The results for the turkey subspace are similar to those of the chicken subspace. For the turkey subspace, $\boldsymbol{w}_2^{\mathrm{S}}$ has the most similar direction to PC8 and PC10, which are very discriminative as indicated by their high classification accuracies. However, $\boldsymbol{w}_2^{\mathrm{CH}}$ has the most similar direction to PC1 and PC3, and $\boldsymbol{w}_2^{\mathrm{CC}}$ has the most similar direction to PC3 and PC5. Although PC1 is discriminative, PC3 and PC5 are not discriminative. Thus $\boldsymbol{w}_2^{\mathrm{CH}}$ and $\boldsymbol{w}_2^{\mathrm{CC}}$ are not as discriminative as $\boldsymbol{w}_2^{\mathrm{S}}$.

Considering all the above results, we can conclude that, for the meat dataset, the normal vector of NSM is more discriminative than those of NCHM and NCCM.



(a) **(b)**

(c) **(d)**

Figure 3.10: The discriminative ability of the normal vectors of NSM, NCHM and NCCM for the meat dataset. (a) and (b): the cosine similarities between the normal vectors and the PCs for the chicken subspace and the turkey subspace, respectively. (c) and (d): the discriminate ability of the PCs of the two subspaces.

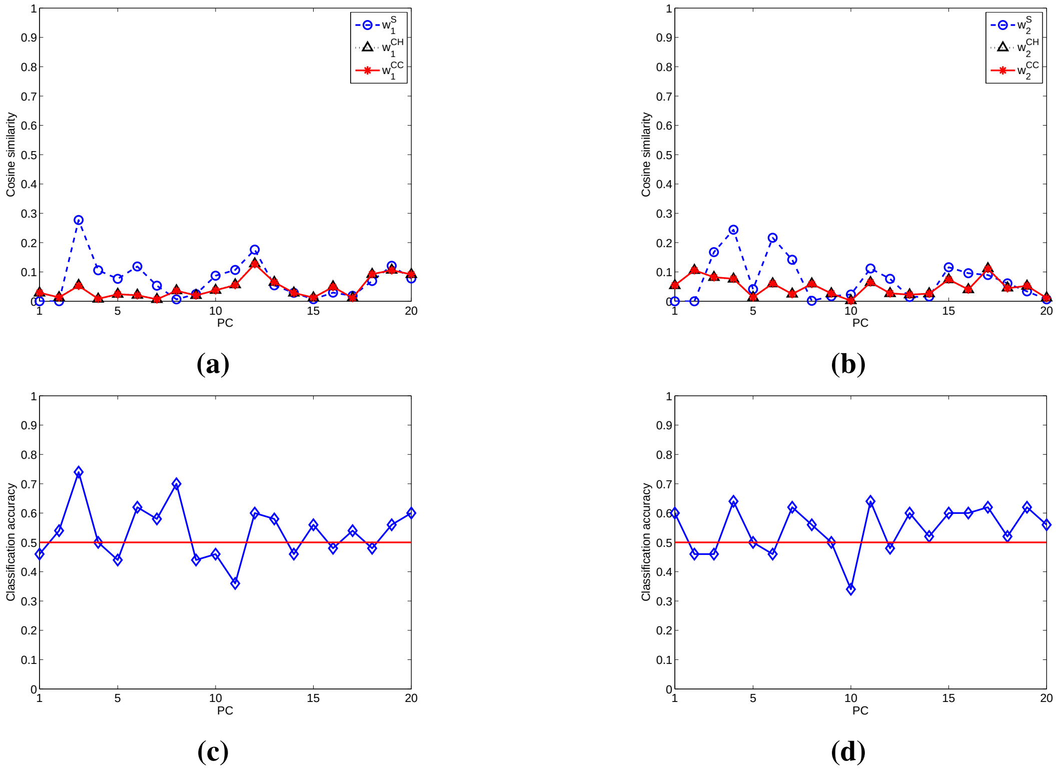
Therefore, it is reasonable that NSM performs better than NCHM and NCCM (Figure 3.8b).

3.3.4.3 The Phenyl dataset

For the Phenyl dataset, NSM performs slightly better than NCHM and NCCM (Figure 3.8c). Again, we show the results of an illustrative sample from the “with Phenyl structure” class that is correctly classified by NSM while wrongly classified by NCHM and NCCM in Figure 3.11.

Different from Figure 3.9 for the fat dataset and Figure 3.10 for the meat dataset, the normal vectors of NSM, NCHM and NCCM are not very similar to any of the PCs, as indicated by the low cosine similarities shown in Figure 3.11a and Figure 3.11b. Also the discriminative ability of the PCs are not high, i.e. around 0.5, as shown in Figure 3.11c and Figure 3.11d.

The curve of $\boldsymbol{w}_1^{\mathrm{S}}$ is very close to those of $\boldsymbol{w}_1^{\mathrm{CH}}$ and $\boldsymbol{w}_1^{\mathrm{CC}}$. However, $\boldsymbol{w}_1^{\mathrm{S}}$ is closer



(a) **(b)** **(c)** **(d)**

Figure 3.11: The discriminative ability of the normal vectors of NSM, NCHM and NCCM for the Phenyl dataset. (a) and (b): the cosine similarities between the normal vectors and the PCs for the “with Phenyl structure” subspace and the “without Phenyl structure” subspace, respectively. (c) and (d): the discriminate ability of the PCs of the two subspaces.

to PC3 (which has the highest classification accuracy) than $\boldsymbol{w}_1^{\mathrm{CH}}$ and $\boldsymbol{w}_1^{\mathrm{CC}}$. Similarly, the biggest difference between $\boldsymbol{w}_2^{\mathrm{S}}$ and $\boldsymbol{w}_2^{\mathrm{CH}}$ and $\boldsymbol{w}_2^{\mathrm{CC}}$ is that $\boldsymbol{w}_2^{\mathrm{S}}$ has slightly higher similarities with PC3, PC4, PC6 and PC7, with PC4 and PC7 slightly discriminative. Thus the slightly more discriminative ability of $\boldsymbol{w}_1^{\mathrm{S}}$ and $\boldsymbol{w}_2^{\mathrm{S}}$ makes NSM slightly better than NCHM and NCCM in classifying the Phenyl dataset (Figure 3.8c).

3.3.5 A scheme to analyse the data distributions

In this section, we would like to explore further the distribution properties of the three datasets, to show which properties make them suitable for one or more of the classification methods. We aim to check two properties of the data of each class: the variation of the data and the shape of the geometric convex class models.

The variational property is important to assess the data distribution. It indicates the most variable information in the data and could be easily found by the leading

PCs.

Since NCHM and NCCM are based on the geometric convex models, the samples that determine the geometric boundaries of the models are of great importance to describe the shapes of the geometric models. Since we aim to analyse the properties related to classification, we care more about the samples on the geometric boundaries that are related to classification, i.e. the samples that are also close to the classification boundary of two classes. Conceptually these samples are similar to support vectors in SVM. We name them as classification representative (CR) samples and propose the following simple, effective scheme to find them.

We observe that the solutions of the coefficient vectors $\boldsymbol{\alpha}_k^{CH}$ and $\boldsymbol{\alpha}_k^{CC}$ in the minimum distance problems (3.6) for NCHM and (3.9) for NCCM are usually sparse, i.e. some of the entries of the coefficient vectors are zeros or very close to zeros. Remember that the coefficient vector is constructed by the linear combination coefficients of the training samples to reconstruct a new test sample. Thus the sparse $\boldsymbol{\alpha}_k^{CH}$ and $\boldsymbol{\alpha}_k^{CC}$ indicate that only a fraction of the training samples are selected to reconstruct a test sample in NCHM and NCCM.

To find the CR samples of one class that are close to the model of the other class, we use the training samples from one class to reconstruct the test sample from the other class. A test sample from class k_1 selects several nearest training samples from class k_2 to reconstruct itself, as the distance from the test sample to the reconstructed sample should be minimised. The selected training samples of class k_2 to reconstruct the test sample from class k_1 satisfy the two requirements of CR samples, i.e. close to both the geometric boundary of the class model and the classification boundary of two classes. Thus the CR samples of class k_2 could be found based on the reconstruction coefficients of the test samples from class k_1. The most frequently selected training samples of class k_2 to reconstruct the test samples from class k_1 are chosen as the CR samples of class k_2. We show an example of finding the CR samples of class 1 based on the test samples of class 2 as follows.

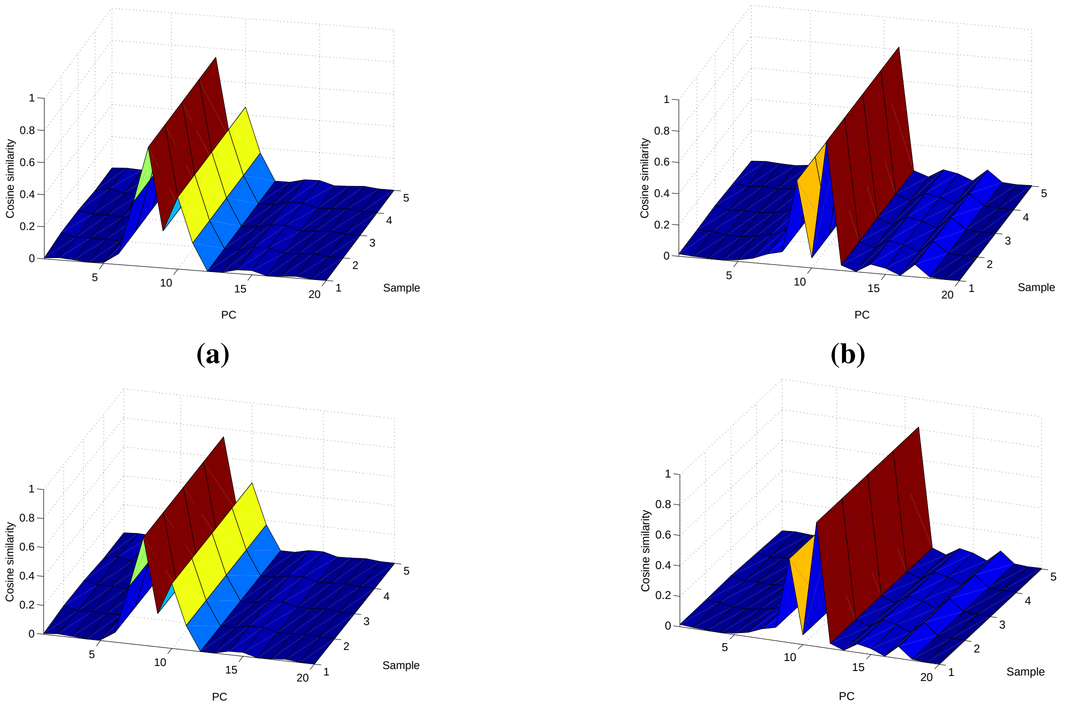
Suppose the CR samples of the convex cone model of class 1 is denoted by $\boldsymbol{X}_{CR}^1 \in \mathbb{R}^{m \times p}$, where the superscript 1 denotes class 1 and m is the number of representative samples. Denote the ith test samples from class 2 as $\boldsymbol{x}_{newi}^{(2)}$ $(i = 1, \dots, n_2)$, where n_2 is the number of test samples from class 2. We solve the following problem for all n_2 samples:

$$\min_{\boldsymbol{\alpha}_{1i}^{CC}} ||\boldsymbol{x}_{newi}^{(2)} - (\boldsymbol{X}_1^T \boldsymbol{\alpha}_{1i}^{CC})^T||_2^2, \text{ s.t. } \boldsymbol{\alpha}_{1i}^{CC} \geq 0, \; i = 1, \dots, n_2, \tag{3.14}$$

where $\boldsymbol{X}_1 \in \mathbb{R}^{n_1 \times p}$ is the training set of class 1. We denote $\boldsymbol{A}_1 = [\boldsymbol{\alpha}_{11}^{CC*}, \boldsymbol{\alpha}_{12}^{CC*}, \dots, \boldsymbol{\alpha}_{1n_2}^{CC*}] \in \mathbb{R}^{n_1 \times n_2}$, where $\boldsymbol{\alpha}_{1i}^{CC*}$ denotes the solution of (3.14). The nonzero entries in the ith column of $\boldsymbol{A}_1$ denote the coefficients of the training samples from class 1 that are selected to reconstruct the ith test sample from class 2. We count the number of the nonzero entries for each row of $\boldsymbol{A}_1$ and denote it as $\boldsymbol{t} \in \mathbb{R}^{n_1 \times 1}$ to represent the frequencies that the n_1 training samples of class 1 are chosen to reconstruct the test sample from class 2. We record the positions of the first m largest frequencies and choose the training samples in the corresponding positions in $\boldsymbol{X}_1$ as the CR samples $\boldsymbol{X}_{CR}^1$. The CR samples of the convex cone model of class 2, $\boldsymbol{X}_{CR}^2$, can be found similarly. Furthermore, the CR samples of a convex hull model can be found in a similar scheme by changing the constraints in (3.14) for a convex cone to the constraints for a convex hull.

The cosine similarities between the data variation directions (i.e. the directions of the PCs) and the directions of the CR samples are measured to estimate the distribution of the data of each class. In the following analysis, we set $m = 5$, i.e. select five CR samples for each class, and calculate their cosine similarity with the first 20 PCs.

Another important property of the distributions of two classes that relates to classification is their separation. We use the PC plot to visually check the separation of two classes. The PCs are selected based on their discriminative ability as in Figure 3.9, Figure 3.10 and Figure 3.11; the PCs with high discriminative ability are selected for the plot.



(a)

(b)

(c)

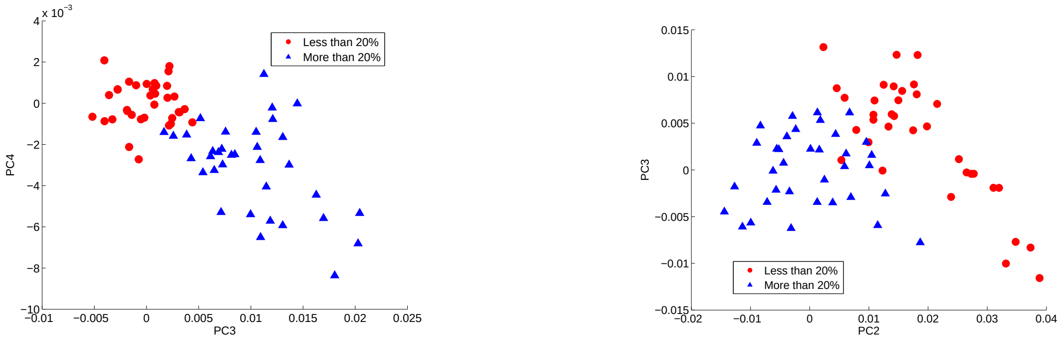
(d)

Figure 3.12: Cosine similarities between the CR samples and the PCs for the fat dataset. (a) and (b): for NCCM on the “less than 20%” class and the “more than 20%” class, respectively. (c) and (d): for NCHM on the two classes.

3.3.5.1 The fat dataset

In the fat dataset, totally 34 PCs could be generated for each class. We present the relationships between the CR samples and the first 20 PCs in Figure 3.12. We first observe that the CR samples have the same cosine similarities with each PC, which indicates that the five CR samples found in our method have the same directions. In addition, Figure 3.12a for NCCM is the same as Figure 3.12c for NCHM; similarly, Figure 3.12b for NCCM is the same as Figure 3.12d for NCHM. This indicates that NCHM and NCCM have the same CR samples, which is consistent with the results in Figure 3.9a and Figure 3.9b that the curves of normal vectors for NCHM and NCCM overlap with each other.

It is also clear that, for both NCHM and NCCM, the CR samples are orthogonal to the first five leading PCs, as indicated by the zero cosine similarities. This result indicates that the main variation direction of the data of each class is orthogonal to the geometric boundaries of that class.



(a) Subspace: “Less than 20%”

(b) Subspace: “More than 20%”

Figure 3.13: PC plots of the fat dataset.

Figure 3.13 shows the projections of the training samples to the two class subspaces. It is clear that the two classes can be well separated by using PC3 and PC4 in the “less than 20%” subspace and by using PC2 and PC3 in the “more than 20%” subspace. This result indicates that the two classes can be well separated on some directions in the original feature space. In addition, the cosine similarities between the first five pairs of PCs of the two classes are 0.789, 0.687, 0.847, 0.944 and 0.880, which suggests that the two classes have similar directions of the most variation.

Based on the above analysis, we could summarise the following properties of the distribution of the fat data. Firstly, the directions of the most variation of the two classes are similar. Secondly, for each class, the CR samples are orthogonal to the leading PCs, i.e. the directions of the most variations. Thirdly, the two classes are separable on some leading PCs. We show an illustrative example of data with such properties in a 2D feature space in Figure 3.14.

As illustrated by Figure 3.14 and empirically validated by Figure 3.8a, the fat dataset gives an example that the geometric convex models of different classes can be well separated and thus are suitable for this dataset.

The reason of NSM providing worse classification performance is that the discriminative information in the leading PCs are not used in the normal vectors, as indicated in Figure 3.9. In an extreme case, if the two classes have almost the same direction of their first PCs, PC$_1^1$ and PC$_1^2$ as shown in Figure 3.14, and the samples from the two classes are separable on this direction, then NSM will fail to classify test samples. This is because the discriminative information only exists in the first

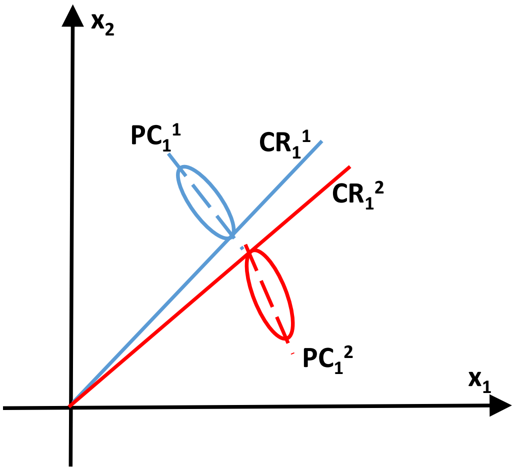


Figure 3.14: An illustration of the distribution of the fat data in a 2D space. The training samples of the two classes are illustrated by blue and red ellipses; the first pair of PCs are PC_1^1 and PC_1^2 for the two classes and the first pair of CR samples are CR_1^1 and CR_1^2.

PCs, but they are used to build the PC class models whereas the residual PCs used to calculate the distances for classification do not have sufficient discriminative information. However, in this case NCHM and NCCM may classify the test samples well, as long as the two classes are separable and the CR samples are close to being orthogonal to the first PCs. This is one of the most suitable cases for NCHM and NCCM than NSM. More general, the results suggest that if the two classes are separable, then NCHM and NCCM can be better classifiers than NSM.

3.3.5.2 The meat dataset

Similarly to the analysis of the fat dataset, we show the relationships between the CR samples and the PCs for the meat dataset in Figure 3.15. In contrast to the result in Figure 3.12 for the fat dataset, Figure 3.15 shows that the cosine similarities between the CR samples and the first three PCs are high for the meat dataset. This suggests that the direction of the CR samples and those of the most variation are very similar.

The PC plots of the meat dataset are shown in Figure 3.16, which shows that the two classes are mixed in the middle, not as separate as in Figure 3.13 for the fat dataset. In addition, the cosine similarities between the first two pairs of the PCs

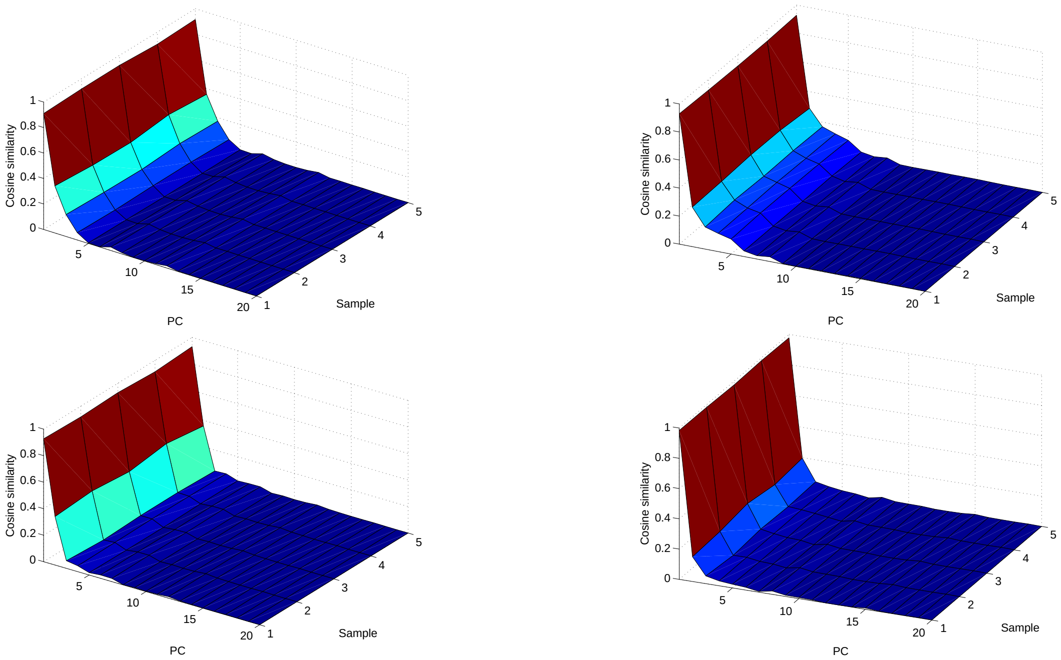
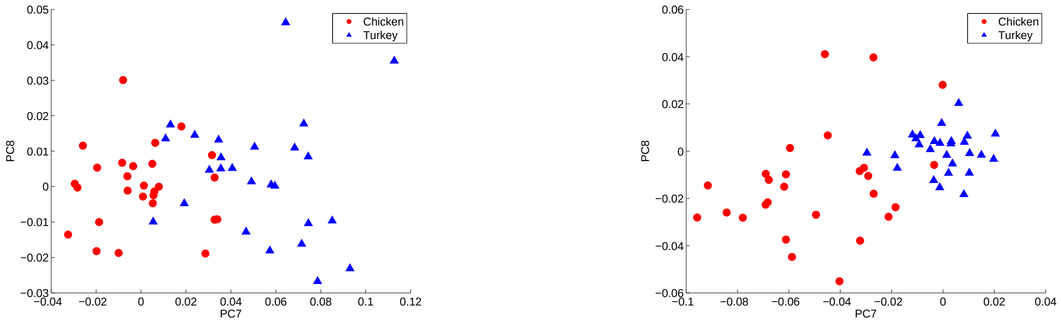


Figure 3.15: Cosine similarities between the CR samples and the PCs for the meat dataset. (a) and (b): for NCCM on the chicken class and the turkey class, respectively. (c) and (d): for NCHM on the two classes.



(a) Subspace: "Chicken"

(b) Subspace: "Turkey"

Figure 3.16: PC plots of the meat dataset.

of the two classes are 0.998 and 0.933, indicating extreme similarity between two classes, especially the first pair.

Hence the properties of the distribution of the meat data can be summarised as follows. Firstly, the directions of the most variation of the two classes are extremely similar. Secondly, for each class, the CR samples have similar directions as the leading PCs. Thirdly, the two classes are not separable. We illustrate the data distribution with such properties in a 2D feature space in Figure 3.17.

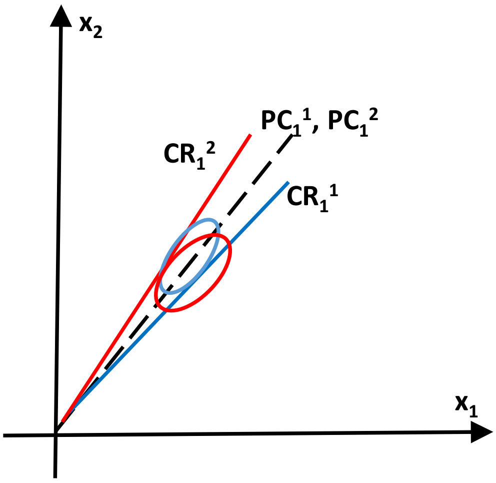


Figure 3.17: An illustration of the distribution of the meat data in a 2D space. The black dashed line indicates the same directions of the first PCs of the two classes.

Since the two classes are mixed together, as illustrated in Figure 3.17, it makes sense that NCHM and NCCM provide bad classification. This is because the convex hull models and the convex cone models are built using all the training samples in the original feature space. The mixture of the training samples results in overlapping of the geometric convex class models. Thus classifying a test sample is hard.

In contrast, NSM can capture the discriminative information from the residual PCs, i.e. the PCs that are not used to build the class models and have nonzero cosine similarities in Figure 3.10. Thus NSM can perform better in this case.

The results of the meat dataset suggests that when there is overlap between the training samples of two classes, NCHM and NCCM should be used with caution. A selection of training samples to build convex models with less overlap might be a remedy to obtain better classification.

3.3.5.3 The Phenyl dataset

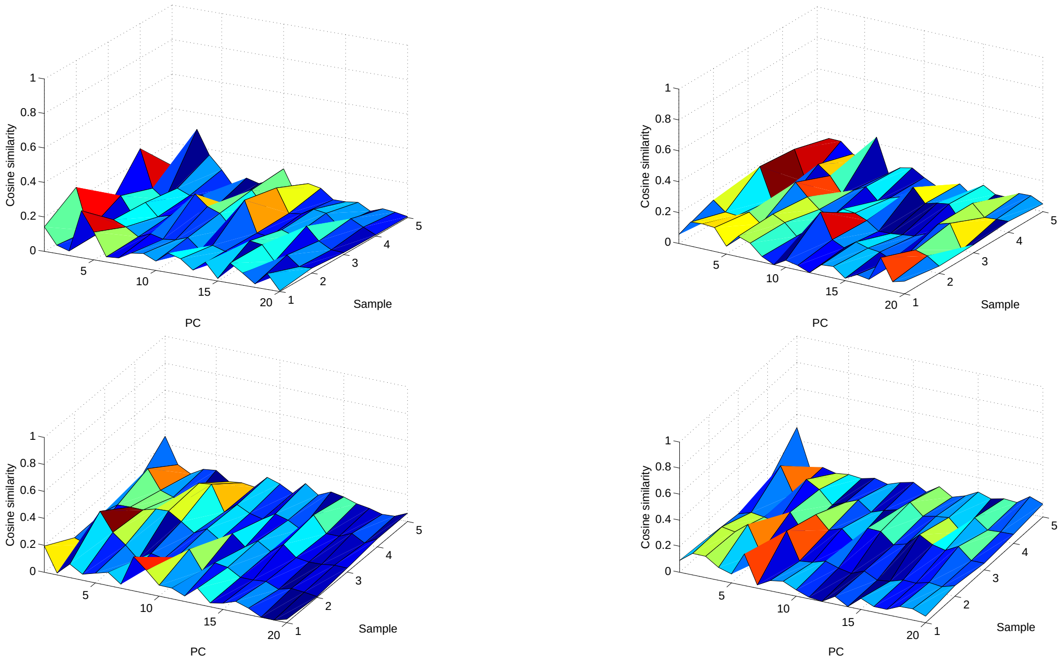
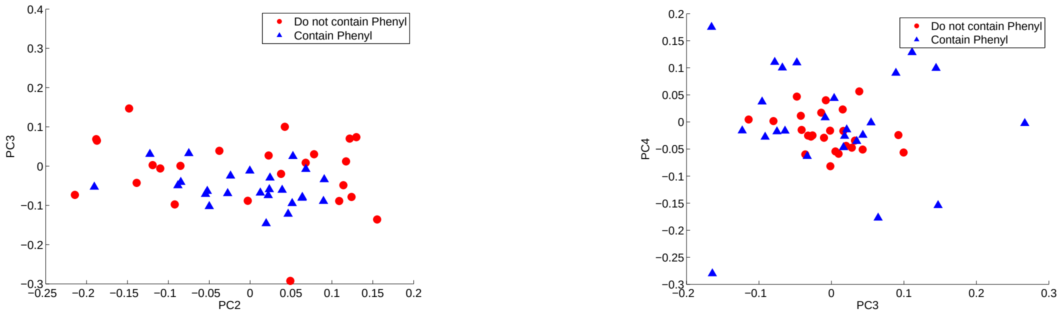


Figure 3.18: Cosine similarities between the CR samples and the PCs for the Phenyl dataset. (a) and (b): for NCCM on the "with Phenyl structure" class and the "without Phenyl structure" class, respectively. (c) and (d): for NCHM on the two classes.

Different from the previous two datasets, there is no clear trends of the similarities between the representative samples and the leading PCs for the Phenyl dataset, as shown in Figure 3.18. Moreover, all the cosine similarities have small values.



(a) Subspace: "Not contain Phenyl"

(b) Subspace: "Contain Phenyl"

Figure 3.19: PC plots of the Phenyl dataset.

The Phenyl dataset shows a heavy mixture of the two classes, as shown in Figure 3.19, which explains the bad classification accuracies for all methods in

Figure 3.8. These suggest that NSM, NCHM and NCCM perform badly for datasets with severe overlap of the two classes, with NSM to perform slightly better than NCHM and NCCM.

To sum up, the analysis in this section suggests the following findings. Firstly, when the two classes are separable, NCHM and NCCM may provide better classification performance than NSM. Especially, when the two classes have almost the same direction of the first PCs and this direction contains most the discriminative information, NSM will fail to classify the two classes; however, in this case NCHM and NCCM can provide good classification performance so long as the two classes are separable and the CR samples are close to being orthogonal to the first PCs. Secondly, when the two classes have some overlapping parts, NSM may have better classification performance than NCHM and NCCM, if it can capture the discriminative information in the dataset. Finally, when the two classes have heavy overlap, we may expect bad performances from all three methods.

3.4 Conclusion

In this chapter, we use geometric convex models as class models in SIMCA instead of the PC class subspaces for spectral data classification. We propose NCCM and provide a thorough investigation of NSM, NCHM and NCCM theoretically and empirically. We prove theoretical results for the minimum distance problem of NCCM, based on the relationship between a convex cone and its polar cone. We establish a separating hyperplane classification (SHC) framework for nearest-class-model methods with arbitrary norms. We analyse the data-dependant classification performances of NSM, NCHM and NCCM, based on the discriminative ability of normal vectors. We also provide a simple and effective method to find the class representative samples and estimate the properties of the data distributions.

Part II

Contributions to the distances used in SIMCA

Part II presents our contributions to the distances used in the classification rule in the test phase of SIMCA. In this part, we focus on studying the distances and fix the PC class subspaces as the class models. Two distances related to the PC models are of great importance in SIMCA: 1) the *squared* orthogonal distance (OD^2), i.e. the *squared* orthogonal Euclidean distance from a test instance to a PC model; and 2) the *squared* score distance (SD^2), i.e. the *squared* Mahalanobis distance from the projection of a test instance to the centre of a PC model. In recent applications, a linear combination of the two distances is used to classify a test instance: the test instance is assigned to the class with the minimum value of the linear combination.

We present our two contributions in Chapter 4 and Chapter 5, respectively. First, in Chapter 4 we investigate the difference of calculating OD^2 between using formulae in the highly-cited SIMCA paper (De Maesschalck et al., 1999) and using those in the original SIMCA paper (Wold, 1976) for low-dimensional and high-dimensional scenarios. Second, in Chapter 5 we propose a method of learning distance to subspace to learn tailored distance metrics for SIMCA.

Chapter 4

On the orthogonal distance of SIMCA for high-dimensional data

The usages of the OD^2, the SD^2 and their distributions are of great interest to pattern classification in chemometrics (Branden and Hubert, 2005; Pomerantsev, 2008; Pomerantsev and Rodionova, 2014). There is a close relationship between the OD^2 from a test instance to a class model and the residual standard deviation of the test instance to the class model. A lot of researchers calculate the OD^2 following the formulae of the residual standard deviations defined in De Maesschalck et al. (1999), instead of following the original formulae defined in Wold (1976). De Maesschalck et al. (1999) show that the residual standard deviation based on the residual matrix can be equivalently calculated by using the residual PC scores based on the PC score matrix. Their work has been cited over a hundred times, including methodological developments (Candolfi et al., 1999; De Maesschalck et al., 2000; Daszykowski et al., 2007), reviews (Uríčková and Sádecká, 2015; Kumar et al., 2014) and applications (Candolfi et al., 1999; Bicciato et al., 2003; Chen et al., 2006; Waddell et al., 2014; Da Silva et al., 2015).

In this chapter, we shall show that the relationship, between the residual standard deviation and the sum of squares of the residual PC scores, shown in De Maesschalck et al. (1999) is not always valid. We shall focus on the difference between the calculation of the OD^2s using the formulae in the original work of SIMCA (Wold, 1976) and that using the formulae in De Maesschalck et al. (1999).

The two OD^{2}s considered here are as follows.

1. The OD^2, $v^{k,l}$, from the training instance l to the model of class k that was built from all training instances. It is closely related to the residual standard deviation, $s^{k,0}$, of class k, as in De Maesschalck et al. (1999) and Wold (1976).
2. The OD^2, $v^{k,new}$, from the new test instance to the model of class k. It is closely related to the residual standard deviation, $s^{k,new}$, of the new test instance to class k, as in De Maesschalck et al. (1999) and Wold (1976).

The above two OD^{2}s are widely used in SIMCA for classification and outlier detection. The sample statistics of $v^{k,l}$ are usually used to provide scaled $v^{k,new}$. For example, in Pomerantsev (2008) and Pomerantsev and Rodionova (2014) only the mean is used, while in Branden and Hubert (2005) both the mean and the standard deviation of $v^{k,l}$ are used. The only difference between $v^{k,l}$ and $v^{k,new}$ is that $v^{k,l}$ is the OD^2 for the training instance while $v^{k,new}$ is the OD^2 for the test instance.

De Maesschalck et al. (1999) provide formulae for $s^{k,0}$ and $s^{k,new}$ using the residual PC scores, which are different in formulation from but supposed to be equivalent to those in the original SIMCA paper (Wold, 1976). We shall show that, although the formula in De Maesschalck et al. (1999) for $s^{k,0}$ is indeed equivalent to the original one in Wold (1976), the formula in De Maesschalck et al. (1999) for $s^{k,new}$ is only precise when the training data of class k have more samples (also called instances) than predictor variables (also called features), i.e. when the number of samples (denoted by n_k) is larger than the number of features (denoted by p). In other words, when the training data of class k are high-dimensional (i.e. $n_k \leq p$, also called "large p, small n" in the statistical literature), the calculation of $s^{k,new}$ in De Maesschalck et al. (1999) is not precise.

Because of the above results, we shall point out that, for high-dimensional data, although the OD^2 $v^{k,l}$ can be accurately calculated by following the (precise) formula of the residual standard deviation $s^{k,0}$ in De Maesschalck et al. (1999), the OD^2 $v^{k,new}$ cannot be accurately calculated by following the (imprecise) formulae of the residual standard deviation $s^{k,new}$ in De Maesschalck et al. (1999). Consequently, inference results of the studies that calculated the OD^{2}s for high-dimensional data using the formulae in De Maesschalck et al. (1999) can be imprecise.

Because high-dimensional data, such as spectral data, are commonly present in chemometrics and many other disciplines involving pattern-recognition tasks and because SIMCA is widely applied in those cases, it is of great interest to practitioners to point out the imprecise calculation of the OD^{2}s for high-dimensional data if we follow the formulae in De Maesschalck et al. (1999), as well as to suggest that the original formulae in Wold (1976) should be adopted in this "large p, small n" paradigm.

4.1 The calculations of SIMCA in De Maesschalck et al. (1999)

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

4.1.1 The training phase of class k

Suppose $\boldsymbol{X} \in \mathbb{R}^{n_k \times p}$ is the training set of class k, in which there are n_k training instances (or say training samples) and each instance is represented by a p-dimensional data vector. To build the PC model of class k, we apply the reduced singular value decomposition (SVD) to the column-centred training set $\boldsymbol{X}_{(c)}$:

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_q \boldsymbol{D}_q (\boldsymbol{V}_q)^T \ , \tag{4.1}$$

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

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

$$\boldsymbol{X}_{(c)} = \boldsymbol{T}_r(\boldsymbol{V}_r)^T + \boldsymbol{E} \ , \tag{4.2}$$

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

In De Maesschalck et al. (1999), the residual standard deviation of class k are expressed in two forms:

$$s^{k,0} = \sqrt{\sum_{l=1}^{n_k}\sum_{j=1}^{p}(e_{lj})^2/[(q-r)(n_k-r-1)]} \tag{4.3}$$

$$= \sqrt{\sum_{l=1}^{n_k}\sum_{i=r+1}^{q}(t_{li})^2/[(q-r)(n_k-r-1)]} \ , \tag{4.4}$$

where e_{lj} is the (l,j)-entry of residual matrix $\boldsymbol{E}$ representing the residual of the lth instance for the jth variable, and t_{li} is the (l,i)-entry of score matrix $\boldsymbol{T}_q$ representing the score of the lth instance for the ith PC. The squared residual standard deviation of class k, $(s^{k,0})^2$, can be considered as the sum of the OD^{2}s from the training instances to the model of class k divided by the degrees of freedom $(q-r)(n_k-r-1)$.

4.1.2 The test phase for class k

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

$$\boldsymbol{x}_{(c)}^{k,new} = \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T + \boldsymbol{e}^{k,new} \ , \tag{4.5}$$

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

In De Maesschalck et al. (1999), the residual standard deviation of the new instance are also expressed in two forms:

$$s^{k,new} = \sqrt{\sum_{j=1}^{p} (e_j^{k,new})^2/(q-r)} \tag{4.6}$$

$$= \sqrt{\sum_{i=r+1}^{q} (t_i^{k,new})^2/(q-r)} \,, \tag{4.7}$$

where $e_j^{k,new}$ and $t_i^{k,new}$ denote the jth element of the residual vector $\boldsymbol{e}^{k,new}$ and the ith element of the PC score vector $\boldsymbol{t}_r^{k,new}$, respectively. The squared residual standard deviation of the new instance, $(s^{k,new})^2$, can be considered as the OD^2 from the new instance to the class k model divided by the degrees of freedom $(q-r)$.

To determine the class of $\boldsymbol{x}_{new}$, the residual standard deviation $s^{k,new}$ of $\boldsymbol{x}_{new}$ is compared to the residual standard deviation $s^{k,0}$ of the training instances of class k. The F-test statistic used in De Maesschalck et al. (1999) to determine whether the two residual variances are significantly different is expressed as

$$F^{k,new} = \frac{(s^{k,new})^2}{(s^{k,0})^2} = \frac{\sum_{i=r+1}^{q}(t_i^{k,new})^2 \; (n_k - r - 1)}{\sum_{l=1}^{n_k}\sum_{i=r+1}^{q}(t_{li})^2} \,. \tag{4.8}$$

4.2 The calculation of $v^{k,l}$ and $v^{k,new}$ in De Maesschalck et al. (1999)

The OD^2 is originally defined as the sum of squares of the residuals from a sample to the class model, which is closely related to the residual standard deviation. The two OD^{2}s discussed in this chapter are calculated in De Maesschalck et al. (1999) as follows.

First, $v^{k,l}$ is originally defined as $\sum_{j=1}^{p}(e_{lj})^2$, which is closely related to $s^{k,0}$, i.e. $\sum_{l=1}^{n_k} v^{k,l} = (s^{k,0})^2(q-r)(n_k-r-1)$. In De Maesschalck et al. (1999), it follows from (4.4) that $v^{k,l}$ can be calculated as

$$v^{k,l} = \sum_{i=r+1}^{q} (t_{li})^2 \,. \tag{4.9}$$

Second, $v^{k,new}$ is originally defined as $\sum_{j=1}^{p}(e_j^{k,new})^2$, which is closely related to $s^{k,new}$, i.e. $v^{k,new} = (s^{k,new})^2(q-r)$. In De Maesschalck et al. (1999), it follows from (4.7) that $v^{k,new}$ can be written as

$$v^{k,new} = \sum_{i=r+1}^{q} (t_i^{k,new})^2 \ . \tag{4.10}$$

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

The calculations for $v^{k,0}$ and $v^{k,new}$ in De Maesschalck et al. (1999) use formulae (4.9) and (4.10), respectively. We shall show that, while formula (4.9) is correct for both the cases of $n_k > p$ and $n_k \leq p$, formula (4.10) is only valid when $n_k > p$.

4.3.1 $v^{k,l}$

The OD^2 $v^{k,l}$ is originally defined on the basis of the residual matrix $\boldsymbol{E}$. The calculation of $v^{k,l}$ in (4.9), which was defined in De Maesschalck et al. (1999), is on the basis of the PC score matrix $\boldsymbol{T}_r$. This is due to the relationship that

$$\sum_{j=1}^{p} (e_{lj})^2 = \sum_{i=r+1}^{q} (t_{li})^2 \ . \tag{4.11}$$

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

4.3.1.1 $n_k > p$

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

Let $\boldsymbol{x}_{(c)}^{l} \in \mathbb{R}^{1\times p}$ denote the l-th training instance in class k, i.e. the l-th row of

$\boldsymbol{X}_{(c)}$. For every $\boldsymbol{x}^l_{(c)}$ ($l = 1,\ldots,n_k$), we have $\boldsymbol{x}^l_{(c)} = \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$ and

$$\begin{aligned}\sum_{j=1}^{p}(e_{lj})^2 &= ||\boldsymbol{x}^l_{(c)} - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_r(\boldsymbol{V}_r)^T||_2^2 \\ &= ||\boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_r(\boldsymbol{V}_r)^T||_2^2 \\ &= ||\boldsymbol{t}^l_q(\boldsymbol{V}_q)^T - \boldsymbol{t}^l_r(\boldsymbol{V}_r)^T||_2^2 \\ &= \sum_{i=r+1}^{q}(t_{li})^2 ,\end{aligned} \tag{4.12}$$

where $||\cdot||_2$ denotes the Euclidean norm of a vector, and $\boldsymbol{t}^l_q$ and $\boldsymbol{t}^l_r$ are the lth row of $\boldsymbol{T}_q$ and $\boldsymbol{T}_r$, respectively. Therefore (4.11) and thus (4.9) are correct when $n_k > p$.

4.3.1.2 $n_k \leq p$

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

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

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_{n_k}\hat{\boldsymbol{D}}_p(\boldsymbol{V}_p)^T , \tag{4.13}$$

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

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

$$\boldsymbol{X}_{(c)} = \boldsymbol{U}_{n_k}\hat{\boldsymbol{D}}_p(\boldsymbol{V}_p)^T \ = \boldsymbol{U}_p\boldsymbol{D}_p(\boldsymbol{V}_p)^T, \tag{4.14}$$

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

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

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

$$\begin{aligned}
||\boldsymbol{m}^l||_2^2 &= ||\boldsymbol{x}^l_{(c)} - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\
&= ||\boldsymbol{x}^l_{(c)}\boldsymbol{V}_p(\boldsymbol{V}_p)^T - \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\
&= ||\boldsymbol{t}^l_p(\boldsymbol{V}_p)^T - \boldsymbol{t}^l_q(\boldsymbol{V}_q)^T||_2^2 \,. \qquad (4.15)
\end{aligned}$$

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

$$\sum_{l=1}^{n_k} ||\boldsymbol{m}^l||_2^2 = \sum_{l=1}^{n_k}\sum_{i=q+1}^{p} (t_{li})^2 = \sum_{i=q+1}^{p} (\lambda_i)^2 \,. \qquad (4.16)$$

The second equation in (4.16) can be shown as follows. $\boldsymbol{X}_{(c)} = \boldsymbol{U}_p\boldsymbol{D}_p(\boldsymbol{V}_p)^T \Rightarrow (\boldsymbol{U}_p)^T\boldsymbol{X}_{(c)}\boldsymbol{V}_p = \boldsymbol{D}_p \Rightarrow (\boldsymbol{U}_p)^T\boldsymbol{T}_p = \boldsymbol{D}_p$. For the ith singular value λ_i in $\boldsymbol{D}_p$, we have $(\lambda_i)^2 = (\boldsymbol{u}_i^T\boldsymbol{t}_i)^2 = \boldsymbol{t}_i^T\boldsymbol{u}_i\boldsymbol{u}_i^T\boldsymbol{t}_i = \boldsymbol{t}_i^T\boldsymbol{t}_i = \sum_{l=1}^{n_k}(t_{li})^2$, where $\boldsymbol{u}_i$ and $\boldsymbol{t}_i$ are the ith columns of $\boldsymbol{U}_p$ and $\boldsymbol{T}_p$, respectively.

Since the last $(p-q)$ singular values are zeros, $\sum_{l=1}^{n_k}||\boldsymbol{m}^l||_2^2 = 0$. Because each term in the sum $\sum_{l=1}^{n_k}||\boldsymbol{m}^l||_2^2$ is nonnegative, $||\boldsymbol{m}^l||_2^2 = 0$ for all l $(l = 1,\ldots,n_k)$. Thus we have $\boldsymbol{x}^l_{(c)} = \boldsymbol{x}^l_{(c)}\boldsymbol{V}_q(\boldsymbol{V}_q)^T$, which means that the first q PCs can perfectly reconstruct the training instances in class k. Using the same proof as in (4.12), we can show that (4.11) and thus (4.9) are also true for $n_k \leq p$.

Therefore, $v^{k,l}$ can be correctly calculated by using (4.9) for both the cases of $n_k > p$ and $n_k \leq p$.

4.3.2 $v^{k,new}$

Following the original SIMCA paper (Wold, 1976), $v^{k,new}$ is defined in terms of the residual vector $\boldsymbol{e}^{k,new}$, while following De Maesschalck et al. (1999), $v^{k,new}$ is formulated in (4.10) by using the PC score $\boldsymbol{t}_r^{k,new}$ of the new sample. We shall show that the formula (4.10) is valid when $n_k > p$ but not valid when $n_k \leq p$, in the following two subsections, respectively.

4.3.2.1 $n_k > p$

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

$$\sum_{j=1}^{p} (e_j^{k,new})^2 = \sum_{i=r+1}^{q} (t_i^{k,new})^2 \ . \tag{4.17}$$

Using a proof similar to (4.12) by replacing $\boldsymbol{x}_{(c)}^l$ with $\boldsymbol{x}_{(c)}^{k,new}$, we can readily show that (4.17) and thus (4.10) are correct for $n_k > p$.

4.3.2.2 $n_k \leq p$

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

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

$$\begin{aligned} ||\boldsymbol{m}^{k,new}||_2^2 &= ||\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new} \boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{x}_{(c)}^{k,new} \boldsymbol{V}_p(\boldsymbol{V}_p)^T - \boldsymbol{x}_{(c)}^{k,new} \boldsymbol{V}_q(\boldsymbol{V}_q)^T||_2^2 \\ &= ||\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T||_2^2 \\ &= \sum_{i=q+1}^{p} (t_i^{k,new})^2 \ , \end{aligned} \tag{4.18}$$

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

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

Hence, if we rewrite

$$\begin{aligned}\boldsymbol{x}_{(c)}^{k,new} &= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T + \boldsymbol{m}^{k,new}\\ &= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T + (\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T) + \boldsymbol{m}^{k,new}\ ,\end{aligned} \tag{4.19}$$

we have

$$\begin{aligned}\boldsymbol{e}^{k,new} &= (\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_q(\boldsymbol{V}_q)^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{V}_r(\boldsymbol{V}_r)^T) + \boldsymbol{m}^{k,new}\\ &= (\boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T) + (\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_q^{k,new}(\boldsymbol{V}_q)^T)\\ &= \boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T\end{aligned} \tag{4.20}$$

and

$$\begin{aligned}\sum_{j=1}^{p}(e_j^{k,new})^2 &= ||\boldsymbol{e}^{k,new}||_2^2\\ &= ||\boldsymbol{t}_p^{k,new}(\boldsymbol{V}_p)^T - \boldsymbol{t}_r^{k,new}(\boldsymbol{V}_r)^T||_2^2\\ &= \sum_{i=r+1}^{p}(t_i^{k,new})^2\\ &= \sum_{i=r+1}^{q}(t_i^{k,new})^2 + \sum_{i=q+1}^{p}(t_i^{k,new})^2\ .\end{aligned} \tag{4.21}$$

Comparing (4.21) with (4.17), we can find an additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ in (4.21), and this term may not be zero. It follows that (4.17) and thus (4.10) are not valid when $n_k \leq p$.

When $n_k \leq p$, $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is hard to estimate because the last $(p-q)$ PCs are randomly calculated by satisfying the orthogonal condition. Nevertheless, it can be harmful to the classification of the new instance of high-dimensional “large p, small n” data, if we use (4.10) to calculate $v^{k,new}$ which omits $\sum_{i=q+1}^{p}(t_i^{k,new})^2$, because the decision making for classification is based on $v^{k,new}$.

4.4 Experiments

In the following experiments, we compare the SIMCA with the OD^2 as originally defined in Wold (1976) (denoted by SIMCA) and the SIMCA with the OD^2 calculated by following De Maesschalck et al. (1999) (denoted by SIMCA-D), using both the simulated datasets and the real datasets. We aim to show that the additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ can be important for classifying high-dimensional data. To simplify the experiment settings, we discuss the effect of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ on two-class classification in the experiments. The effect of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ on multi-class classification can be readily extended.

4.4.1 Classification rule

New test instances are classified by following the classification rule of the robust SIMCA (RSIMCA) (Branden and Hubert, 2005), which is a linear combination of the OD^2 and the SD^2 of a new test instance. That is, a new test instance is classified to the class with the minimum value of

$$\gamma\frac{\mathrm{OD}_k^2}{c_{\mathrm{OD}^2}^k}+(1-\gamma)\frac{\mathrm{SD}_k^2}{c_{\mathrm{SD}^2}^k}\ , \tag{4.22}$$

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

Since OD_k^2 is the only term that is different between SIMCA and SIMCA-D, the value of the second term in (4.22) does not affect the difference between SIMCA and SIMCA-D. We force the value of the second term in (4.22) to zero by setting $\gamma = 1$, to simplify the experiments.

4.4.2 Validation criterion

We use the overall misclassification percentage (MP) as the validation criterion following the experiments in Branden and Hubert (2005). We use the one-assignment-rule suggested in Branden and Hubert (2005), i.e. a test sample is assigned to one of the known classes with the smallest F-value, to simplify the calculation of the MP and obtain unambiguous final results. The MP is defined as

$$\mathrm{MP} = \sum_{k=1}^{K} n_k^t / N^t \ , \tag{4.23}$$

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

4.4.3 Datasets

4.4.3.1 Simulated datasets

Simulated datasets are generated by following the experiments in Pomerantsev and Rodionova (2014). Assume that a sample vector $\boldsymbol{x}$ is the sum of two independent normal random components:

$$\boldsymbol{x} = \boldsymbol{\delta} + \boldsymbol{\varepsilon} \ , \tag{4.24}$$

where

$$\boldsymbol{\delta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \text{ and } \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}) \ . \tag{4.25}$$

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

Table 4.1: Simulation settings. Notation: K, number of classes; D, number of datasets; n_k, number of samples in each class.

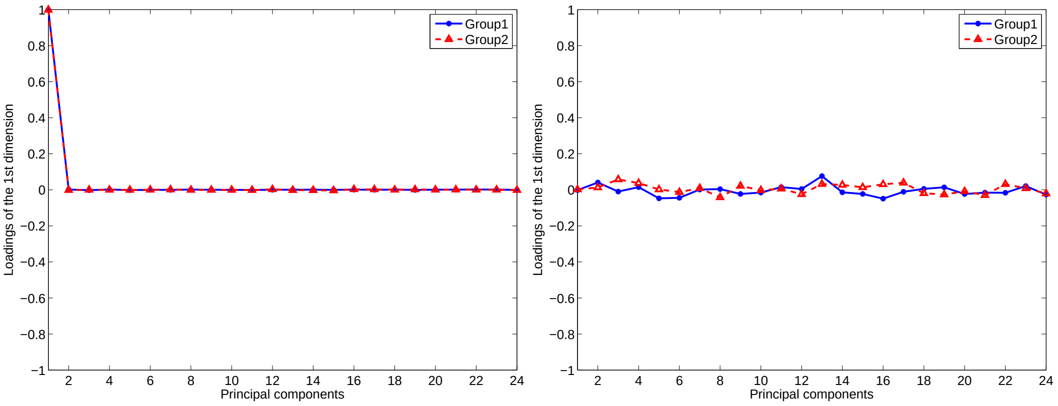
	Simulation A	Simulation B
$\boldsymbol{\mu}_1$	$\boldsymbol{0}_p$	$\boldsymbol{0}_p$
$\boldsymbol{\mu}_2$	$(10, \boldsymbol{0}_{p-1}^T)^T$	$(10, \boldsymbol{0}_{p-1}^T)^T$
$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$	$\begin{bmatrix} 5000 & 0.1 & 0.1 & \cdots & 0.1 \\ 0.1 & 0.1 & 0.1 & \cdots & 0.1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

Two sets of parameters, simulation A and simulation B, are devised to show the following two situations, respectively: 1) $\sum_{i=q+1}^{p} (t_i^{k,new})^2$ is not important for classification; and 2) $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ may be important for classification. The details of the two simulation settings are summarised in Table 4.1.

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

It is clear in Table 4.1 that the only difference between simulation A and simulation B is the values of $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$, which determines the importance of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ for classification. In both simulations, the first dimensions of the feature vectors contain major discriminative information since $\mu_{11} = 0$ and $\mu_{21} = 10$, while other dimensions contain little discriminative information since $\mu_{1i} = \mu_{2i} = 0$ $(i \neq 1)$. Therefore, the variance of the first dimension determines how the discriminative information between two classes is distributed to the PCs. The discriminative information left in the residuals for classification is determined by the discriminative information in the first few PCs used in the class model.

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



(a) Simulation A.

(b) Simulation B.

Figure 4.1: The loading plots of the first dimension.

Here we show an example to demonstrate the above argument. Two datasets with $p = 1250$ are generated. Applying PCA separately to the two classes of each dataset, we obtain the PCs for each class. We record the first entries of all the PCs in each class, i.e. $\boldsymbol{V}_q(1,:)$, and plot them against the PCs sorted in decreasing order of singular values, as shown in Figure 4.1 for simulation A and simulation B, respectively. These loadings indicate the contributions of the first dimensions of the feature vectors to the PCs.

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

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

4.4.3.2 Real datasets

Two real datasets are used in the experiments: the low-dimensional iris dataset and the high-dimensional Phenyl dataset. The iris dataset (Fisher, 1936) contains

150 samples with three classes: each class contains 50 samples. Each sample is described by four features. The Phenyl dataset is described in Section 2.2.1 of Chapter 2.

4.4.4 Experiment settings

In each dataset, we randomly select 25 samples from each class to generate the training set. The remaining samples generate the test set. We repeat this procedure 100 times and perform the two methods, SIMCA and SIMCA-D, on each training-test split.

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

4.4.5 Results

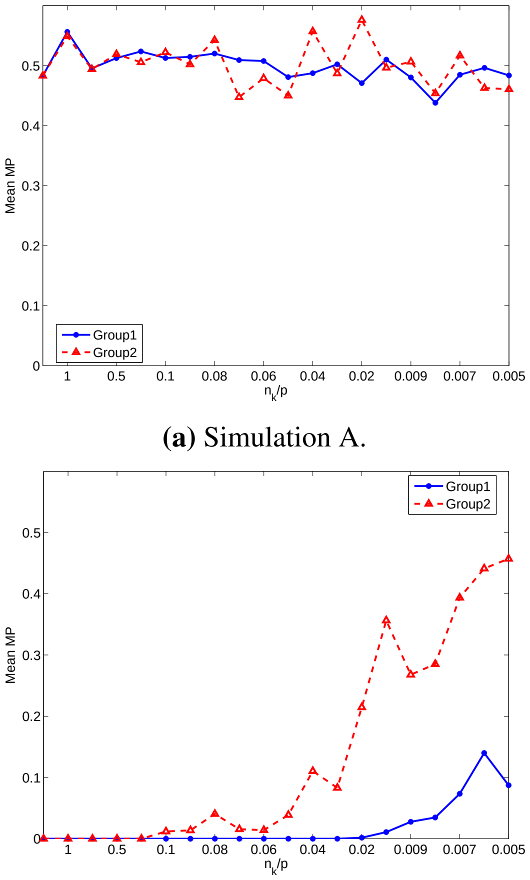
4.4.5.1 Simulated datasets

To explore the effect of the n_k/p ratio on the performances of SIMCA and SIMCA-D, we plot the mean MP against the n_k/p ratio in Figure 4.2 for simulation A and simulation B, respectively. It is clear that the mean MPs of SIMCA and SIMCA-D are the same when $n_k/p = 1.5$, i.e. in the low-dimensional situation, in each of the simulation settings, as indicated by the leftmost points in each panel of Figure 4.2.

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

In simulation A, the mean MPs of the two methods are similar for all n_k/p ratios, as shown in Figure 4.2a. This indicates that ignoring $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ in the calculation of the OD^2 does not affect the classification results in this simulation, because in this case $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is not important for classification. In addition, since the residuals are not discriminative, the mean MP varies around 0.5.

In simulation B, the difference between the mean MPs of the two methods becomes larger as n_k/p becomes smaller (i.e. when the data are higher dimensional), as shown in Figure 4.2b. Since in this simulation the first few PCs used in class models contain little discriminative information, the residual $\sum_{j=1}^{p}(e_j^{k,new})^2$ is im-



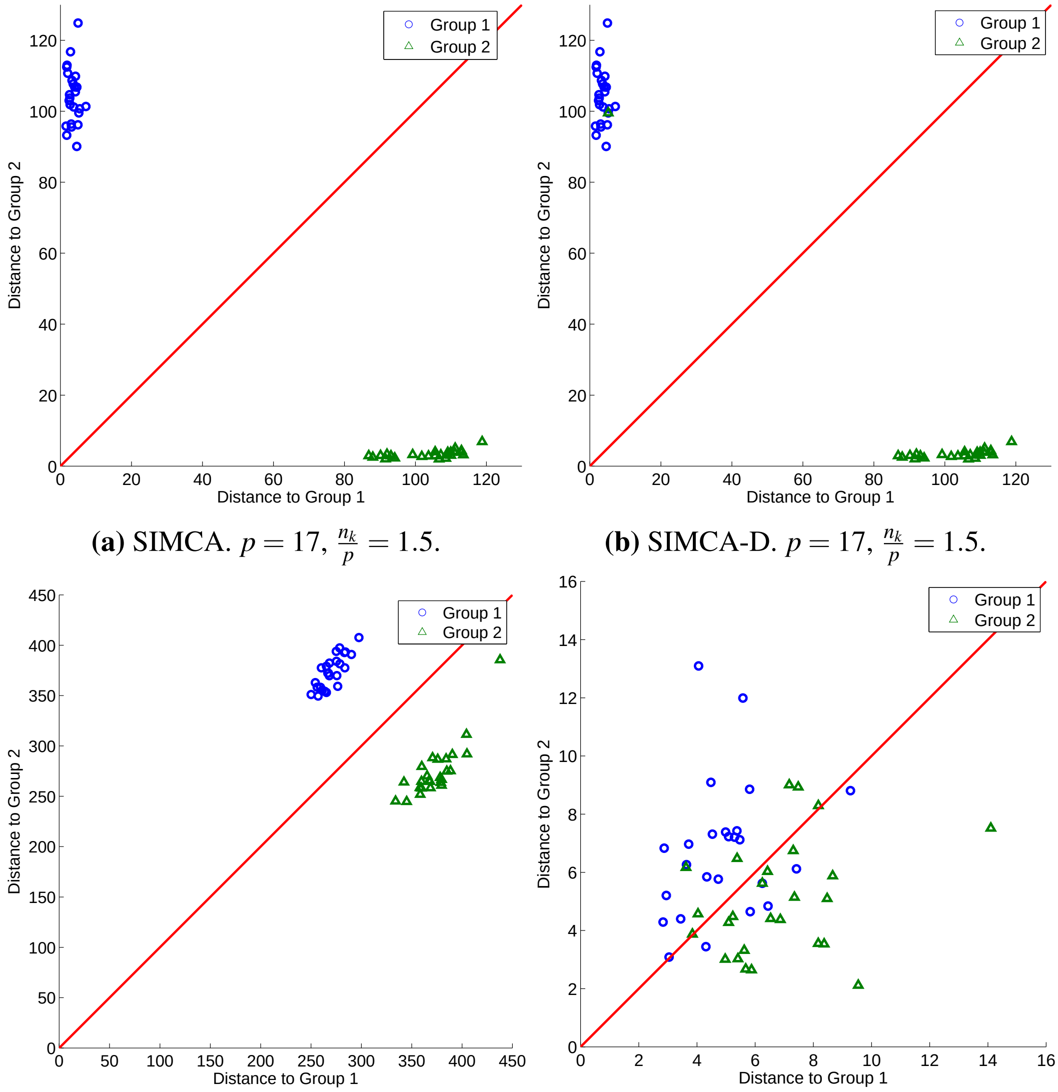
(a) Simulation A.

(b) Simulation B.

Figure 4.2: The plots of mean MP against n_k/p.

portant for classification. SIMCA performs pretty well for almost all the n_k/p ratios because $\sum_{j=1}^{p}(e_j^{k,new})^2$ captures the discriminative information for classification. In contrast, SIMCA-D, which only uses $\sum_{i=r+1}^{q}(t_i^{k,new})^2$ for classification and ignores $\sum_{i=q+1}^{p}(t_i^{k,new})^2$, cannot capture the discriminative information in $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ and can be suboptimal in classification, especially when n_k/p is small (i.e. when the data dimension is high). For example, the mean MP of SIMCA-D worsens to around 0.4 when n_k/p decreases to 0.008.

In addition for simulation B, we show an example of how $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ affects the classification performance using the Coomans' plots. Figure 4.3 shows the Coomans' plots of the test samples on one training-test split of each simulated dataset. The Coomans' plot (Vandeginste and Massart, 1998) shows the orthogonal distance from the test samples to two class models at the same time. In our exper-



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

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

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

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

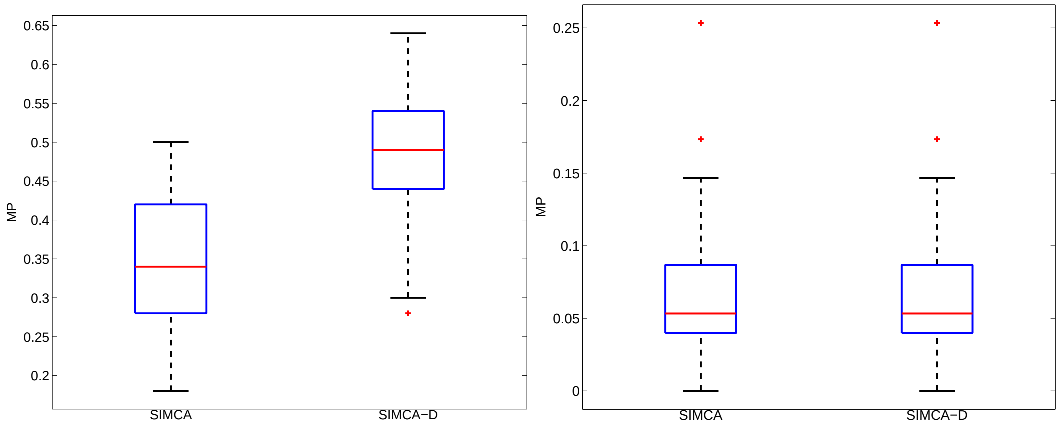
Figure 4.3: Coomans' plots.

iments, the horizontal and vertical axes denote the OD^{2}s to Group 1 and Group 2, respectively. In Figure 4.3, the red reference line divides the Coomans' plot into two parts: in the upper triangular part, the distance to Group 1 is smaller than that to Group 2; in the lower triangular part, it is the other way around.

Since SIMCA and SIMCA-D have the same q and r, the Coomans' plots reflect the difference between the OD^{2}s of these two methods.

When $n_k/p = 1.5$ (i.e. low-dimensional), the Coomans' plots of the two methods are the same. When $n_k/p = 0.02$ (i.e. high-dimensional), the Coomans' plots of the two methods are different. We observe large differences between the values of OD^{2}s in Figure 4.3c and Figure 4.3d, which indicates that the value of $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is large. Including $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ can perfectly separate the two groups as shown in Figure 4.3c; however, omitting $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ results in a mixture of the two groups as shown in Figure 4.3d. This indicates that the additional term $\sum_{i=q+1}^{p}(t_i^{k,new})^2$ is important for classification in this high-dimensional simulated dataset.

4.4.5.2 Real datasets



(a) The Phenyl data.

(b) The iris data.

Figure 4.4: The box plots of the MP for the real datasets.

Figure 4.4 shows the box plots of the MP for the real datasets. In the high-dimensional Phenyl dataset, SIMCA-D provides worse classification performance than the original SIMCA. In the low-dimensional iris dataset, the two methods provide the same results. This pattern for the real datasets is consistent with that for the simulated datasets.

4.5 Conclusion

We have investigated the formulae in De Maesschalck et al. (1999) of calculating two OD^{2}s, $v^{k,l}$ and $v^{k,new}$. We have shown that the formula for $v^{k,new}$ in De Maesschalck et al. (1999) is not valid for high-dimensional data (i.e. when $n_k \leq p$). The experiments on both the simulated datasets and the real datasets have confirmed that the formula following De Maesschalck et al. (1999) can result in worse classification performance than the original one in Wold (1976). Therefore, we suggest that the original formulae in Wold (1976) for calculating the OD^{2}s, rather than the formulae in De Maesschalck et al. (1999), should be used for high-dimensional data with more features than samples (i.e. when $n_k \leq p$).

Chapter 5

Learning distance to subspace

Given the PC class subspaces, the classification rule used in SIMCA is usually dependent on two distances from the test sample to the class subspaces: OD^2 and SD^2. In Wold's version of SIMCA (Wold, 1976), only OD^2 is used in the classification rule. Recently, a linear combination of OD^2 and SD^2 is widely adopted as the classification rule in SIMCA, such as the robust SIMCA (RSIMCA) (Branden and Hubert, 2005) and the SIMCA function from the PLS Toolbox in MATLAB. The weights for OD^2 and SD^2 in the linear combination can be determined by users with prior knowledge, or can be tuned by cross-validation using the training data.

The distances between the test samples and the class subspaces are of great importance for classification, since they determine the classification results. OD^2 uses the Euclidean distance while SD^2 uses the Mahalanobis distance. Instead of predefining the distance metrics to be used in the classification rule, distance metric learning methods emerging in the machine learning community enable us to learn tailored distance metrics automatically from data and to improve the classification performance (Xing et al., 2003; Alipanahi et al., 2008; Weinberger and Saul, 2009).

Distance metric learning methods identify distance metrics based on a set of similarity/dissimilarity constraints between training samples: the samples from the same classes are similar while the samples from different classes are dissimilar. Thus, in terms of the distance metric learned, the samples from the same class become closer to each other while those from different classes become farther apart.

It is important to notice that the distance metric learning methods in literature aim to improve the classification performance of the classification methods that are based on distance between samples, such as k-nearest neighbours (kNN). Thus the distance metrics are for distances between samples. However, the distance metrics used in subspace-based classification methods, such as SIMCA, measure the distances between samples and class subspaces. This unfortunately makes the established distance metric learning methods difficult to be applied directly to subspace-based classification methods.

In this chapter, we propose a distance metric learning method tailored for the classification rule of SIMCA to improve its classification performance. We first analyse the classification rules of SIMCA used in literature to derive a general formulation for them. We show that the general formulation is based on two parameterisation matrices with different sizes: the larger one is for the distance measurement in the original feature space and the smaller one is for the distance measurement in the PC class subspace. Hence, different classification rules of SIMCA in the literature can be shown actually using different distance metrics in the general formulation.

We define this general formulation as the distance metric from a test 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 methods, 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 to the wrong class subspaces. Using the learned distance as the similarity measurement, 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)". To evaluate the effectiveness of LD2S, we compare the classification performances of SIMCA with Wold's classification rule (SIMCA-W), SIMCA with the classification rule of RSIMCA (SIMCA-R) and SIMCA with the classification rule learned from LD2S (SIMCA-LD2S) using a real-world dataset.

5.1 Methodology

5.1.1 SIMCA

5.1.1.1 Principal component (PC) class subspace

The calculations of the PC class subspaces are the same as those in Section 4.1.1 of Chapter 4. To make the calculations afterwards clear, we show the calculations of the PC class subspaces here again with slightly different notation from those in Section 4.1.1 of Chapter 4.

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 through using the reduced singular value decomposition (SVD).

$$\boldsymbol{X}_{k(c)} = \boldsymbol{U}_{q_k} \boldsymbol{D}_{q_k} \boldsymbol{V}_{q_k}^T, \tag{5.1}$$

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

$$\boldsymbol{T}_{q_k} = \boldsymbol{U}_{q_k} \boldsymbol{D}_{q_k} = \boldsymbol{X}_{k(c)} \boldsymbol{V}_{q_k} \in \mathbb{R}^{n_k \times q_k}. \tag{5.2}$$

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

$$\boldsymbol{X}_{k(c)} = \boldsymbol{U}_{r_k} \boldsymbol{D}_{r_k} \boldsymbol{V}_{r_k}^T + \boldsymbol{E}_k, \tag{5.3}$$

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 built 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 $\mathscr{L}_k$.

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

$$\boldsymbol{x}_{(c)}^{k,new} = \boldsymbol{t}^{k,new} \boldsymbol{V}_{r_k}^T + \boldsymbol{e}^{k,new}, \tag{5.4}$$

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.

5.1.1.2 The squared orthogonal distance and the squared score distance

Given the PC class subspaces, the new sample $\boldsymbol{x}_{new}$ is classified to one of the classes 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 to class k, OD$_k^2$, from $\boldsymbol{x}_{new}^c$ to the subspace of the kth class is defined based on the residual $\boldsymbol{e}^{k,new}$ in (5.4):

$$\mathrm{OD}_k^2 = \sum_{j=1}^{p} (e_j^{k,new})^2 = \boldsymbol{e}^{k,new}(\boldsymbol{e}^{k,new})^T, \tag{5.5}$$

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

Rewriting (5.4), we have

$$\boldsymbol{e}^{k,new} = \boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k = \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{I}_p - \boldsymbol{P}_k), \tag{5.6}$$

where $\boldsymbol{I}_p$ denotes the p-by-p identity matrix. $\boldsymbol{e}^{k,new}$ can then be considered as the difference vector between $\boldsymbol{x}_{(c)}^{k,new}$ and its projection on $\mathscr{L}_k$, $\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k$. The orthogonal complement of $\mathscr{L}_k$ is $\mathscr{L}_k^{\perp}$ 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 $\mathscr{L}_k^{\perp}$. Since $\boldsymbol{e}^{k,new}$ is orthogonal to $\mathscr{L}_k$, the distance based on $\boldsymbol{e}^{k,new}$ is called the orthogonal distance.

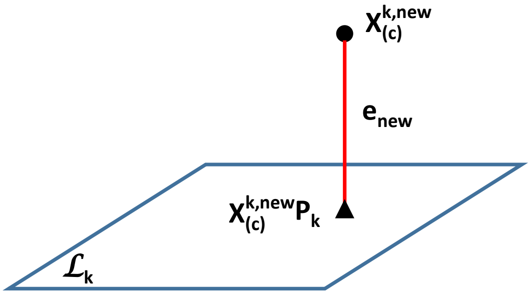


Figure 5.1: An illustration of OD_k^2 in a 3-dimensional feature space.

An illustration of OD_k^2 in a 3-dimensional feature space is shown in Figure 5.1. The new instance $\boldsymbol{x}_{(c)}^{k,new}$ is shown as the black dot; the class subspace $\mathscr{L}_k$ is shown as the dark blue 2-dimensional plane; and the projection of $\boldsymbol{x}_{(c)}^{k,new}$ to $\mathscr{L}_k$, $\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k$, 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 $\mathscr{L}_k$. 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 $\mathscr{L}_k$:

$$\mathrm{SD}_k^2 = \sum_{i=1}^{r}(t_i^{k,new}/d_i)^2 = \boldsymbol{t}^{k,new}\boldsymbol{D}_{r_k}^{-2}(\boldsymbol{t}^{k,new})^T, \tag{5.7}$$

where $\boldsymbol{D}_{r_k}$ is the diagonal matrix of singular values in (5.3). 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 $1/d_1 \leq 1/d_2 \leq \ldots \leq 1/d_{r_k}$.

Note that

$$\boldsymbol{u}^{k,new} = \boldsymbol{t}^{k,new}\boldsymbol{D}_{r_k}^{-1} \tag{5.8}$$

is the standardised score of $\boldsymbol{x}_{(c)}^{k,new}$ on $\mathscr{L}_k$. Then (5.7) can be written as

$$\mathrm{SD}_k^2 = \boldsymbol{u}^{k,new}(\boldsymbol{u}^{k,new})^T, \tag{5.9}$$

which indicates that SD_k^2 is the squared Frobenius norm of $\boldsymbol{u}^{k,new}$. An illustration

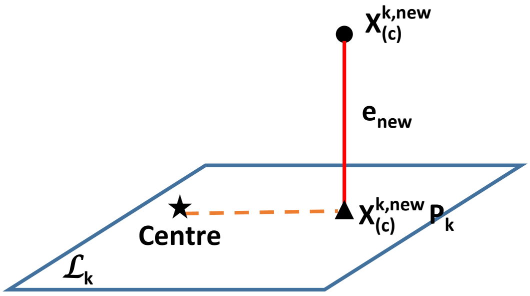


Figure 5.2: An illustration of SD_k^2 in a 3-dimensional feature space.

of SD_k^2 in a 3-dimensional feature space is shown in Figure 5.2. In addition to the symbols in Figure 5.1, the centre of the class subspace, $\mathscr{L}_k$, 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. SD_k^2 is then the reweighted length of the orange dashed line.

5.1.1.3 The classification rules

When Wold (1976) first proposed SIMCA, the classification rule was defined only based on OD_k^2. $\boldsymbol{x}_{new}$ is assigned using the F-value:

$$F^{k,new} = \frac{\mathrm{OD}_k^2}{||\boldsymbol{E}||_2^2/(n_k - r - 1)}, \tag{5.10}$$

where $||\boldsymbol{E}||_2^2/(n_k - r - 1)$ is the adjustment coefficient for OD_k^2, which is calculated from the training set of the kth class. The classification rule in Wold (1976) assigns $\boldsymbol{x}_{new}$ to the class with the smallest $F^{k,new}$.

Recently, a linear combination of OD_k^2 and SD_k^2 is often used as the classification rule, such as the classification rule in the robust SIMCA (RSIMCA) (Branden and Hubert, 2005) and in the SIMCA function from the PLS Toolbox in MATLAB. The difference between these classification rules is dependent on the coefficients used in the linear combination. In this chapter, we use the classification rule in RSIMCA as a representative of this category of classification rules. In RSIMCA, the following classification rule is used:

$$\gamma\left(\frac{\mathrm{OD}_k^2}{c_{\mathrm{OD}^2}^k}\right)^2+(1-\gamma)\left(\frac{\mathrm{SD}_k^2}{c_{\mathrm{SD}^2}^k}\right)^2, \tag{5.11}$$

where $\gamma \in [0,1]$ and $c_{\mathrm{OD}^2}^k$ and $c_{\mathrm{SD}^2}^k$ are the cutoff values of OD_k^2 and SD_k^2 calculated from the training set of the kth class. When $\gamma = 1$, (5.11) only depends on OD_k^2, and is set the same as (5.10) if the cutoff value $c_{\mathrm{OD}^2}^k$ in (5.11) is the same as the adjustment coefficient in (5.10). When $\gamma = 0$, (5.11) 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.

De Maesschalck et al. (1999) propose to use the Hawkin's distance and the Gnanadesikan's distance as the classification rule. However, these distances are based on the formulae in De Maesschalck et al. (1999), which are not suitable for high-dimensional data. Thus we do not discuss these two distances here.

5.1.2 A general formulation for the classification rules in SIMCA

Although the classification rules used in SIMCA are in different forms, as shown in (5.10) and (5.11), we shall show that they can be written using the following general formulation:

$$\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{M}_1^k(\boldsymbol{x}_{(c)}^{k,new})^T-\boldsymbol{t}^{k,new}\boldsymbol{M}_2^k(\boldsymbol{t}^{k,new})^T, \tag{5.12}$$

with different $\boldsymbol{M}_1^k \in \mathbb{R}^{p\times p}$ and $\boldsymbol{M}_2^k \in \mathbb{R}^{r_k\times r_k}$. In this section, we derive this general formulation based on the classification rules (5.10) and (5.11), and show $\boldsymbol{M}_1^k$ and $\boldsymbol{M}_2^k$ for (5.10) and (5.11), respectively. Based on the derived general formulation of the classification rules, we will define the distance to subspace and propose a method to learn distance to subspace in the next section.

Substituting (5.6) into (5.5), we obtain

$$\begin{aligned}
\mathrm{OD}_k^2 &= (\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k)(\boldsymbol{x}_{(c)}^{k,new} - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k)^T \\
&= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - 2\boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k(\boldsymbol{x}_{(c)}^{k,new})^T + \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k^2(\boldsymbol{x}_{(c)}^{k,new})^T \\
&= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{P}_k(\boldsymbol{x}_{(c)}^{k,new})^T \\
&= \boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T, \qquad (5.13)
\end{aligned}$$

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 $\mathscr{L}_k$ in Figure 5.2.

Then the classification rule (5.10) can be written as

$$\begin{aligned}
F^{k,new} &= \frac{\boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T}{||\boldsymbol{E}||_2^2/(n_k - r_k - 1)} \\
&= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{M}_{1(Wold)}^k(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}\boldsymbol{M}_{2(Wold)}^k(\boldsymbol{t}^{k,new})^T, \qquad (5.14)
\end{aligned}$$

where $\boldsymbol{M}_{1(Wold)}^k = \frac{1}{h_1}\boldsymbol{I}_p$, $\boldsymbol{M}_{2(Wold)}^k = \frac{1}{h_1}\boldsymbol{I}_{r_k}$ and $h_1 = ||\boldsymbol{E}||_2^2/(n_k - r_k - 1)$. Equation (5.14) indicates that the classification rule of Wold 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 RSIMCA, we substitute (5.13) to (5.11):

$$\begin{aligned}
&\frac{\gamma}{(c_{\mathrm{OD}^2}^k)^2}(\boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T) + \frac{1-\gamma}{(c_{\mathrm{SD}^2}^k)^2}\boldsymbol{t}^{k,new}\boldsymbol{D}_r^{-2}(\boldsymbol{t}^{k,new})^T \\
&= \frac{\gamma}{(c_{\mathrm{OD}^2}^k)^2}\boldsymbol{x}_{(c)}^{k,new}(\boldsymbol{x}_{(c)}^{k,new})^T - \sum_{i=1}^{r}(-\frac{1-\gamma}{(c_{\mathrm{SD}^2}^k)^2} + \frac{\gamma}{(c_{\mathrm{OD}^2}^k)^2 d_i^2})t_i^2 \\
&= \boldsymbol{x}_{(c)}^{k,new}\boldsymbol{M}_{1(R)}^k(\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new}\boldsymbol{M}_{2(R)}^k(\boldsymbol{t}^{k,new})^T, \qquad (5.15)
\end{aligned}$$

where $\boldsymbol{M}_{1(R)}^k = \frac{1}{h_2}\boldsymbol{I}_p$, $h_2 = \frac{\gamma}{(c_{\mathrm{OD}^2}^k)^2}$ and $\boldsymbol{M}_{2(R)}^k$ 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 Wold in (5.14), (5.15) indicates that the classification rule of RSIMCA 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 different weights to the r_k dimensions in the linear combination of the scores $\boldsymbol{t}^{k,new}(\boldsymbol{t}^{k,new})^T$.

5.1.3 Learning distance to subspace

In Wold (1976) and Branden and Hubert (2005), $\boldsymbol{x}_{new}$ is classified to the class with the smallest value calculated from (5.10) and (5.11), respectively. Thus, using the general formulation, we classify $\boldsymbol{x}_{new}$ to the class with the smallest value calculated from (5.12). We define the general formulation (5.12) as the distance from $\boldsymbol{x}_{new}$ to the kth class subspace. In this way, the classification rule is to assign $\boldsymbol{x}_{new}$ to the nearest class subspace based on the distance to subspace defined in (5.12).

The distance to subspace for the kth class defined in (5.12) 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 $\mathscr{L}_k$, 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 $\mathscr{L}_k$ to the centre of $\mathscr{L}_k$.

$\boldsymbol{M}^k_1$ 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 Wold (1976) and Branden and Hubert (2005), 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.

Established distance metric learning methods are sample-based, i.e. the distances are measured between samples. However, in SIMCA, the distance is calculated between a sample and a class subspace. Thus we need to develop a method of learning the distance metric from sample to subspace, to learn the distance metrics in SIMCA. 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 to 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.

5.1.3.1 Distance metric

In this section, we briefly introduce 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 liked pseudo metric:

$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)\boldsymbol{M}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T}, \tag{5.16}$$

which is parameterised by $\boldsymbol{M}$. $\boldsymbol{M}$ 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 original Mahalanobis distance. If $\boldsymbol{M}$ is the identity matrix, then $d_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is exactly the Euclidean distance.

5.1.3.2 Distance to subspace

Different from the distance metric between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ defined in (5.16), we define the squared distance metric between a sample $\boldsymbol{x}$ and a class subspace $\mathscr{L}_k$ using the general formulation in (5.12):

$$d^2(\boldsymbol{x}, \mathscr{L}_k) = \boldsymbol{x}^k_{(c)} \boldsymbol{M}^k_1 (\boldsymbol{x}^k_{(c)})^T - \boldsymbol{t}^k \boldsymbol{M}^k_2 (\boldsymbol{t}^k)^T, \tag{5.17}$$

where $\boldsymbol{x}^k_{(c)}$ denotes the sample mean-centred by the mean of the training samples of the kth class, $\boldsymbol{M}^k_1 \in \mathbb{R}^{p \times p}$ 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 $\boldsymbol{M}^k_2 \in \mathbb{R}^{r_k \times r_k}$ is the parameterisation matrix for the distance in the PC subspace of the kth class. Then $d^2(\boldsymbol{x}, \mathscr{L}_k)$ 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 $\mathscr{L}_k$ and the squared distance from the projection of the sample to the centre of $\mathscr{L}_k$.

5.1.3.3 Learned distance to subspace

To learn good distance metrics between samples and class subspaces, we propose the following similarity/dissimilarity constraints:

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

$\boldsymbol{D} = \{(\boldsymbol{x}_i, \mathscr{L}_k) \mid \boldsymbol{x}_i$ 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 $\boldsymbol{X}_1 = [\boldsymbol{x}^T_{1(1)}, \dots, \boldsymbol{x}^T_{1(n_1)}]^T \in \mathbb{R}^{n_1 \times p}$, 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 $\boldsymbol{X}_2 = [\boldsymbol{x}^T_{2(1)}, \dots, \boldsymbol{x}^T_{2(n_2)}]^T \in \mathbb{R}^{n_2 \times p}$. Thus the similarity/dissimilarity constraints become

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

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

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 constraint $\boldsymbol{S}$, while maximise the sum of those that fall into the dissimilarity constraint $\boldsymbol{D}$. However, simply optimising the sums of the distances suffers from loosing 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 that to the correct class for each training sample large enough for classification by using the following constraints:

$$\begin{aligned} &d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_2) - d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_1) \geq 1, \text{ for } i = 1, \dots, n_1, \text{ and} \\ &d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_1) - d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_2) \geq 1, \text{ for } j = 1, \dots, n_2. \end{aligned} \tag{5.18}$$

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:

$$\min_{\xi_{1(i)}, \xi_{2(j)}, \boldsymbol{M}_1^k, \boldsymbol{M}_2^k} \quad \sum_{i=1}^{n_1} \xi_{1(i)} + \sum_{j=1}^{n_2} \xi_{2(j)} \tag{5.19}$$

$$\text{s.t.} \quad d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_2) - d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_1) \geq 1 - \xi_{1(i)}, \;\; \xi_{1(i)} \geq 0, \tag{5.20}$$

$$d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_1) - d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_2) \geq 1 - \xi_{2(j)}, \;\; \xi_{2(j)} \geq 0, \tag{5.21}$$

$$\boldsymbol{M}_1^k \geq 0 \text{ and } \boldsymbol{M}_2^k \geq 0. \tag{5.22}$$

The constraints in (5.20) and (5.21) can be rewritten as

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

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

$$\begin{aligned} \min_{\boldsymbol{M}_1^k, \boldsymbol{M}_2^k} \; &\sum_{i=1}^{n_1} [1 + d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_1) - d^2(\boldsymbol{x}_{1(i)}, \mathscr{L}_2)]_+ + \\ &\sum_{j=1}^{n_2} [1 + d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_2) - d^2(\boldsymbol{x}_{2(j)}, \mathscr{L}_1)]_+ \\ &\text{s.t. } \boldsymbol{M}_1^k \geq 0, \;\; \boldsymbol{M}_2^k \geq 0. \end{aligned} \tag{5.23}$$

The hinge losses used in (5.23) only penalise the samples that do not satisfy (5.18), while assign zero loss to the correct class for the samples that satisfy (5.18) using SIMCA. In this way, the hinge loss makes full use of the effectiveness of SIMCA. 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 (Weinberger et al., 2006).

Suppose we denote $\boldsymbol{M}_1^{k*}$ and $\boldsymbol{M}_2^{k*}$ $(k = 1,2)$ as the solutions of (5.23). Then the distance from a test sample $\boldsymbol{x}_{new}$ to the kth class subspace is

$$d^2(\boldsymbol{x}_{new}, \mathscr{L}_k) = \boldsymbol{x}_{(c)}^{k,new} \boldsymbol{M}_1^{k*} (\boldsymbol{x}_{(c)}^{k,new})^T - \boldsymbol{t}^{k,new} \boldsymbol{M}_2^{k*} (\boldsymbol{t}^{k,new})^T. \tag{5.24}$$

We compare $d^2(\boldsymbol{x}_{new}, \mathscr{L}_1)$ and $d^2(\boldsymbol{x}_{new}, \mathscr{L}_2)$, 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 (5.14) and (5.15), $\boldsymbol{M}_{1(Wold)}^k = \frac{1}{h_1}\boldsymbol{I}_p$ and $\boldsymbol{M}_{1(R)}^k = \frac{1}{h_2}\boldsymbol{I}_p$ are identity matrices with common coefficients $1/h_1$ and $1/h_2$ for all dimensions, respectively. Therefore, in this chapter, we learn $\boldsymbol{M}_1^k = c_k\boldsymbol{I}_p$(with $c_k \geq 0$) and $\boldsymbol{M}_2^k = \text{diag}(m_{21}^k, m_{22}^k, \ldots, m_{2r_k}^k)$ (with each element nonnegative), as natural and practically-interpretable extensions of those used in (5.14) and (5.15).

5.2 Experiments

In the following experiments, SIMCA with classification rule (5.10) (SIMCA-W), SIMCA with classification rule (5.11) (SIMCA-R) and SIMCA with classification rule (5.24) (SIMCA-LD2S) are compared using the Phenyl dataset, the fat dataset (Ferraty and Vieu, 2006) and the meat dataset (Arnalds et al., 2004). Detailed descriptions for the three datasets can be found in Chapter 2 Section 2.2.1.

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

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

For the fat dataset, we use all 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 55 chicken samples and 54 turkey samples in the dataset. We also select the first 100 dimensions from the 350 dimensions for the experiments in this chapter, i.e. $p = 100$.

Therefore, for the three datasets, the five training set sizes are $n_1 = n_2 = 10$, $n_1 = n_2 = 20$, $n_1 = n_2 = 30$, $n_1 = n_2 = 40$ and $n_1 = n_2 = 50$. The rest of the samples in the datasets are used as test samples.

5.2.1 Experiment settings

In SIMCA-W, SIMCA-R and SIMCA-LD2S, the numbers of PCs, r_k, are tuned by 5-fold cross-validation using the training set to minimise the classification error.

In SIMCA-R, $c^k_{OD} = (\hat{\mu} + \hat{\sigma} z_{0.975})^{3/2}$, 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.

In SIMCA-LD2S, the optimisation problem (5.23) is solved by 'cvx' in MATLAB.

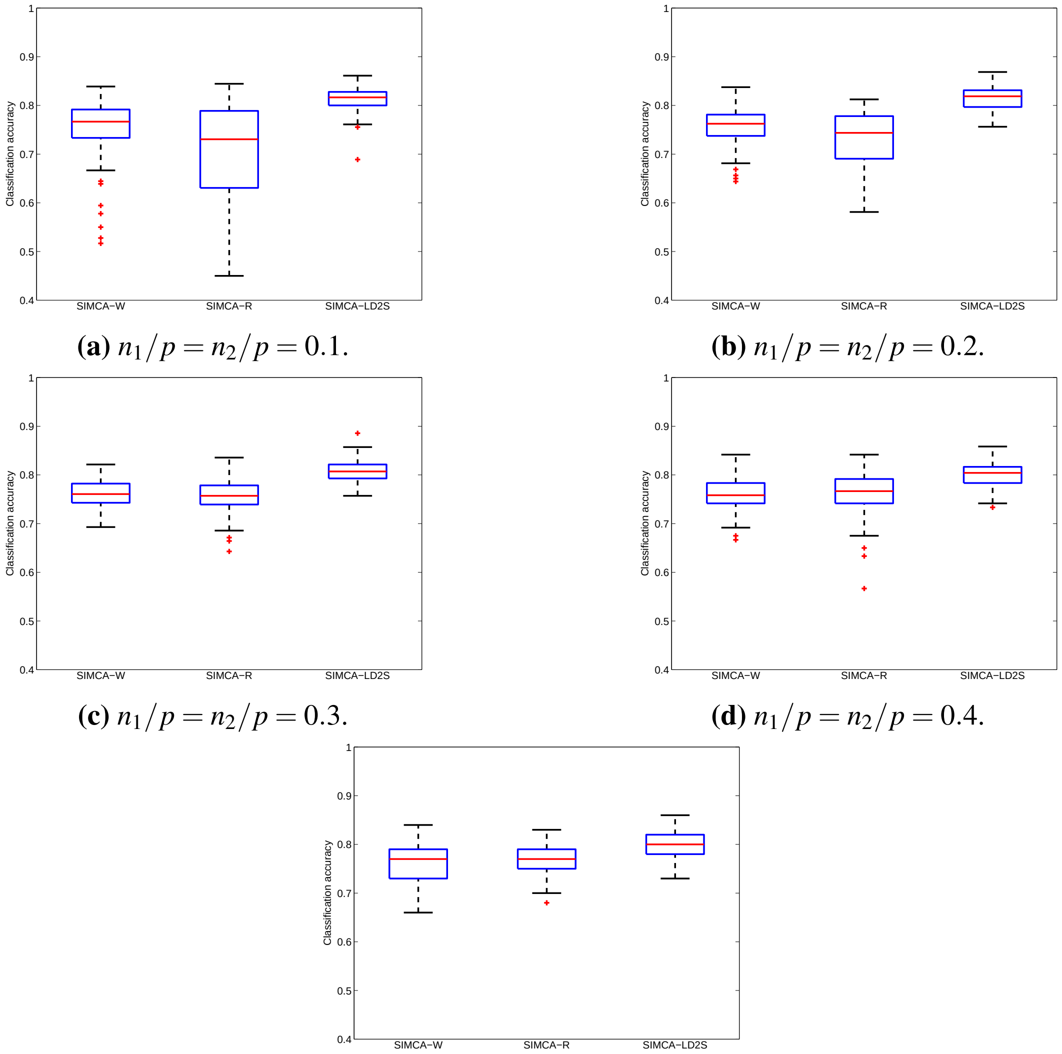
All the experiments are repeated 100 times and the classification accuracies are recorded.

5.2.2 Results

5.2.2.1 The Phenyl dataset

The classification results of the Phenyl dataset demonstrate the superior classification performance of SIMCA-LD2S, as shown in Figure 5.3 and Figure 5.4, compared with SIMCA-W and SIMCA-R over all n_k/p ratios.

However, the classification performance of SIMCA-LD2S cannot always be



(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 5.3: Classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the Phenyl dataset.

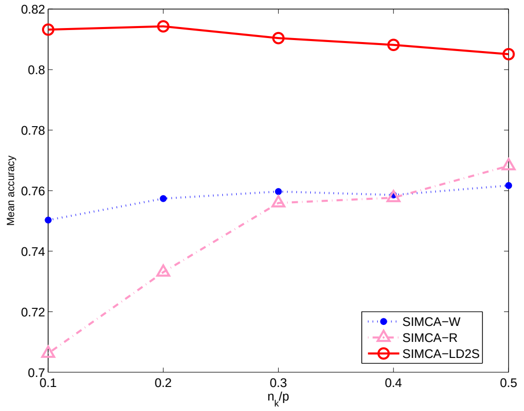
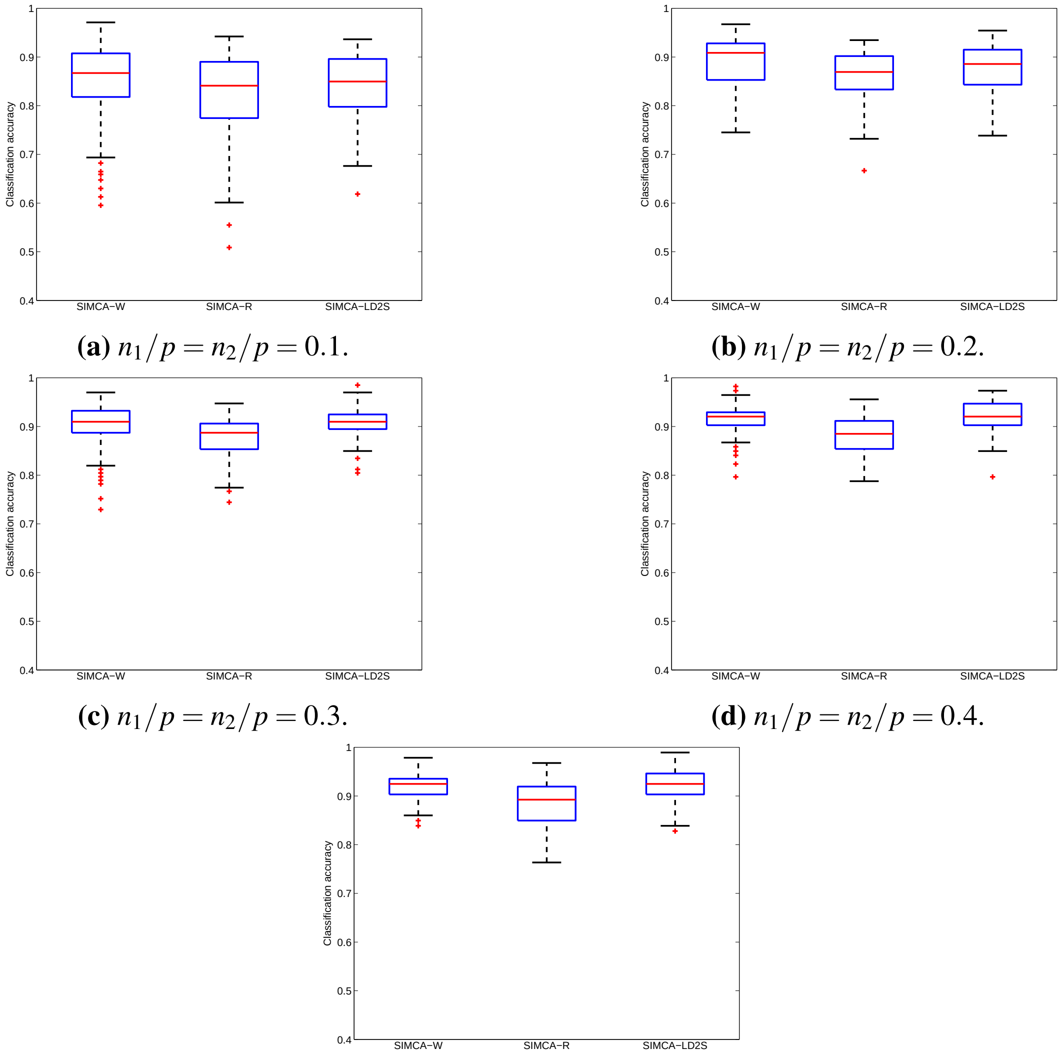


Figure 5.4: Mean classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the Phenyl dataset.

better than those of SIMCA-W and SIMCA-R over all scenarios, in particular under small n_k/p ratios. In the following two sections, we show two examples that SIMCA-LD2S performs worse than SIMCA-W and SIMCA-R for small n_k/p ratios while better than SIMCA-W and SIMCA-R for large n_k/p ratios. This is because there are more parameters in SIMCA-LD2S to be learned than SIMCA-W and SIMCA-R, and SIMCA-LD2S needs more training samples to achieve good classification performance for some data.

5.2.2.2 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 5.5: Classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the fat dataset.

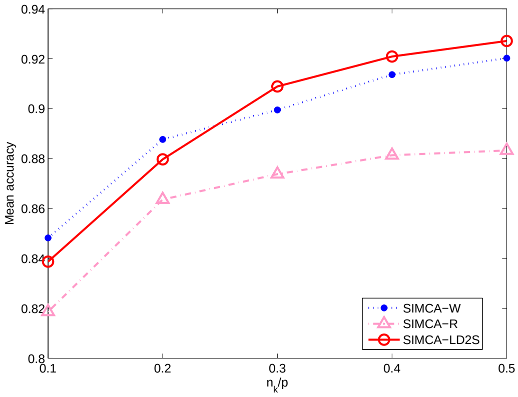


Figure 5.6: Mean classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the fat dataset.

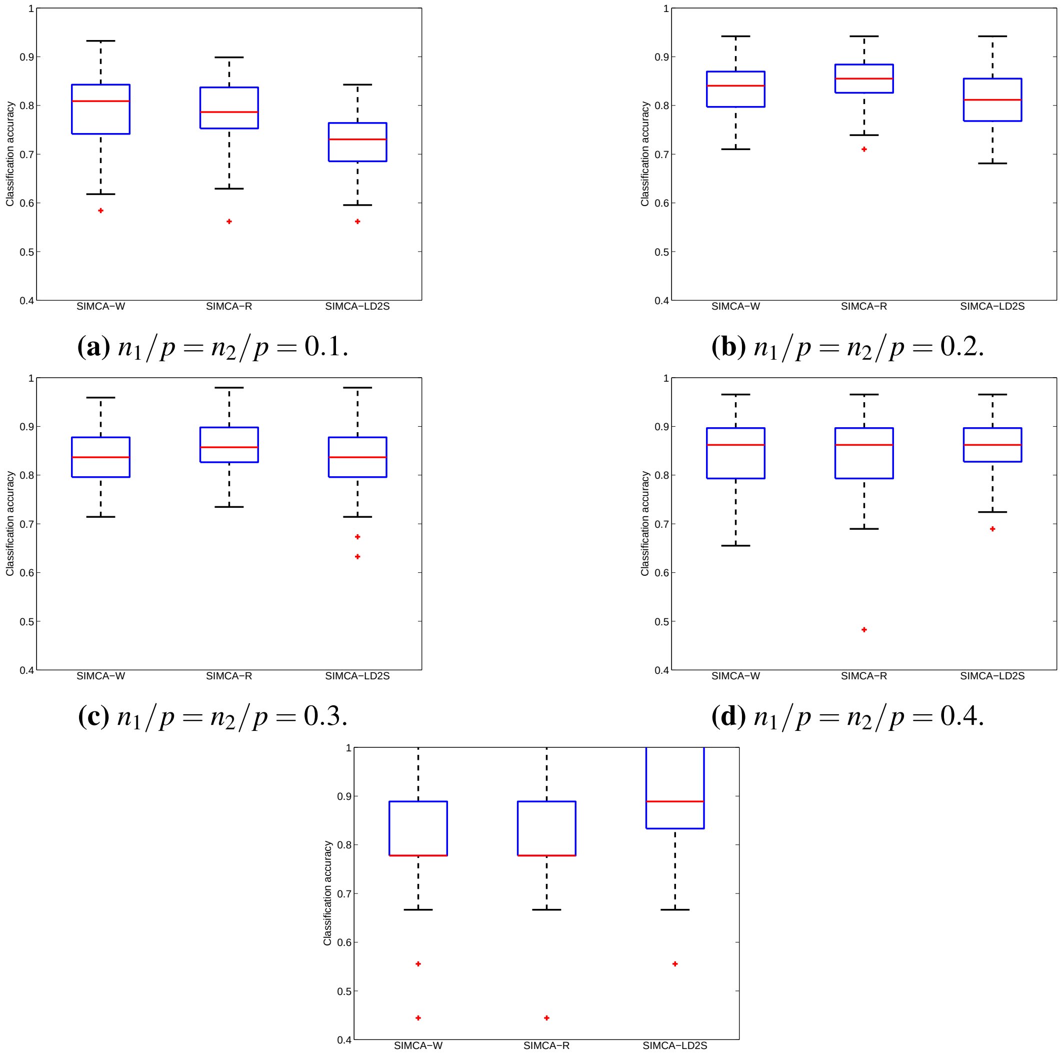
In the fat dataset, the classification performance of SIMCA-LD2S is worse than SIMCA-W when $n_k/p < 0.3$ and is better than SIMCA-W when $n_k/p \geq 0.3$, as shown in Figure 5.5 and Figure 5.6. SIMCA-R provides the worst classification accuracies over all n_k/p ratios.

5.2.2.3 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 $n_k/p < 0.4$, SIMCA-LD2S performs much worse than SIMCA-W and SIMCA-R, especially for $n_k/p = 0.1$. However, when $n_k/p = 0.5$, the classification accuracies of SIMCA-LD2S become much better than those of SIMCA-W and SIMCA-R, as shown in Figure 5.7(e) and Figure 5.8. The classification results of the meat dataset suggest that SIMCA-LD2S needs $n_k/p > 0.4$ to achieve superior classification performance for the meat dataset.

5.2.2.4 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 classification rule based on LD2S may perform poorly in classification since the n_k/p ratio is not large enough to learn the parameters in LD2S.



(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 5.7: Classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the meat dataset.

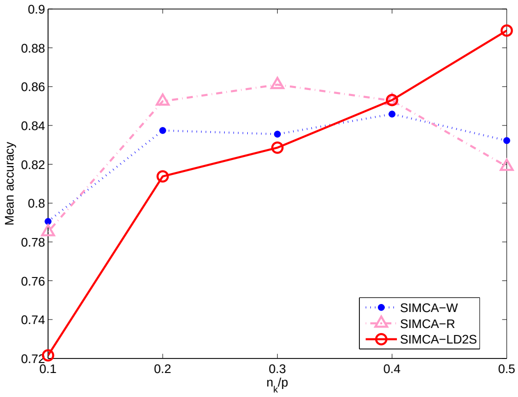


Figure 5.8: Mean classification accuracies of SIMCA-W, SIMCA-R and SIMCA-LD2S for the meat dataset.

5.3 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 SIMCA. The classification performances on three datasets demonstrate the effectiveness of learning distance metrics from data when the n_k/p ratio is large enough.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we present four reinforcements for SIMCA, with the first two related to the class models used in SIMCA and the last two related to the distances used in SIMCA.

First, SIMCA suffers from the problem that the class subspaces are built separately ignoring the discriminative between-class information. We have tackled this problem by projecting the original data to a subspace more discriminative than the original feature space before applying SIMCA. We have proposed the DOS projection to generate such a discriminative subspace which is spanned by the eigenvectors of the generating matrix with high discriminative ability. The experiments on three real-world spectral datasets have demonstrated the effectiveness of the DOS projection.

Second, we have proposed NCCM and have provided a thorough comparison of NSM, NCHM and NCCM theoretically and empirically. We have proved the theoretical dual analysis results for the dual problem of NCCM, based on the relationship between a convex cone and its polar cone. We have also established an SHC framework for nearest-class-model methods with arbitrary norms to inspire future research. Empirically, we have proposed an effective method to explore the properties of the data.

Third, we have investigated the calculations of the two OD^2, $v^{k,l}$ and $v^{k,new}$, using the formulae in De Maesschalck et al. (1999). We have shown that the formula for $v^{k,new}$ in De Maesschalck et al. (1999) is not precise for high-dimensional data. The experiments on both the simulated datasets and the real datasets have confirmed that using the formula in De Maesschalck et al. (1999) can proved worse classification results than using the original one in Wold (1976). Therefore, we suggest to use the original formulae in Wold (1976) to calculate the OD^{2}s for high-dimensional data.

Fourth, we have derived a general formulation for the classification rules used in literature and have defined it as the distance metric to subspace. We have proposed to learn the two parameterisation matrices in the distance metric to subspace adaptively from data using the learning distance to subspace method. Our proposed LD2S forces the samples to be closer to their correct class subspaces while be farther away from their wrong class subspaces.

6.2 Future work

Based on our work in the thesis, more research on SIMCA and subspace-based classification methods could be done in the future.

Firstly, subspace-based classification methods have been generalised to multi-view or tensor versions (Zhang et al., 2013, 2015, 2016) recently. Inspired by these research, it is interesting to extend the DOS projection to multi-view or tensor versions.

Secondly, further improvements for learning distance to subspace could be developed, such as adding regularisation of the matrices to be learned in the optimisation problem. Besides the spectral data, the method of learning distance to subspace could be applied to other types of data. As nature extensions of the work in Chapter 5, we could first extend $\boldsymbol{M}_1^k$ to diagonal matrices, instead of the simple identity matrices with learned coefficients in Chapter 5. Furthermore, with sufficient amount of data, we could even learn the full parameterisation matrices instead of the simple diagonal matrices to improve the classification performance, since the full matrices allow more variability. Advanced techniques are needed to learn tailored full matrices.

Thirdly, the classification methods discussed in this thesis are only applied to high-dimensional spectral data which only measure the spectral information of the samples. Hyperspectral image data are 3-dimensional data cubes that measure both spectral and spatial information and have attracted a lot of attentions recently. It is natural to extend the methods proposed in this thesis to classify hyperspectral image data. In such extensions, how to include the spatial information from the images is an interesting work meriting investigation.

Appendix A

Spectral Nonlocal Restoration of Hyperspectral Images With Low-Rank Property

Spectral Nonlocal Restoration of Hyperspectral Images With Low-Rank Property

Rui Zhu, Mingzhi Dong, and Jing-Hao Xue

***Abstract*—Restoration is important in preprocessing hyperspectral images (HSI) to improve their visual quality and the accuracy in target detection or classification. In this paper, we propose a new low-rank spectral nonlocal approach (LRSNL) to the simultaneous removal of a mixture of different types of noises, such as Gaussian noises, salt and pepper impulse noises, and fixed-pattern noises including stripes and dead pixel lines. The low-rank (LR) property is exploited to obtain precleaned patches, which can then be better clustered in our spectral nonlocal method (SNL). The SNL method takes both spectral and spatial information into consideration to remove mixed noises as well as preserve the fine structures of images. Experiments on both synthetic and real data demonstrate that LRSNL, although simple, is an effective approach to the restoration of HSI.**

***Index Terms*—Hyperspectral image, low rank (LR), nonlocal means, restoration, spectral and spatial information.**

I. Introduction

HYPERSPECTRAL images (HSI) are captured on 100s of narrow spectral bands ranging from 400 to 2400 nm, represented as a three-dimensional (3-D) data cube containing both spectral and spatial information. During the capture of HSI, various kinds of noises are introduced, polluting the images. The noises also affect further HSI applications such as classification, target detection, and unmixing. In order to recover clean images and facilitate further applications, image restoration is required as a preprocessing.

The restoration of HSI has attracted considerable attention recently [1]–[10]. The 3-D representation of HSI makes the HSI restoration different from the traditional two-dimensional (2-D) image restoration, with both spectral and spatial information at our disposal.

Common denoising methods, such as maximum noise fraction (MNF) [4], orthogonal, or oblique subspace projection [5], [6], and frequency domain filtering [7], [8], reconstruct the image in a transformed domain. They, however, fail to restore image edges effectively. Wavelet-based restoration methods [8]–[10] can preserve details of images such as edges. However, it depends on prior knowledge to choose an appropriate type of wavelet transform. Besides being represented in a transformed domain, spatial information in the original image can be exploited directly. Most of the methods that consider spatial information are based on local information from neighbouring pixels. However, local methods exploit limited information of the true image. In contrast, nonlocal approaches use information from the whole image, based on the assumption that a small patch of the image can be represented by similar patches in other places of the same image [11]. In this way, the fine spatial structures of the image can be preserved. Qian and Ye [1] adopted this idea and applied a nonlocal sparse model to the HSI restoration, in which the overlapped patches of the image are clustered and a sparse learning method is applied to each cluster. In [1], patches in each cluster are assumed to be represented by the same dictionary. However, how to choose the dictionary is based on certain prior knowledge.

Without using prior knowledge, Golbabaee and Vandergheynst [2] and Zhang *et al.* [3] solved the HSI restoration problem utilising the low-rank (LR) property of HSI. The LR property can be attributed to the high correlation between hyperspectral signatures of pixels. Hence, the images can be expressed by a linear combination of a limited number of endmembers. In [3], an LR matrix recovery model was developed to simultaneously remove several types of noises, such as Gaussian noises, impulse noises, stripes, and dead lines. Stripes and dead lines are fixed-pattern bad pixels due to variations in detection [5], [8], [12]. Impulse noises, stripes, and dead lines can be sparse, since they only appear in few bands or few pixels within a band.

However, the LR methods, mainly exploiting the spectral correlation between spectral bands, may not preserve fine spatial structures. On the other hand, the nonlocal techniques mainly exploit the spatial correlation between spatial patches.

Hence, to exploit the best of both worlds, in this paper we propose a new low-rank spectral nonlocal (LRSNL) approach, which will consider both spectral and spatial information. It combines both the LR property of HSI and the nonlocal method for the HSI restoration. In addition, we extend the standard nonlocal approach for 2-D images to 3-D HSI, using spectral information to remove the mixed noises as well as preserve the fine spatial structures of the image.

II. Methodology

The proposed HSI restoration approach (LRSNL) contains two major parts: 1) using the LR property to obtain pre-cleaned patches and 2) applying the spectral nonlocal (SNL) method to restore the image. The LR precleaning is to improve the performance of the nonlocal restoration. The importance of precleaning has been shown in the experiments of [13] and [14], where better clustering results of the patches are obtained after a

Manuscript received October 27, 2014; accepted November 10, 2014. Date of publication December 01, 2014; date of current version July 30, 2015.

The authors are with the Department of Statistical Science, University College London, London WC1E 6BT, U.K. (e-mail: rui.zhu.12@ucl.ac.uk; mingzhi.dong.13@ucl.ac.uk; jinghao.xue@ucl.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2014.2370062

first round of denoising. We shall also demonstrate this through our experiments.

A. LR Precleaning of HSI

To explain the LR property of HSI, we first transform the 3-D data cube into a 2-D matrix. Suppose the size of an HSI data cube is $M \times N \times Q$, where M and N represent the total numbers of pixels in height and width, and Q is the number of spectral bands. The cube can be rearranged as a 2-D matrix of $(M \times N) \times Q$, with each column representing the reflectance from a specific spectral band, and each row representing the spectral signature of a specific pixel. Note that the spatial information is nevertheless lost after this transformation.

The LR property can be associated with the linear mixing model of HSI. In the linear mixing model, HSI are considered as a linear mixture of several endmembers: $\widetilde{\boldsymbol{U}} = \mathbf{A}\mathbf{S}^T$, where $\widetilde{\boldsymbol{U}}$ is the transformed 2-D matrix of the HSI and $\mathbf{A}$ is an $(M \times N) \times K$ matrix representing the abundance of K endmembers; the endmembers are concatenated into a $Q \times K$ matrix $\mathbf{S}$. Since there are a limited number of endmembers, the rank of $\widetilde{\boldsymbol{U}}$ is limited [2].

The captured noisy HSI can be modeled as

$$\boldsymbol{V} = \boldsymbol{U} + \boldsymbol{N} \tag{1}$$

where $\boldsymbol{V}$ is the noisy HSI cube, $\boldsymbol{U}$ is the true, clean HSI cube, and $\mathbf{N}$ denotes the noise [15].

To preclean the noisy $\boldsymbol{V}$, the HSI cube is first divided into small patches of size $m \times m \times Q$, where m is much smaller than $\min(M, N)$. Each patch is centred at a pixel, thus the number of patches is $M \times N$. All the patches are transformed to 2-D matrices of size $(m \times m) \times Q$. For pixel (i, j), $i = 1, \ldots, M$ and $j = 1, \ldots, N$, its noisy patch matrix $\boldsymbol{V}_{ij}$ is precleaned by using the LR property of HSI

$$\widehat{U}_{ij} = \arg\min_{U_{ij}} \| V_{ij} - U_{ij} \|_F^2 \ \text{ s.t. } \ rank(\boldsymbol{U}_{ij}) \leq K \tag{2}$$

where $\boldsymbol{V}_{ij}$ and $\boldsymbol{U}_{ij}$ denote the noisy and clean patch matrices centred at (i, j), respectively, $\| \cdot \|_F$ denotes the Frobenius norm of matrix, and K is a predefined constant that indicates the maximal rank of the clean patch matrix [15].

As we mentioned, the LR methods only consider the spectral correlation, and thus may not preserve the fine spatial structures of the image. Fig. 1 shows the LR restoration results from LRMR [3] for two bands of a synthetic Indian Pines dataset. (The construction of this synthetic dataset will be detailed in Section III-A). We can observe that in both cases using only the LR property tends to over-smooth the images. To further recover the fine spatial structures, we propose a spectral nonlocal approach.

B. Spectral Nonlocal Restoration of HSI

The standard nonlocal means algorithm (NL) for 2-D images [11] considers the spatial information of images and aims to preserve the fine structures during image restoration. In NL, the image is divided into small patches and each pixel is restored as the weighted average of the pixels that have a neighborhood similar to the neighborhood of the target pixel. Although NL can effectively remove Gaussian noises, it cannot handle fixed-pattern noises such as dead pixel lines and stripes. For a dead pixel, the pixels that have the most similar neighbors will be the neighboring dead pixels, hence the neighboring dead pixels will have large weights and the restoration of a dead pixel is still a dead pixel.

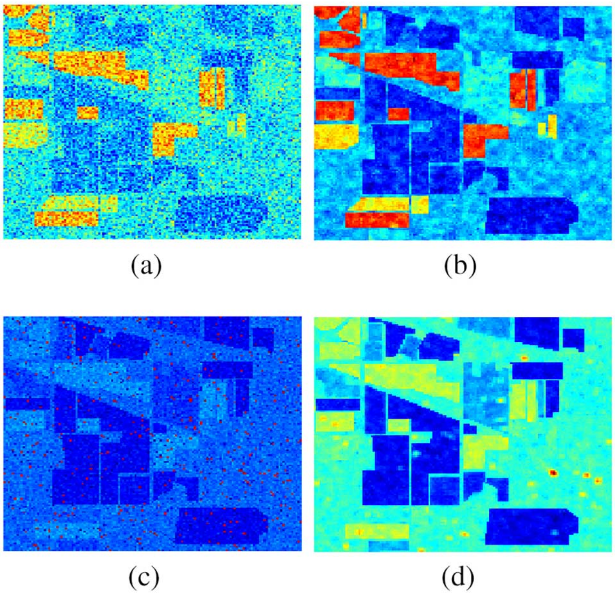
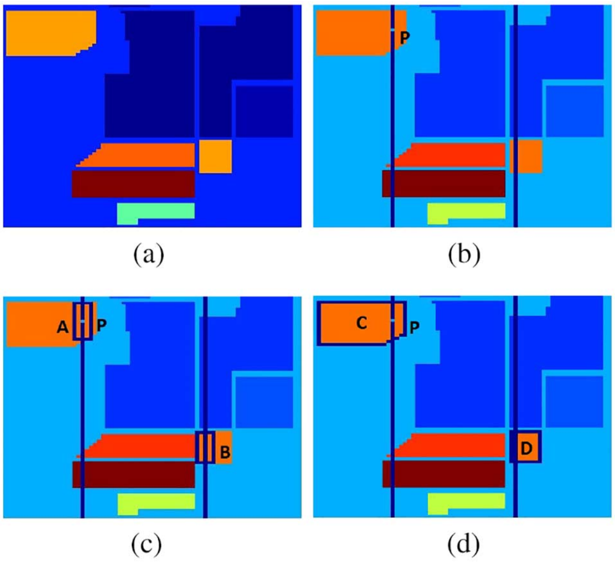


Fig. 1. LR restoration of two images: images with (a) Gaussian noises and (c) a mixture of Gaussian and impulse noises, and their LRMR results in (b) and (d), respectively.

To extend NL for HIS reconstruction, we incorporate the spectral information into NL. In our proposed method LRSNL, we assume that the weights of pixels, that have a neighborhood similar to that of the target pixel, are the same over all spectral bands. These weights are thus calculated based on the mean dissimilarity between patches over all bands. As a result, if dead lines and stripes are few, the effect of these noises will be small and the bands containing these noises can be restored by using information from other spectral bands. In this way, we extend the standard NL to a SNL, such that it can be readily applied to HIS to reduce various types of noises.

Fig. 2 illustrates the difference between NL and SNL for HIS. Fig. 2(a) shows a part of a spectral band of the Indian Pines synthetic data. The areas with the same colour have the same land cover. Fig. 2(b) shows the noisy image with two dead pixel lines, and P is a dead pixel on the left-hand line. The colour of P, different from other dead pixels, is to visually indicate its position. The true value of pixel P is 0.190 and the noisy value is 0. Fig. 2(c) and (d) shows the pixels similar to P found by NL and SNL, respectively. The dead pixels in squares A and B are the similar pixels found by NL, so clearly P will be restored as a dead pixel with value remaining 0. In contrast, the similar pixels found by SNL are all the pixels in squares C and D. Although there are dead pixels in the two squares, a large number of normal pixels will overwhelm the influence of the dead pixels. The restored value of P by using SNL is actually 0.178, close to its true value.



(a) (b) (c) (d)

Fig. 2. Comparison of NL and SNL for dead lines: (a) original; (b) noisy; (c) NL; and (d) SNL.

Let us describe the SNL algorithm as follows. Instead of calculating the similarity between patches based on the precleaned 2-D matrix, we transform the 2-D matrix back to the 3-D cube and calculate the similarity based on this cube. The dissimilarity between two patches, respectively, centred at pixels (i, j) and (k, l), can be defined as

$$D_{ij,kl} = \frac{1}{Q}\sum_{q=1}^{Q} \| \widehat{U}_{ij,q} - \widehat{U}_{kl,q} \|_F^2, \quad k \neq i \text{ or } l \neq j \tag{3}$$

where q indexes the spectral bands. The pixel (i, j) can be recovered by a weighted average of all other pixels in the image. The weight that pixel (k, l) carries to pixel (i, j) can be expressed as

$$w_{ij,kl} = \frac{e^{-D_{ij,kl}/h^2}}{\sum_{k,l} e^{-D_{ij,kl}/h^2}} \tag{4}$$

where h is the parameter indicating the decay of the exponential function, which reduces the weight with the dissimilarity between the two patches.

From (3), we can see that the dissimilarity between two patches is evaluated as the average of the dissimilarity over all spectral bands. That is, the weights for restoring each pixel take advantage of the spectral information available. Hence, pixels affected by impulse noises or dead pixels can then be restored through using information from other spectral bands.

In NL and SNL, each patch is compared with all other patches and all the associated weights are calculated. This will result in high-computational costs when the image is large. To reduce the costs, Buades *et al.* [11] suggest to set a searching area, compute the dissimilarity between the patches within this area, and restore a pixel based on the weighted average only within this area.

Although the proposed SNL can remove mixed noises and preserve the fine structures of images, it cannot perform well when pixel values are largely affected by noises since the pixels are restored as the weighted average of pixels within the image. Using LR as a precleaning step will remove some noises and thus lead to better clustering and restoration.

Algorithm 1. LRSNL: Low-Rank Spectral Nonlocal

```
Input: $\boldsymbol{V}$, $m$, $K$, $h$
Output: $\boldsymbol{U}^{cleaned}$
1: Divide the data cube $\boldsymbol{V}$ into overlapped patches of size $m \times m \times Q$. Transform each patch into a 2-D matrix of size $(m \times m) \times Q$.
2: Preclean patches using the low-rank property as (2).
3: Calculate the weights between the precleaned patches using (3) and (4).
4: Restore each pixel using the weighted average of all other pixels in the searching area to obtain $\boldsymbol{U}^{cleaned}$.
```

Therefore, the proposed LRSNL can be summarized in Algorithm 1.

III. Experiments

A. Synthetic Data Experiments

1) Data and Experimental Settings: An Indian Pine dataset is used for our synthetic experiments. The dataset is created based on the ground truth of Indian Pine (http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes) and the spectral signatures from the USGS digital spectral library (http://speclab.cr.usgs.gov/spectral.lib06). The ground truth describes the real land cover materials of the Indian Pine area and thus this synthetic dataset can be viewed as clean HSI that represent a real-world situation. This dataset has been widely used for validating the techniques of hyperspectral image processing and analysis [1]. The image of Indian Pine is of size 145 × 145 and the spectral signatures in the library describe the reflectance of 223 spectral bands. According to the ground truth, pixels of the image are classified into 17 categories. Each pixel is assigned with a spectral signature based on its category. Thus, the synthetic data cube is of size 145 × 145 × 223 with reflectance values within range $[0, 0.5]$.

The performance of restoration methods is evaluated in two ways. First, the restored images and spectral signatures are shown directly for visual comparison. Since, there are numerous pixels and spectral bands, only a few of them are presented in this paper. Second, the performance is also quantitatively measured by the improved signal to noise ratio (ISNR) for each spectral band [1]

$$ISNR_i = 10\log_{10}\frac{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}[u_i^{noised}(x,y) - u_i(x,y)]^2}{\sum\limits_{x=1}^{M}\sum\limits_{y=1}^{N}[u_i^{cleaned}(x,y) - u_i(x,y)]^2} \tag{5}$$

where M and N are the numbers of rows and columns of the image of a specific spectral band, $u_i^{noised}(x,y)$ is the noisy value of a pixel (x, y) of band i, $u_i(x,y)$ is its true value, and $u_i^{cleaned}(x,y)$ is its restored value.

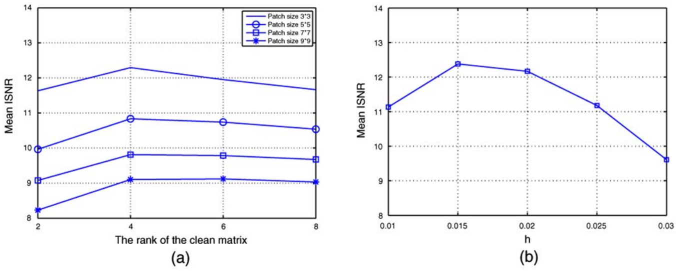


Fig. 3. Effect of tuning parameters: (a) the patch size and the rank of the clean matrix; and (b) the filtering parameter h.

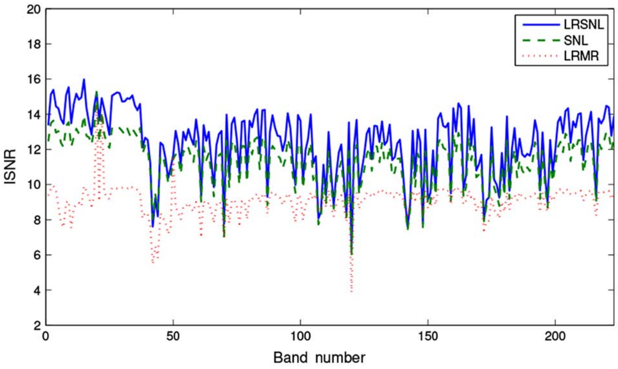


Fig. 4. ISNR of LRMR, SNL, and the proposed LRSNL.

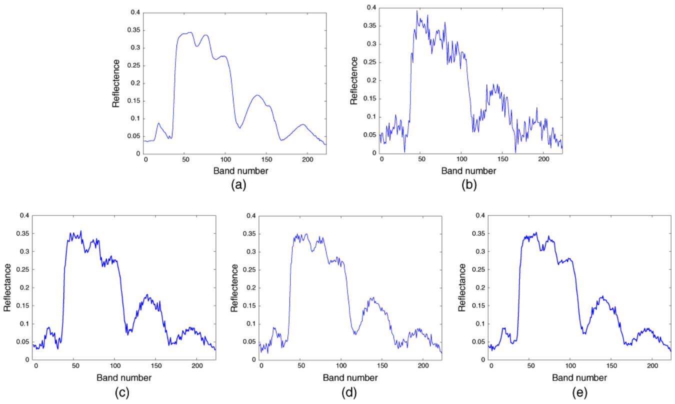


Fig. 5. Restoration of the spectral signature of pixel (136, 21): (a) original; (b) noisy; (c) LRMR; (d) SNL; and (e) LRSNL.

As with [3], our synthetic dataset covers four types of noises: 1) Gaussian noises with standard deviation ranging from 0.01 to 0.03 are added randomly to all the spectral bands; 2) 20% salt and pepper impulse noises are added to band 20 and band 22; 3) dead lines are added to band 5 to band 14 in the same positions; and (4) stripes are added to band 50 and 70. Due to the similarity between dead lines and stripes, we omit the presentation of results for stripes in this paper.

The proposed method (LRSNL) is compared with the LR matrix recovery method (LRMR) [3] and the SNL that does not have the LR precleaning step. LRMR transforms the 3-D cube into a 2-D matrix and takes advantage of the LR property of the 2-D matrix. The mixed noises are removed by using the LR matrix recovery model, which treats the clean image as a LR matrix and treats the noises, such as impulse noise and dead lines, as a sparse matrix. The GoDec algorithm [16] is used to solve the optimization problem in LRMR. We also compare LRSNL with SNL to show the effect of precleaning.

There are three parameters in Algorithm 1 to be tuned: the patch size, the rank of the clean matrix, and the filtering parameter h. The average ISNR is chosen as the performance measure. The performance of LRSNL with respect to the patch size and the rank of the clean matrix is shown in Fig. 3(a). Since the standard deviation is in the range of $[0.01, 0.03]$, h is simply set to the mean of this interval, 0.02. The performance is relatively stable when the rank is larger than 4, given the patch size. Hence, when we explore the effect of the filtering parameter h, we fix the patch size to 3×3 and the rank to 4. Fig. 3(b) plots the performance of LRSNL with respect h in this case. It shows that the value of h is slightly better to be 0.015 than 0.02. Hence, we set the value of h in (4) to 0.015.

For all methods, the 3-D cube is divided into small patches of size $3 \times 3 \times 223$, and each small patch is transformed into a 2-D matrix of size 9×223. In LRMR, the rank of the clean matrix is chosen from $\{2, 4, 6, 8\}$ and the cardinality of the sparse matrix is chosen from $\{30, 50, 70, 100\}$. The 16 combinations of the two parameters are evaluated and the best combination is chosen based on the average ISNR. The combination of rank 2 and cardinality 50 provides the best performance and is chosen for the experiments. In our LRSNL, the rank is set to 4. To reduce the computational cost, the searching area is set to a 21×21 square centred at the target pixel in the SNL step of LRSNL, by following the experiments in [11].

2) Results: Fig. 4 is the plot of ISNR versus all bands. It shows that our method can restore the noisy images better than do LRMR and SNL in almost all spectral bands. We note that the performances of LRSNL, LRMR, and SNL at band 140 are almost the same. This is mainly because only small Gaussian noise with a standard deviation of 0.016 has been added to the band. LRMR can perform well on bands with such small Gaussian noise, but compared with LRSNL and SNL it cannot remove large mixed noises in other bands. The restored spectral signatures of pixel $(136, 21)$ are shown in Fig. 5. Compared with the original spectral signature, LRSNL also provides the best results while LRMR performs the worst.

A synthetic image with only Gaussian noises and its restored images are shown in Fig. 6. The result from LRMR shows that large Gaussian noises cannot be effectively removed, edges are over-smoothed, and fine details are lost. Compared with LRMR, SNL, and LRSNL remove most of Gaussian noises and recover the fine details of the original image. The colours of the results of LRSNL are much closer to those of the original image compared with those of SNL, which indicates that LRSNL produces an image closer to the original image.

Fig. 7 presents the restoration results of an image with a mixture of Gaussian and impulse noises. LRSNL performs the best among the three methods. Blurred white dots in Fig. 7(c) indicate that LRMR performs badly on removing impulse noises. Gaussian noises also still exist in the LRMR results. LRSNL and SNL can remove most of the impulse noises, but SNL provides a much darker image than does LRSNL.

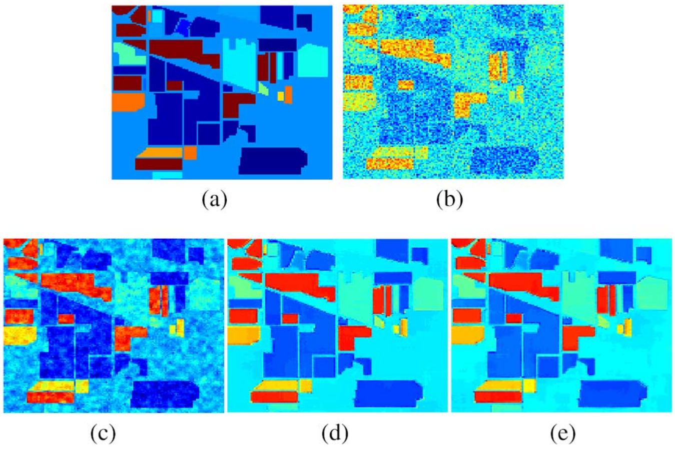


Fig. 6. Restoration of band 30 with Gaussian noises: (a) original; (b) noisy; (c) LRMR; (d) SNL; and (e) LRSNL.

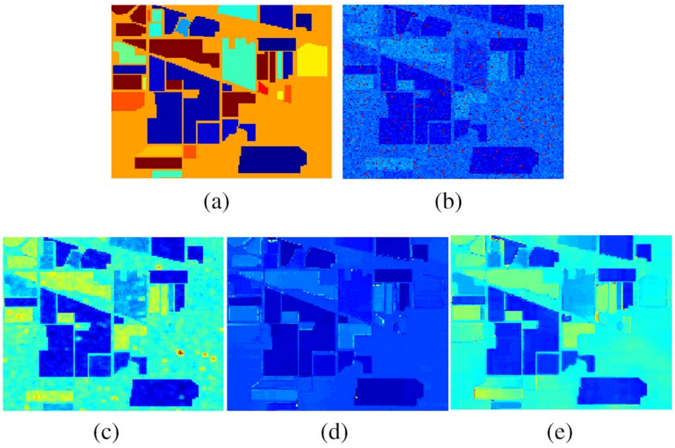


Fig. 7. Restoration of band 20 with a mixture of Gaussian and impulse noises: (a) original; (b) noisy; (c) LRMR; (d) SNL; and (e) LRSNL.

Compared with LRMR and SNL, LRSNL also shows superior performance against stripes and dead pixel lines. Fig. 8 displays the restoration results of an image with a mixture of Gaussian noises and dead pixel lines, in (c) of which the blurred black lines indicate that LRMR cannot effectively remove the dead lines. Some short lines in Fig. 8(d) indicate that SNL alone cannot effectively remove the dead pixel lines that appear over several bands. Moreover, the two dead pixel lines on the right-hand side are on the edges of land covers, and Fig. 8(e) shows that LRSNL still performs well on these dead pixel lines.

In summary, from Figs. 4 to 8, we can observe that the proposed LRSNL approach performs well in all the four situations. LRMR cannot effectively remove the mixed noises, and the fine structures within the images are also lost in its restored results. SNL performs better than LRMR but worse than LRSNL, as the patches are not precleaned. The colours of the restored results confirm that the restored values of SNL are worse than those of LRSNL. SNL also cannot effectively remove the dead pixel lines that appear successively in several bands. In contrast, LRSNL can effectively remove the mixed noises as well as preserve the fine spatial structures.

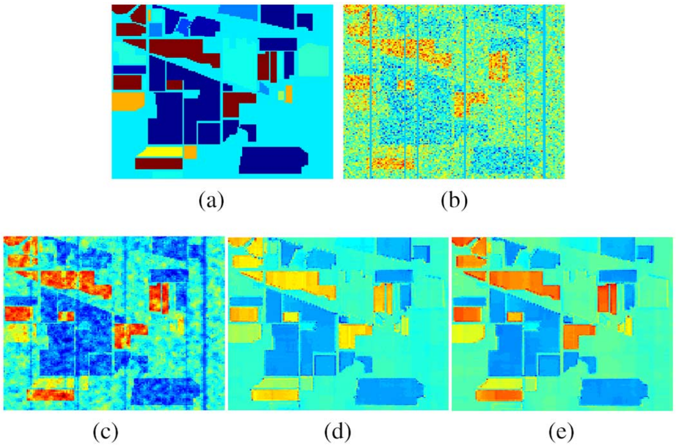


Fig. 8. Restoration of band 14 with a mixture of Gaussian noises and dead pixel lines: (a) original; (b) noisy; (c) LRMR; (d) SNL; and (e) LRSNL.

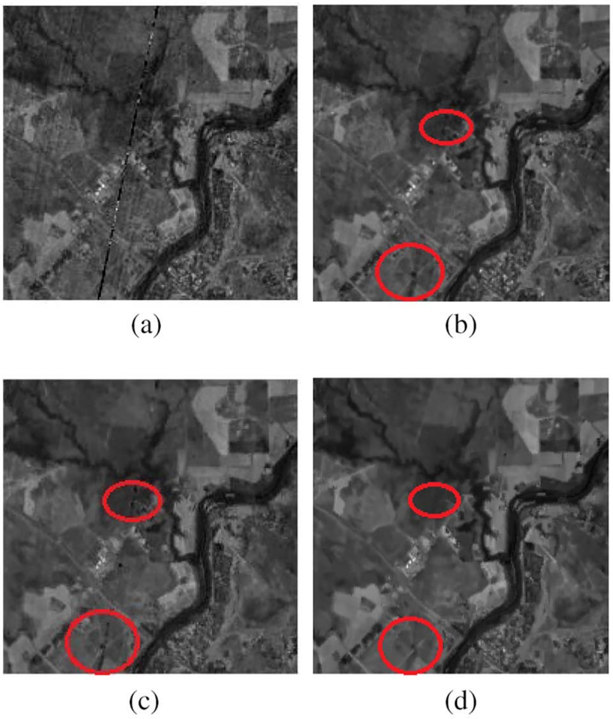


Fig. 9. Restoration of band 130 of an EO-1 Hyperion dataset: (a) original; (b) LRMR; (c) SNL; and (d) LRSNL.

B. Real-Data Experiments

An EO-1 Hyperion image dataset is used in our real-data experiments (http://eros.usgs.gov/find-data). The original dataset is of size $3371 \times 931 \times 242$. A subset of size $200 \times 200 \times 163$ is used here after the removal of water pollution bands. The pixel values of each band are normalized to $[0, 1]$ before experiments. For all methods, the dataset is first divided into patches of size $3 \times 3 \times 163$ and transformed into a 2-D matrix of size 9×163. As with the experiments in Section III-A, for LRMR, the rank of the clean image is set to 2 and the cardinality of the sparse matrix is set to 50. For LRSNL, the rank is set to 4 and the parameter h is set to 0.015. The searching area is set to a 21×21 square centred at the target pixel.

A large number of spectral bands of the original hyperspectral data cube are polluted by a mixture of dead pixel lines, stripes, and other noises. The restoration results of band 130 are shown in Fig. 9. LRMR can only remove part of dead pixel lines and stripes, as shown in Fig. 9(b). It also tends to over-smooth some edges. Although SNL preserves more fine structures compared with LRMR, the dead pixel line still can be spotted as shown in Fig. 9(c). Apparently, LRSNL performs the best among the three methods. It can remove almost all the noises and preserve the details as well, as shown in Fig. 9(d).

IV. Conclusion

In this paper, we have proposed LRSNL, a simple and effective restoration method for hyperspectral images. In LRSNL, the standard NL algorithm is extended to SNL to take advantage of both spectral and spatial information. Hence, a mixture of different types of noises can be removed simultaneously, and at the same time the fine details and local structures of the clean image can be preserved. For a better clustering of the patches in SNL, the LR property of the clean hyperspectral image is exploited in a precleaning step. The experiments have demonstrated the effectiveness of LRSNL and the importance of the precleaning step.

LRSNL treats all spectral bands the same and simply uses the average of all the bands to calculate similarities between patches. However, when spectral bands are of different importance, an adaptive weighting scheme is better to be developed.

Acknowledgment

The authors would like to thank A. Kazakeviciute for discussion on LRMR and the two anonymous reviewers for their constructive comments, in particular those having led to an improved literature review, model description, and experimental design for our work. They also acknowledge the receipt of research scholarships (RZ) from the Department of Statistical Science and the MAPS Faculty at UCL, a UCL Overseas Research Scholarship (MD), and a China Scholarship Council award (MD).

References

[1] Y. Qian and M. Ye, "Hyperspectral imagery restoration using nonlocal spectral–spatial structured sparse representation with noise estimation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 499–515, Apr. 2013.

[2] M. Golbabaee and P. Vandergheynst, "Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2012, pp. 2741–2744.

[3] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.

[4] A. A. Green, M. Berman, P. Switzer, and M. D. Craig, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 1, pp. 65–74, Jan. 1988.

[5] N. Acito, M. Diani, and G. Corsini, "Subspace-based striping noise reduction in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 4, pp. 1325–1342, Apr. 2011.

[6] Q. Wang, L. Zhang, Q. Tong, and F. Zhang, "Hyperspectral imagery denoising based on oblique subspace projection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2468–2480, Jun. 2014.

[7] J. G. Liu and G. L. K. Morgan, "FFT selective and adaptive filtering for removal of systematic noise in ETM+ imageodesy images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3716–3724, Dec. 2006.

[8] R. Pande-Chhetri and A. Abd-Elrahman, "De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 5, pp. 620–636, 2011.

[9] A. Duijster, P. Scheunders, and S. De Backer, "Wavelet-based EM algorithm for multispectral-image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3892–3898, Nov. 2009.

[10] B. Rasti, J. Sveinsson, M. Ulfarsson, and J. Benediktsson, "Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2458–2467, Jun. 2014.

[11] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

[12] Q. Li, H. Li, Z. Lu, Q. Lu, and W. Li, "Denoising of hyperspectral images employing two-phase matrix decomposition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 9, pp. 3742–3754, 2014.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, 2009, pp. 2272–2279.

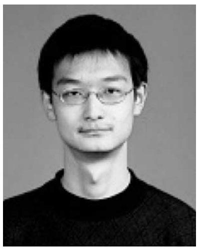
[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[15] X. Zhou and W. Yu, "Low-rank modeling and its applications in medical image analysis," in *Proc. SPIE Defense Secur. Sens.*, 2013, p. 87500V.

[16] T. Zhou and D. Tao, "GoDec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 33–40.

Rui Zhu received the B.S. degree in engineering from Xiamen University, Xiamen, China, in 2012, and the M.Sc. degree in statistics from the University College London, London, U.K., in 2013. She is currently pursuing the Ph.D. degree at the Department of statistical science, University College London.

Her research interests include spectral data analysis, hyperspectral image analysis, and metric learning.



Mingzhi Dong received the B.Eng. degree in automation, and the M.Eng. degree in signal and information processing from Beijing University of Posts and Telecommunications, Beijing, China, in 2010 and 2013, respectively.

He is currently a Research Student with the Department of Statistical Science, University College London, London, U.K. His research interests include data analysis, pattern recognition, and machine learning.

Jing-Hao Xue received the Dr. Eng. degree in signal and information processing from Tsinghua University, Beijing, China, in 1998, and the Ph.D. degree in statistics from the University of Glasgow, Glasgow, U.K., in 2008.

Since 2008, he has been with the Department of Statistical Science, University College London, London, U.K., as a Lecturer and Senior Lecturer. His research interests include statistical classification, high-dimensional data analysis, computer vision, and pattern recognition.

Appendix B

MvSSIM: A Quality Assessment Index for Hyperspectral Images

MvSSIM: A Quality Assessment Index for Hyperspectral Images

Rui Zhu[a], Fei Zhou[b], Jing-Hao Xue[a,*]

[a] *Department of Statistical Science, University College London, London WC1E 6BT, UK*
[b] *The Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China*

Abstract

Quality assessment indexes play a fundamental role in the analysis of hyperspectral images (HSI) cubes. To assess the quality of an HSI cube, the structural similarity (SSIM) index has been widely applied in a band-by-band manner, as SSIM was originally designed for 2D images, and then the mean SSIM (MeanSSIM) index over all bands is adopted. MeanSSIM fails to accommodate the spectral structure which is a unique characteristic of HSI. Hence in this paper, we propose a new and simple multivariate SSIM (MvSSIM) index for HSI, by treating the pixel spectrum as a multivariate random vector. MvSSIM maintains SSIM's ability to assess the spatial structural similarity via correlation between two images of the same band; and adds an ability to assess the spectral structural similarity via covariance among different bands. MvSSIM is well founded on multivariate statistics and can be easily implemented through simple sample statistics involving mean vectors, covariance matrices and cross-covariance matrices. Experiments show

[*] Corresponding author. Tel.: +44-20-7679-1863; Fax: +44-20-3108-3105
Email addresses: `r.zhu.12@ucl.ac.uk` (Rui Zhu), `flyingzhou@sz.tsinghua.edu.cn` (Fei Zhou), `jinghao.xue@ucl.ac.uk` (Jing-Hao Xue)

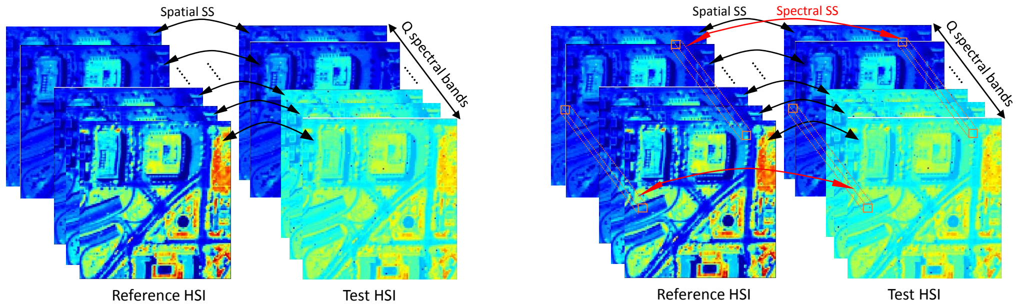
that MvSSIM is a proper quality assessment index for distorted HSIs with different kinds of degradations.

Keywords: Hyperspectral images, quality assessment, structural similarity (SSIM), spectral structure, spatial structure.

1. Introduction

Hyperspectral images (HSI) are captured on 100s of narrow spectral bands ranging from 400 to 2400 nm, represented as a 3D data cube containing both the spatial structure in two dimensions and the spectral structure in the other dimension. Quality assessment plays a fundamental role in HSI analysis, especially in image restoration. Image restoration aims to recover clean HSI and thus facilitate further analysis such as classification, target detection and unmixing. A good quality assessment index can identify well-cleaned HSI.

The structural similarity (SSIM) index has been widely used in the quality assessment of HSI [1–6]. SSIM was originally designed for traditional 2D greyscale images to assess the image quality resembling human perception [7–10]. SSIM can evaluate the similarity in the spatial structure between two images (a reference image and a test image). Recently, many extensions of SSIM for 2D images have been proposed, such as multi-scale SSIM [11], complex wavelet SSIM [12], information content weighting SSIM [13] and intra-and-inter patch similarity [14], among others. As with these works, in this paper we focus on the full reference assessment, i.e. a reference image (an HSI cube in our case) is provided.



(a) MeanSSIM.

(b) MvSSIM.

Figure 1: Illustration of MeanSSIM and MvSSIM (‘SS’ for structural similarity).

In the literature on using SSIM for HSI, usually a band-by-band manner is adopted for the 3D cube. The SSIM index for the image of each spectral band is calculated and then the mean of all these SSIM indexes (MeanSSIM) is taken as the quality measure of the whole HSI cube, as illustrated in Figure 1a. This simple strategy can compare the within-band spatial structure between each pair of images for the same band in the reference HSI and the test HSI. However, the similarity in the cross-band spectral structure has been neglected, although such information is rich, unique and crucial in HSI. It is well known that both spatial and spectral structures are of great importance in the analysis of HSI and omitting the spectral structure is undesirable. Alparone et al. [15] and Garzelli and Nencini [16], extend SSIM to HSI by representing the pixel spectrum as a hypercomplex number. However, restricted by the properties of hypercomplex numbers, their index needs a recursive procedure to compute, making it not as popular as MeanSSIM in HSI restoration and denoising.

In this context, we propose in this paper a new and simple quality assessment index for HSI, termed multivariate SSIM (MvSSIM). In a 2D image a pixel is treated as a univariate random variable by SSIM; in contrast, in an HSI cube a pixel is in nature a multivariate random vector. By replacing the univariate sampling statistics in SSIM with their multivariate versions, MvSSIM generalises SSIM to HSI. Compared with MeanSSIM, MvSSIM can assess both the within-band spatial structural similarity, between images of the same band, and the cross-band spectral structural similarity, between spectra of the same pixel, as illustrated in Figure 1b between a reference cube and a test cube. MvSSIM is well founded on multivariate statistics and can be easily implemented through simple multivariate sample statistics involving mean vectors, covariance matrices and cross-covariance matrices. Experiments show that MvSSIM is a proper quality assessment index for distorted HSIs with different kinds of noises.

2. MvSSIM for hyperspectral images

2.1. SSIM

SSIM is a quality assessment index originally designed for 2D greyscale images. Suppose we have two images $\boldsymbol{x}$ and $\boldsymbol{y}$, both containing $N = a \times b$ pixels: $\boldsymbol{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{y} = [y_1, \ldots, y_N]^T \in \mathbb{R}^{N\times 1}$, aligned with each other. In SSIM, the N pixels of a 2D image are treated as N realisations of a univariate random variable: x_i and y_i $(i = 1, \ldots, N)$ are the realisations of random variables x and y, respectively.

SSIM consists of three comparisons between $\boldsymbol{x}$ and $\boldsymbol{y}$: the similarity of luminance, $l(\boldsymbol{x}, \boldsymbol{y})$; the similarity of contrast, $c(\boldsymbol{x}, \boldsymbol{y})$; and the similarity of structure, $s(\boldsymbol{x}, \boldsymbol{y})$. It is defined as the product of the powers of these three similarities:

$$\mathrm{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = [l(\boldsymbol{x}, \boldsymbol{y})]^{\alpha} \times [c(\boldsymbol{x}, \boldsymbol{y})]^{\beta} \times [s(\boldsymbol{x}, \boldsymbol{y})]^{\gamma}, \tag{1}$$

where α, β and γ are three positive exponents adjusting the relative importance of the similarities and often all set to 1.

The three similarities are calculated by using the sample statistics of x and y. First, the similarity of luminance $l(\boldsymbol{x}, \boldsymbol{y})$ is obtained by comparing the sample means $\bar{x}$ and $\bar{y}$:

$$l(\boldsymbol{x}, \boldsymbol{y}) = \frac{2\bar{x}\bar{y} + C_1}{\bar{x}^2 + \bar{y}^2 + C_1}, \tag{2}$$

where $\bar{x} = \frac{1}{N}\sum\limits_{i=1}^{N} x_i$ and $\bar{y} = \frac{1}{N}\sum\limits_{i=1}^{N} y_i$, and C_1 is a constant that controls the stability of the fraction when $\bar{x}^2 + \bar{y}^2$ is close to zero. Constants C_2 and C_3 in the other two similarities play the same role as C_1.

Second, the similarity of contrast $c(\boldsymbol{x}, \boldsymbol{y})$ is obtained by comparing the sample standard deviations s_x and s_y:

$$c(\boldsymbol{x}, \boldsymbol{y}) = \frac{2s_x s_y + C_2}{s_x^2 + s_y^2 + C_2}, \tag{3}$$

where $s_x^2 = \frac{1}{N-1}\sum\limits_{i=1}^{N}(x_i - \bar{x})^2$ and $s_y^2 = \frac{1}{N-1}\sum\limits_{i=1}^{N}(y_i - \bar{y})^2$ are the sample variances.

Third, the similarity of structure $s(\boldsymbol{x}, \boldsymbol{y})$ is calculated as the sample correlation coefficient of x and y:

$$s(\boldsymbol{x}, \boldsymbol{y}) = \frac{s_{xy}^2 + C_3}{s_x s_y + C_3}, \tag{4}$$

where $s_{xy}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$ is the sample cross-variance. The sample correlation coefficient measures the linear dependency between x and y, indicating the similarity between two within-image spatial structures of the two images, which were vectorised into a pair of two N-element vectors. Thus $s(\boldsymbol{x}, \boldsymbol{y})$ is of great important in SSIM for assessing the spatial structural similarity of two images.

SSIM possesses the following three good properties as a similarity index. First, SSIM is symmetric, i.e. $\mathrm{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{SSIM}(\boldsymbol{y}, \boldsymbol{x})$. Second, the value of SSIM is bounded, i.e. $\mathrm{SSIM}(\boldsymbol{x}, \boldsymbol{y}) \in [-1, 1]$. Third, SSIM has a unique maximum, i.e. $\mathrm{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = 1$ if and only if $\boldsymbol{x} = \boldsymbol{y}$.

2.2. MeanSSIM

When SSIM is used in the quality assessment of HSI, it is commonly applied in a band-by-band manner. That is, an SSIM index is obtained for a pair of images of the same band, and then the mean index over bands is used as the quality measure of the test HSI cube against the reference cube, as illustrated in Figure 1a. We call this measure the mean SSIM (MeanSSIM) index.

Suppose we have two HSI cubes, $\boldsymbol{X}_H \in \mathbb{R}^{a \times b \times Q}$ and $\boldsymbol{Y}_H \in \mathbb{R}^{a \times b \times Q}$, where a and b represent the numbers of pixels in height and width, and Q is the number of spectral bands. $\boldsymbol{X}_H$ and $\boldsymbol{Y}_H$ can be rearranged as 2D matrices $\boldsymbol{X} = [\boldsymbol{x}_1^c, \boldsymbol{x}_2^c, \ldots, \boldsymbol{x}_Q^c] \in \mathbb{R}^{N \times Q}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1^c, \boldsymbol{y}_2^c, \ldots, \boldsymbol{y}_Q^c] \in \mathbb{R}^{N \times Q}$, where $N = a \times b$ denotes the total number of pixels and $\boldsymbol{x}_q^c \in \mathbb{R}^{N \times 1}$ and $\boldsymbol{y}_q^c \in \mathbb{R}^{N \times 1}$ represent the image vectors of the qth spectral band of $\boldsymbol{X}_H$ and

$\boldsymbol{Y}_H$, respectively. The MeanSSIM index is calculated as

$$\text{MeanSSIM} = \frac{1}{Q}\sum_{q=1}^{Q} \text{SSIM}(\boldsymbol{x}_q^c, \boldsymbol{y}_q^c). \tag{5}$$

MeanSSIM can explore the similarity in spatial structure of each pair of band images. However, due to its band-by-band manner, it fails to adequately explore the cross-band spectral structure in HSI, while the spectrum of each pixel, i.e. each row of $\boldsymbol{X}$ or $\boldsymbol{Y}$, contains crucial information like its chemical components. Thus, in addition to assessing the within-band spatial structural similarity between two images of the same band, assessing the cross-band spectral structural similarity between two spectra at the same spatial position should also be considered in the quality assessment of HSI.

2.3. MvSSIM

Since an HSI cube contains both spatial structure and spectral structure, its quality assessment should contain assessments for both structures. Hence in this paper, we propose multivariate SSIM (MvSSIM) for the quality assessment of HSI, generalising SSIM via multivariate sample statistics.

In MvSSIM, the spectrum of each pixel of an HSI cube is treated as a realisation of a Q-dimensional random vector. To be more specific, we rewrite $\boldsymbol{X} \in \mathbb{R}^{N\times p}$ and $\boldsymbol{Y} \in \mathbb{R}^{N\times p}$ as $\boldsymbol{X} = [\boldsymbol{x}_1^r, \boldsymbol{x}_2^r, \ldots, \boldsymbol{x}_N^r]^T$ and $\boldsymbol{Y} = [\boldsymbol{y}_1^r, \boldsymbol{y}_2^r, \ldots, \boldsymbol{y}_N^r]^T$, where $\boldsymbol{x}_n^r \in \mathbb{R}^{Q\times 1}$ and $\boldsymbol{y}_n^r \in \mathbb{R}^{Q\times 1}$ represent the spectra of the nth pixel of $\boldsymbol{X}_H$ and $\boldsymbol{Y}_H$, respectively. Here $\boldsymbol{x}_n^r$ and $\boldsymbol{y}_n^r$ are considered as the realisations of Q-dimensional random vectors $X \in \mathbb{R}^{Q\times 1}$ and $Y \in \mathbb{R}^{Q\times 1}$, respectively.

As an extension of SSIM, MvSSIM also consists of three similarity measurements between $\boldsymbol{X}$ and $\boldsymbol{Y}$, i.e. $l(\boldsymbol{X},\boldsymbol{Y})$, $c(\boldsymbol{X},\boldsymbol{Y})$ and $s(\boldsymbol{X},\boldsymbol{Y})$. These three similarities are defined on the following multivariate sample statistics of X and Y:

i) the sample means,

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n^r \in \mathbb{R}^{Q\times 1},\ \bar{Y} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}_n^r \in \mathbb{R}^{Q\times 1}; \tag{6}$$

ii) the sample covariance matrices,

$$\boldsymbol{\Sigma}_X = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n^r - \bar{X})(\boldsymbol{x}_n^r - \bar{X})^T \in \mathbb{R}^{Q\times Q}\ , \tag{7}$$

$$\boldsymbol{\Sigma}_Y = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{y}_n^r - \bar{Y})(\boldsymbol{y}_n^r - \bar{Y})^T \in \mathbb{R}^{Q\times Q}; \tag{8}$$

and iii) the sample cross-covariance matrix,

$$\boldsymbol{\Sigma}_{XY} = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n^r - \bar{X})(\boldsymbol{y}_n^r - \bar{Y})^T \in \mathbb{R}^{Q\times Q}. \tag{9}$$

Different from the univariate sample statistics in SSIM, the sample statistics in MvSSIM are vectors or matrices, rather than scalars. Thus the comparisons between scalars in SSIM should be extended to comparisons between vectors or matrices in MvSSIM. The extensions from $l(\boldsymbol{x},\boldsymbol{y})$, $c(\boldsymbol{x},\boldsymbol{y})$ and $s(\boldsymbol{x},\boldsymbol{y})$ to $l(\boldsymbol{X},\boldsymbol{Y})$, $c(\boldsymbol{X},\boldsymbol{Y})$ and $s(\boldsymbol{X},\boldsymbol{Y})$ are described as follows.

2.3.1. From $l(\boldsymbol{x},\boldsymbol{y})$ to $l(\boldsymbol{X},\boldsymbol{Y})$

As with $l(\boldsymbol{x},\boldsymbol{y})$, $l(\boldsymbol{X},\boldsymbol{Y})$ measures the luminance similarity between images by comparing the sample mean vectors, $\bar{X}$ and $\bar{Y}$. Because $l(\boldsymbol{X},\boldsymbol{Y})$ compares the luminance similarity, the spectral structure is not included in this term and the inner products of vectors are used to make the numerator and denominator scalars. We define

$$l(\boldsymbol{X},\boldsymbol{Y}) = \frac{2\langle\bar{X},\bar{Y}\rangle + C_1}{\langle\bar{X},\bar{X}\rangle + \langle\bar{Y},\bar{Y}\rangle + C_1} = \frac{2\sum\limits_{q=1}^{Q}\bar{x}_q\bar{y}_q + C_1}{\sum\limits_{q=1}^{Q}(\bar{x}_q^2 + \bar{y}_q^2) + C_1}, \tag{10}$$

where $\langle\ ,\ \rangle$ denotes the inner product of two vectors, and $\bar{x}_q$ and $\bar{y}_q$ are the qth entries of $\bar{X}$ and $\bar{Y}$, respectively.

It is easy to show that $l(\boldsymbol{X},\boldsymbol{Y}) \in [0,1]$ and $l(\boldsymbol{X},\boldsymbol{Y}) = 1$ when $\boldsymbol{X} = \boldsymbol{Y}$. If $Q = 1$, i.e. the HSI becomes a 2-D image, (10) degenerates into (2) of SSIM.

2.3.2. From $c(\boldsymbol{x},\boldsymbol{y})$ to $c(\boldsymbol{X},\boldsymbol{Y})$

Similar to $c(\boldsymbol{x},\boldsymbol{y})$, $c(\boldsymbol{X},\boldsymbol{Y})$ compares the similarity between sample covariance matrices $\boldsymbol{\Sigma}_X$ and $\boldsymbol{\Sigma}_Y$. A sample covariance matrix (e.g. $\boldsymbol{\Sigma}_X$) contains the variances within individual bands (of $\boldsymbol{X}$) in its diagonal entries, and the covariances between different spectral bands (of $\boldsymbol{X}$) in its off-diagonal entries. Hence when we compare $\boldsymbol{X}$ and $\boldsymbol{Y}$ through $\boldsymbol{\Sigma}_X$ and $\boldsymbol{\Sigma}_Y$, we can achieve two comparisons simultaneously: comparing the contrasts of two images of the same band via the two standard deviations of this band, and comparing the contrasts of two spectra of the same spatial position via the covariances between different bands.

To make use of both the spatial and spectral information and to make the numerator and the denominator scalars, a natural choice is to use the nuclear norm to summarise the sample covariance matrix. Hence we define $c(\boldsymbol{X}, \boldsymbol{Y})$ as

$$c(\boldsymbol{X}, \boldsymbol{Y}) = \frac{2||\boldsymbol{\Sigma}_X||_*^{\frac{1}{2}}||\boldsymbol{\Sigma}_Y||_*^{\frac{1}{2}} + C_2}{||\boldsymbol{\Sigma}_X||_* + ||\boldsymbol{\Sigma}_Y||_* + C_2} = \frac{2\sqrt{\lambda^s}\sqrt{d^s} + C_2}{\lambda^s + d^s + C_2}, \tag{11}$$

where $|| \; ||_*$ is the nuclear norm, $\lambda^s = \sum\limits_{q=1}^{Q} \lambda_q$, $d^s = \sum\limits_{q=1}^{Q} d_q$, and λ_q and d_q are the singular values of $\boldsymbol{\Sigma}_X$ and $\boldsymbol{\Sigma}_Y$, respectively.

The similarity $c(\boldsymbol{X}, \boldsymbol{Y})$ can take values in $[0, 1]$, and $c(\boldsymbol{X}, \boldsymbol{Y}) = 1$ when $\boldsymbol{X} = \boldsymbol{Y}$. If $Q = 1$, we treat the spectral norm of a scalar as itself and (11) is equivalent to (3) of SSIM.

2.3.3. From $s(\boldsymbol{x}, \boldsymbol{y})$ to $s(\boldsymbol{X}, \boldsymbol{Y})$

The term $s(\boldsymbol{x}, \boldsymbol{y})$ measures the spatial structural similarity between two images and is vital for SSIM resembling human perception. Preserving this good property of SSIM, we also adopt the correlation coefficient for MvSSIM. We define $s(\boldsymbol{X}, \boldsymbol{Y})$ as

$$\begin{aligned} s(\boldsymbol{X}, \boldsymbol{Y}) &= \frac{1}{Q} \operatorname{trace}((\boldsymbol{\Sigma}_{\boldsymbol{XY}} + C_3 \boldsymbol{I}_Q)(\boldsymbol{\Gamma}_X^{\frac{1}{2}} \boldsymbol{\Gamma}_Y^{\frac{1}{2}} + C_3 \boldsymbol{I}_Q)^{-1}) \\ &= \frac{1}{Q} \sum_{q=1}^{Q} \frac{\sigma_{XYq}^2 + C_3}{\sigma_{Xq}\sigma_{Yq} + C_3}, \end{aligned} \tag{12}$$

where $\boldsymbol{\Gamma}_X$ and $\boldsymbol{\Gamma}_Y$ are diagonal matrices composed of the diagonal elements of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{Y}}$, respectively; and σ_{XYq}^2, σ_{Xq}^2 and σ_{Yq}^2 are the qth diagonal entry of $\boldsymbol{\Sigma}_{\boldsymbol{X}}$, $\boldsymbol{\Sigma}_{\boldsymbol{Y}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{XY}}$, respectively. It is obvious that $s(\boldsymbol{X}, \boldsymbol{Y})$ is the mean of correlation coefficients of all spectral bands.

The similarity $s(\boldsymbol{X}, \boldsymbol{Y}) \in [-1, 1]$, and $s(\boldsymbol{X}, \boldsymbol{Y}) = 1$ when $\boldsymbol{X} = \boldsymbol{Y}$. If $Q = 1$, (12) degenerates into (4) of SSIM.

2.3.4. MvSSIM

Combing the three similarity measurements defined above, the MvSSIM index of $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be written in a similar formulation to SSIM:

$$\text{MvSSIM}(\boldsymbol{X}, \boldsymbol{Y}) = [l(\boldsymbol{X}, \boldsymbol{Y})]^{\alpha} \times [c(\boldsymbol{X}, \boldsymbol{Y})]^{\beta} \times [s(\boldsymbol{X}, \boldsymbol{Y})]^{\gamma}, \tag{13}$$

where as with SSIM α, β and γ are three positive exponents that adjust the relative importance of the components.

Among these three terms, $l(\boldsymbol{X}, \boldsymbol{Y})$ and $s(\boldsymbol{X}, \boldsymbol{Y})$ measure the similarity between band images in luminance and spatial structure, while $c(\boldsymbol{X}, \boldsymbol{Y})$ measures the similarity between both band images and pixel spectra. Thus in MvSSIM, both the within-band spatial structural similarity and the cross-band spectral structural similarity are assessed.

Moreover, comparing (1)-(4) with (10)-(13), we can find that MvSSIM is a natural generalisation of SSIM, and thus it can be readily embedded into other state-of-the-art SSIM-based quality assessment indexes such as [11–14].

3. Experiments

Besides MeanSSIM, MvSSIM is also compared with three other SSIM-based quality assessment indexes in literature, namely Q_{λ}, Q_m [17] and $Q2^n$ [16].

The index Q_λ measures the minimum SSIM between the pair of spectra of the same pixel among all pixels; Q_m is the product of Q_λ and the minimum SSIM between the pair of images of the same band among all bands; and $Q2^n$ is an extension of SSIM by expressing the spectrum as a hypercomplex number.

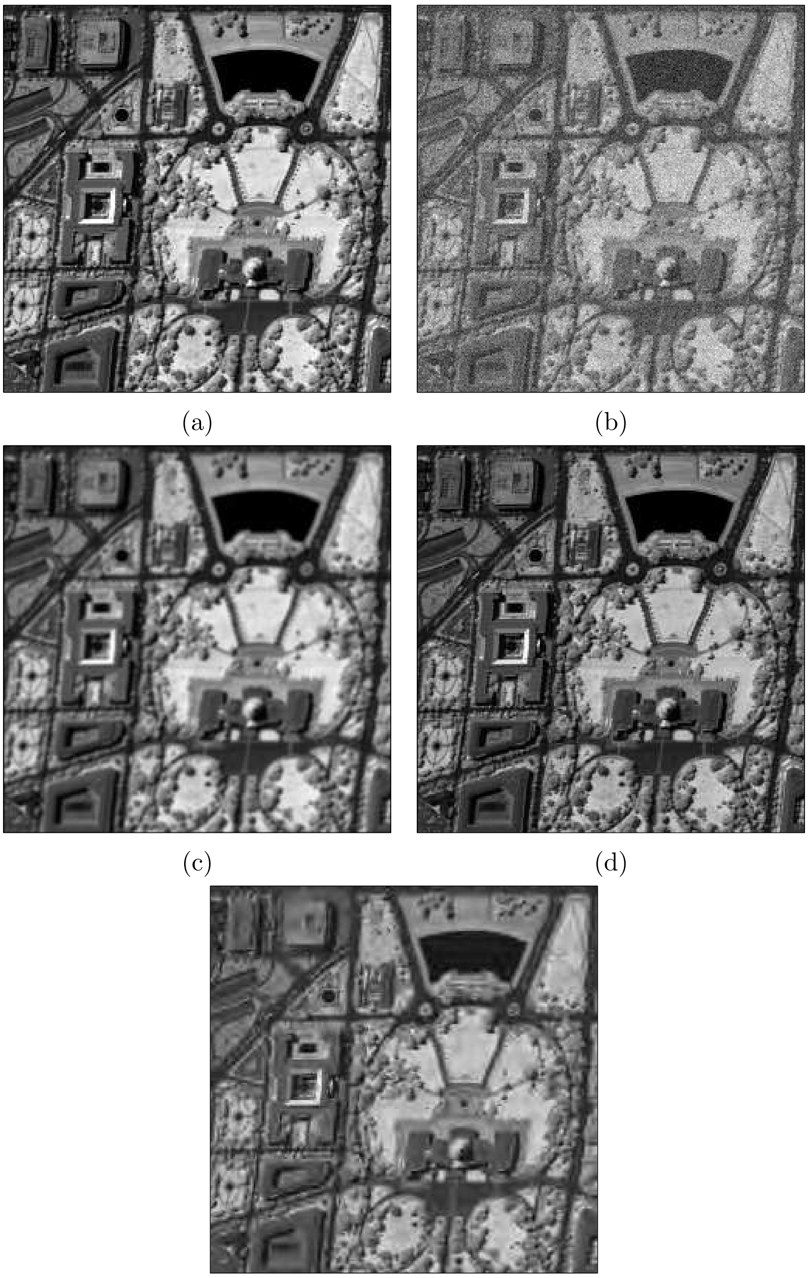
The five quality assessment indexes could be categorised into the following three groups: 1) Q_λ, which measures spectral similarities between spectra of the same pixel; 2) MeanSSIM, which measures spatial similarities between images of the same band; and 3) Q_m, $Q2^n$ and MvSSIM, which measure both spectral and spatial similarities.

3.1. Dataset

The Washington DC dataset is used for the synthetic experiments. The dataset is of size $250 \times 250 \times 191$, where 250×250 is the size of the image of each spectral band and 191 is the number of bands. The original HSI cube serves as the reference cube while its noisy version acts as a test cube.

3.2. Experiment settings

MeanSSIM is computed using the MATLAB function 'ssim' with the default setting: window size is 11, $C_1 = 0.01$ and $C_2 = 0.03$. For MvSSIM, a patch of size $5 \times 5 \times 191$ moves from pixel to pixel, the index of each patch is calculated, and then the mean index of all the patches is taken as the index of the whole HSI. We set constants C_i of MvSSIM to 0 and exponents α, β and γ to 1 for simplicity. The index $Q2^n$ is calculated by using the pansharpening toolbox of [18]. The block size is set to 32 and the block shift size is set to 32, as suggested in [16].



(a)

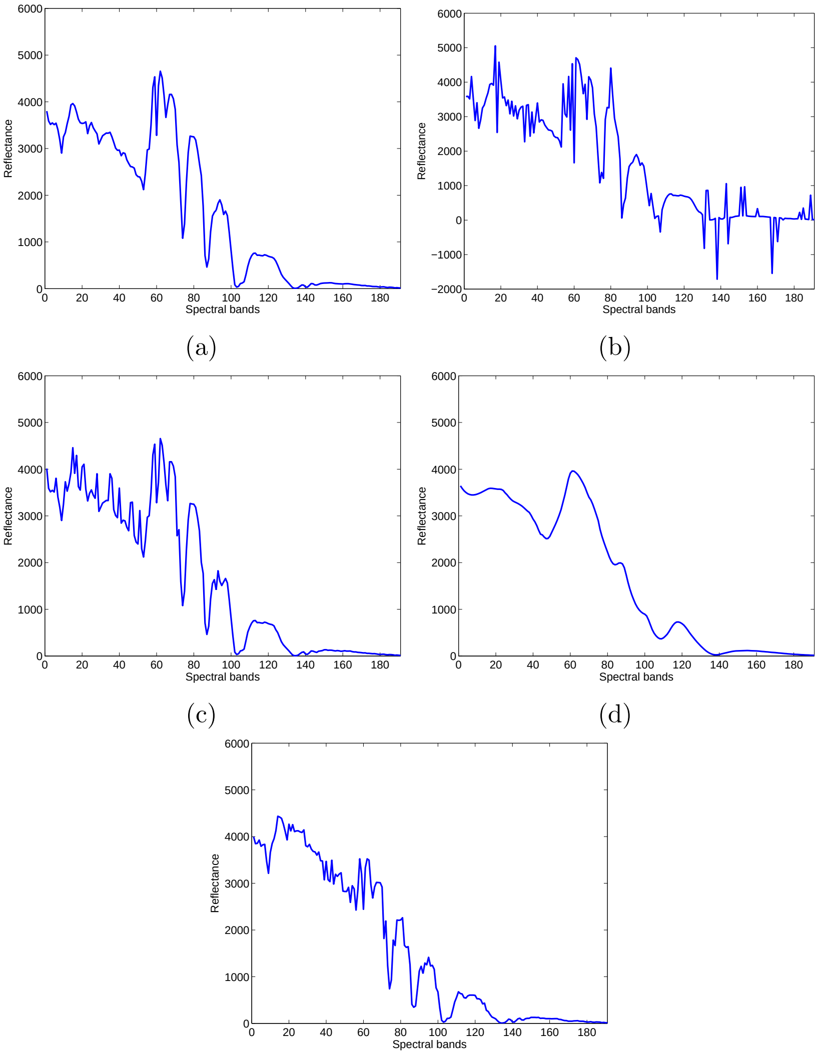
(b)

(c)

(d)

(e)

Figure 2: The reference image and noisy images of band 80. (a) Reference. (b) Gaussian white noise (variance 60). (c) Gaussian smoothing noise (standard deviation 1). (d) Savitzky-Golay smoothing noises (frame size 11). (e) JPEG2000 compression (compression ratio 30).



(a)

(b)

(c)

(d)

(e)

Figure 3: The reference spectrum and noisy spectra of the pixel at position (50, 50). (a) Reference. (b) Gaussian white noise (variance 60). (c) Gaussian smoothing noise (standard deviation 1). (d) Savitzky-Golay smoothing noises (frame size 11). (e) JPEG2000 compression (compression ratio 30).

Following the experiments in [17], four typical degradations are applied to the HSI to evaluate the quality assessment indexes: Gaussian white additive noise, spatial smoothing, spectral smoothing and lossy compression. The index values are calculated for different levels of degradations.

First, Gaussian white additive noises are added to 50 randomly-selected bands of the spectra. We test 10 different variances: from 10 to 100 with a step of 10, i.e. 10 different noisy HSIs are created with different variances.

Second, Gaussian smoothing filters are applied to 50 randomly-selected bands to create spatially blurred band images, i.e. in the spatial dimensions of the HSI. Eight different standard deviations of the Gaussian smoothing kernels are tested: 0.1, 0.5, 1, 5, 10, 50, 100 and 500, i.e. eight different noisy HSIs are created with different standard deviations.

Third, Savitzky-Golay smoothing filter is applied to the spectra of all pixels to create smooth spectra, i.e. in the spectral dimension of the HSI. We test eight different frame sizes: 5, 11, 31, 71, 91, 131, 171 and 191, i.e. eight different noisy HSIs are created with different frame sizes..

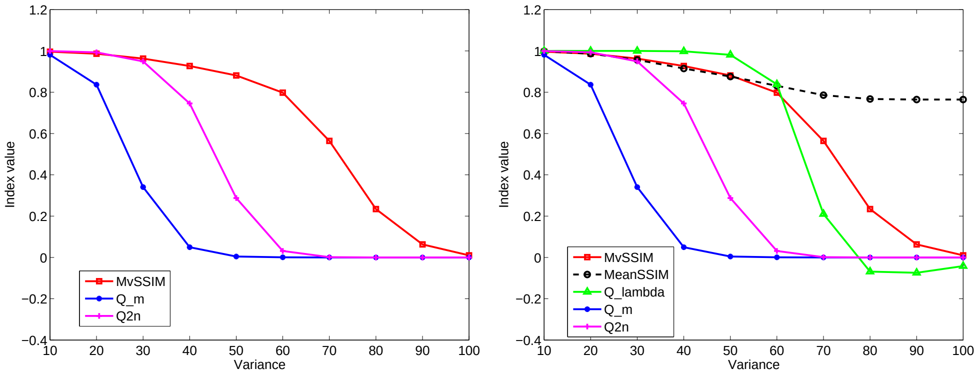
Fourth, JPEG2000 compression is applied to the HSI in a band-by-band way. We test five different compression ratios: from 10 to 50 with a step of 10, i.e. five different noisy HSIs are created with different compression ratios.

The reference image and noisy images of band 80 and the reference spectrum and noisy spectra of pixel (50, 50) are shown in Figure 2 and Figure 3.

3.3. Results

3.3.1. Gaussian white additive noise

Figure 4 shows the assessments for the HSIs contaminated by the Gaussian white additive noises of different variances, which represent different de-



(a) (b)

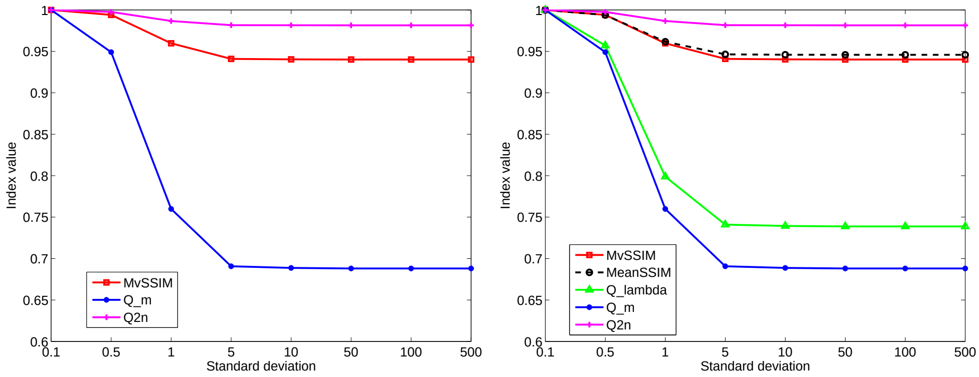
Figure 4: Assessments for the Gaussian white additive noise contaminated HSIs.

grees of contamination. The performances of the three indexes that measure both spectral and spatial similarities are shown in Figure 4a. It is obvious that Q_m is the most sensitive to the Gaussian white additive noise, $Q2^n$ is less sensitive, and MvSSIM is the least sensitive. However, sensitivity is not the only criterion to evaluate the performances of the indexes. The changes in the spatial structure and the spectral structure should also be considered when carrying out such evaluation.

We use MeanSSIM as a measurement for the spatial structural change and Q_λ as a measurement for the spectral structural change, and plot the performances of these two indexes in Figure 4b. In the plot, the value of Q_λ is high when the variance is less than 60 and drops fast when the variance becomes large; this indicates that the spectral structure changes little when the white noise is light but can change dramatically when the white noise is heavy. In the meantime, the figure shows that the value of MeanSSIM is relatively stable; this indicates that the spatial structure does not change much with the variance of white noise. This is because MeanSSIM averages out white noise over bands that the low similarities between contaminated band images are compensated by high similarities between other band images.

Considering the above behaviours of MeanSSIM and Q_λ, we prefer MvSSIM in the Gaussian white noise case even though it is the least sensitive index in Figure 4a. As shown in Figure 4b, it is clear that the values of Q_m and $Q2^n$ are close to zero even when the values of Q_λ are still close to one; this indicates that Q_m and $Q2^n$ fail to consider the high spectral structural similarity in this case and are over-sensitive to the Gaussian white noise. In contrast, MvSSIM provides large values when the values of Q_λ are large. Also, compared with Q_λ, MvSSIM is more desired because it also reflects the spatial structural similarity, making it between MeanSSIM and Q_λ in the case of Gaussian white noise.

3.3.2. Gaussian smoothing noise



(a) (b)

Figure 5: Assessments for the Gaussian smoothing noise contaminated HSIs.

Figure 5a shows the assessments for the HSIs contaminated by the Gaussian smoothing noise: Q_m is the most sensitive to the Gaussian smoothing noise, MvSSIM is less sensitive, and $Q2^n$ is the least sensitive.

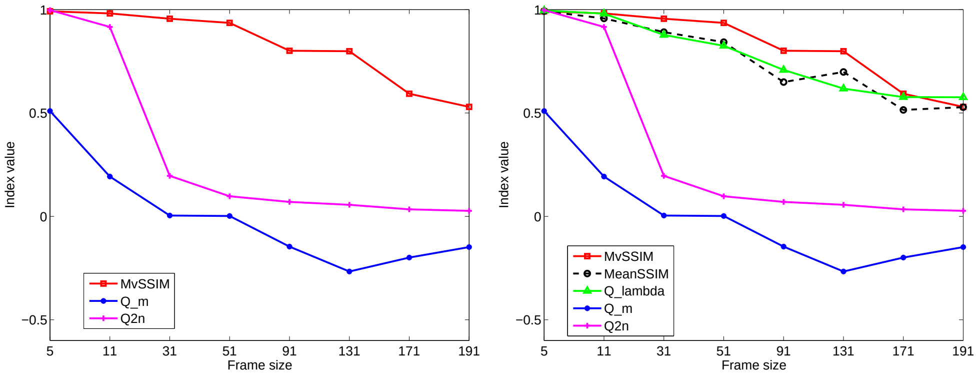
Similarly to the case of Gaussian white noise, we use MeanSSIM to consider the spatial structural similarity and use Q_λ to consider the spectral structural similarity, as plotted in in Figure 5b to evaluate the relative performances of MvSSIM, Q_m and $Q2^n$. The value of Q_λ drops quickly when the standard deviation of the Gaussian smooth noise is larger than one, while the value of MeanSSIM is less sensitive to the Gaussian smoothing noise compared with that of Q_λ.

When Q_λ largely decreases due to the noise, $Q2^n$ remains relatively stable; this indicates that $Q2^n$ fails to respond well to the decrease in the spectral structural similarity introduced by the Gaussian smoothing noise. In contrast, Q_m reflects well the changes in the spectral structural similarity. However, Q_m fails to consider the strong spatial structural similarity as indicated by the big values of MeanSSIM. Compared with $Q2^n$ and Q_m, MvSSIM is a more desired candidate to assess the Gaussian smoothing noise contaminated HSIs. It is between MeanSSIM and Q_λ, demonstrating a reasonable compromise between the spatial structural similarity and the spectral structural similarity.

3.3.3. Savitzky-Golay smoothing noise

Figure 6a shows the assessments for the HSIs contaminated by the Savitzky-Golay smoothing noise: Q_m is the most sensitive to the Savitzky-Golay smoothing noise, $Q2^n$ is less sensitive, and MvSSIM is the least sensitive.

Considering the behaviours of MeanSSIM and Q_λ in Figure 6b, the insensitive performance of MvSSIM is reasonable. It is obvious that Q_λ and MeanSSIM are not sensitive to the Savitzky-Golay spectral smoothing noise,



(a) (b)

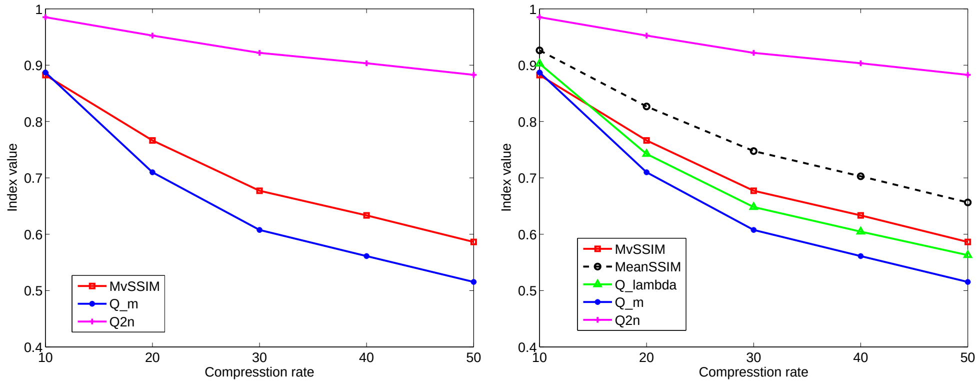
Figure 6: Assessments for the Savitzky-Golay smoothing noise contaminated HSIs.

i.e. neither the spatial and spectral structures are dramatically affected by the spectral smoothing noise. It makes sense that the spectral structural similarity is not largely affected by the Savitzky-Golay smoothing noise, because it is well known that the Savitzky-Golay filter can keep original signal structure while removing noises with proper frame sizes [19]. Thus the large values of MvSSIM is reasonable as it assesses both spatial and spectral structural similarities. However, Q_m and $Q2^n$ provide small values when the values of MeanSSIM and Q_λ are still large, which indicates that Q_m and $Q2^n$ are over-sensitive to the spectral smoothing noise.

3.3.4. JPEG2000 compression noise

Figure 7a shows the assessments of the HSIs contaminated by the JPEG2000 compression noise: Q_m is the most sensitive to the JPEG2000 compression noise, MvSSIM is less sensitive, and $Q2n$ is the least sensitive.

Considering the behaviours of MeanSSIM and Q_λ in Figure 7b, the comparative evaluation of MvSSIM, Q_m and $Q2^n$ is similar to that in 3.3.2:



(a) Gaussian white additive noise.

(b) Gaussian smoothing noise.

Figure 7: Assessments of the JPEG2000 compression noise contaminated HSIs.

$Q2^n$ does not manage to respond well to the spectral and spatial structural changes; Q_m is over-sensitive to the JPEG2000 compression noise; and MvSSIM provides index values between Q_λ and MeanSSIM, which indicates that MvSSIM more properly measures the influence of both spectral and spatial structural similarities. Thus we can prefer MvSSIM for assessing the HSIs contaminated by the JPEG2000 compression noise.

3.3.5. Summary

Two summaries could be made from these experiment results.

First, MvSSIM could provide appropriate assessments for noisy HSIs.

Second, as the indexes can perform differently for different kinds of noises, by combining the performances of the indexes for a noisy HSI, we could estimate the type of the noise added to the HSI based on the patterns of the indexes, as suggested by [17]. For example, when MvSSIM is the least sensitive to different levels of noises, there may be smoothing noise along the spectral dimension.

4. Conclusion

In this paper, we proposed a new quality assessment method called MvSSIM for 3D HSI cubes. MvSSIM explores both spatial and spectral similarities of HSI cubes. It can assess the similarities in both the within-band spatial structure and the cross-band spectral structure, by treating each pixel spectrum as a realisation of a multivariate random vector. The experiments demonstrated that MvSSIM is a proper index of quality assessment for various types of noises.

References

[1] Q. Yuan, L. Zhang, H. Shen, Hyperspectral image denoising employing a spectral–spatial adaptive total variation model, IEEE Transactions on Geoscience and Remote Sensing 50 (10) (2012) 3660–3677.

[2] H. Zhang, W. He, L. Zhang, H. Shen, Q. Yuan, Hyperspectral image restoration using low-rank matrix recovery, IEEE Transactions on Geoscience and Remote Sensing 52 (8) (2014) 4729–4743.

[3] J. Ren, J. Zabalza, S. Marshall, J. Zheng, Effective feature extraction and data reduction in remote sensing using hyperspectral imaging, IEEE Signal Processing Magazine 31 (4) (2014) 149–154.

[4] J. Li, Q. Yuan, H. Shen, L. Zhang, Hyperspectral image recovery employing a multidimensional nonlocal total variation model, Signal Processing 111 (2015) 230–248.

[5] Y.-Q. Zhao, J. Yang, Hyperspectral image denoising via sparse representation and low-rank constraint, IEEE Transactions on Geoscience and Remote Sensing 53 (1) (2015) 296–308.

[6] M. Wang, J. Yu, J.-H. Xue, W. Sun, Denoising of hyperspectral images using group low-rank representation, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 9 (9) (2016) 4420–4427.

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[8] N. Yun, Z. Feng, J. Yang, J. Lei, The objective quality assessment of stereo image, Neurocomputing 120 (2013) 121–129.

[9] F. Zhou, Q. Liao, Single-frame image super-resolution inspired by perceptual criteria, IET Image Processing 9 (1) (2015) 1–11.

[10] W. Sun, F. Zhou, Q. Liao, MDID: A multiply distorted image database for image quality assessment, Pattern Recognition 61 (2017) 153–168.

[11] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on, Vol. 2, IEEE, 2003, pp. 1398–1402.

[12] M. P. Sampat, Z. Wang, S. Gupta, A. C. Bovik, M. K. Markey, Complex wavelet structural similarity: A new image similarity index, IEEE Transactions on Image Processing 18 (11) (2009) 2385–2401.

[13] Z. Wang, Q. Li, Information content weighting for perceptual image quality assessment, IEEE Transactions on Image Processing 20 (5) (2011) 1185–1198.

[14] F. Zhou, Z. Lu, C. Wang, W. Sun, S.-T. Xia, Q. Liao, Image quality assessment based on inter-patch and intra-patch similarity, PloS one 10 (3) (2015) e0116312.

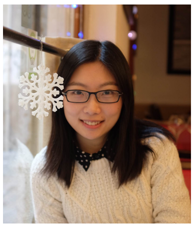
[15] L. Alparone, S. Baronti, A. Garzelli, F. Nencini, A global quality measurement of pan-sharpened multispectral imagery, IEEE Geoscience and Remote Sensing Letters 1 (4) (2004) 313–317.

[16] A. Garzelli, F. Nencini, Hypercomplex quality assessment of multi/hyperspectral images, IEEE Geoscience and Remote Sensing Letters 6 (4) (2009) 662–665.

[17] E. Christophe, D. Léger, C. Mailhes, Quality criteria benchmark for hyperspectral imagery, IEEE Transactions on Geoscience and Remote Sensing 43 (9) (2005) 2103–2114.

[18] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, L. Wald, A critical comparison among pansharpening algorithms, IEEE Transactions on Geoscience and Remote Sensing 53 (5) (2015) 2565–2586.

[19] M. Browne, N. Mayer, T. R. Cutmore, A multiscale polynomial filter for adaptive smoothing, Digital Signal Processing 17 (1) (2007) 69–75.



Rui Zhu received the B.S. degree in engineering from Xiamen University in 2012 and the M.Sc. degree in statistics from University College London in 2013. She is currently working towards the Ph.D. degree in the Department of Statistical Science, University College London. Her research interests include spectral data analysis, hyperspectral image analysis, subspace-based classification methods and metric learning.

Fei Zhou received the B.Eng. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China in 2007, and the Ph.D. degree in electronics engineering from Tsinghua University, Beijing, China in 2013. Since 2013, he has been a Postdoctoral Fellow with the Graduate School at Shenzhen, Tsinghua University, Shenzhen, China. His research interests include applications of image processing and pattern recognition in video surveillance, image super-resolution, image interpolation, image quality assessment, and object tracking.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. Since 2008, he has worked in the Department of Statistical Science at University College London as a Lecturer and Senior Lecturer. His current research interests include statistical classification, high-dimensional data analysis, pattern recognition and image analysis.

Bibliography

Alipanahi, B., M. Biggs, A. Ghodsi, et al. (2008). Distance metric learning vs. fisher discriminant analysis. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, Volume 2, pp. 598–603.

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

Basri, K. N., M. N. Hussain, J. Bakar, Z. Sharif, M. F. A. Khir, and A. S. Zoolfakar (2017). Classification and quantification of palm oil adulteration via portable NIR spectroscopy. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy 173*, 335–342.

Bennett, K. P. and E. J. Bredensteiner (2000). Duality and geometry in SVM classifiers. In *ICML*, pp. 57–64.

Berrueta, L. A., R. M. Alonso-Salces, and K. Héberger (2007). Supervised pattern recognition in food analysis. *Journal of Chromatography A 1158*(1–2), 196–214.

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

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

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

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

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

Cevikalp, H. (2016). Best fitting hyperplanes for classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Cevikalp, H. and B. Triggs (2010). Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2567–2573. IEEE.

Cevikalp, H., B. Triggs, and R. Polikar (2008). Nearest hyperdisk methods for high-dimensional classification. In *ICML*, pp. 120–127. ACM.

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

Da Silva, N. C., M. F. Pimentel, R. S. Honorato, M. Talhavini, A. O. Maldaner, and F. A. Honorato (2015). Classification of Brazilian and foreign gasolines adulterated with alcohol using infrared spectroscopy. *Forensic Science International 253*, 33–42.

Daszykowski, M., K. Kaczmarek, I. Stanimirova, Y. Vander Heyden, and B. Walczak (2007). Robust SIMCA-bounding influence of outliers. *Chemometrics and Intelligent Laboratory Systems 87*(1), 95–103.

Davis, C. B., K. W. Busch, D. H. Rabbe, M. A. Busch, and J. R. Lusk (2015). Rapid, non-destructive, textile classification using simca on diffuse near-infrared reflectance spectra. *Journal of Modern Physics 6*(06), 711.

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

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

Downey, G. (1994). Tutorial review. Qualitative analysis in the near-infrared region. *Analyst 119*(11), 2367–2375.

Ferraty, F. and P. Vieu (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Science & Business Media.

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

Fujimoto, T. and S. Tsuchikawa (2010). Identification of dead and sound knots by near infrared spectroscopy. *Journal of Near Infrared Spectroscopy 18*(6), 473–479.

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

Holloway, J., T. Priya, A. Veeraraghavan, and S. Prasad (2014). Image classification in natural scenes: Are a few selective spectral channels sufficient? In *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 655–659. IEEE.

Jaiswal, P., S. N. Jha, J. Kaur, A. Borah, and H. Ramya (2016). Detection of aflatoxin m1 in milk using spectroscopy and multivariate analyses. *Food Chemistry*.

Jayadeva, R. Khemchandani, and S. Chandra (2007). Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*(5), 905–910.

Kobayashi, T. and N. Otsu (2008). Cone-restricted subspace methods. In *19th International Conference on Pattern Recognition (ICPR 2008)*, pp. 1–4. IEEE.

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

Li, X.-l., S.-l. Yi, S.-l. He, Q. Lv, R.-j. Xie, Y.-q. Zheng, and L. Deng (2016). Identification of pummelo cultivars by using vis/nir spectra and pattern recognition methods. *Precision Agriculture 17*(3), 365–374.

Li, Z., P.-P. Wang, C.-C. Huang, H. Shang, S.-Y. Pan, and X.-J. Li (2014). Application of VIS/NIR spectroscopy for chinese liquor discrimination. *Food Analytical Methods 7*(6), 1337–1344.

Luenberger, D. G. (1969). *Optimization by Vector Space Methods*. John Wiley & Sons.

Mangasarian, O. L. and E. W. Wild (2006). Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*(1), 69–74.

Márquez, C., M. I. López, I. Ruisánchez, and M. P. Callao (2016). Ft-raman and NIR spectroscopy data fusion strategy for multivariate qualitative analysis of food fraud. *Talanta 161*, 80–86.

Mertens, B., M. Thompson, and T. Fearn (1994). Principal component outlier detection and SIMCA: a synthesis. *Analyst 119*, 2777–2784.

Moreau, J.-J. (1962). Décomposition orthogonale dun espace hilbertien selon deux cônes mutuellement polaires. *CR Acad. Sci. Paris 255*, 238–240.

Nalbantov, G., P. Groenen, and C. Bioch (2006). Nearest convex hull classification. Technical report, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute.

Pan, Z., G. Healey, M. Prasad, and B. Tromberg (2003). Face recognition in hyperspectral images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*(12), 1552–1560.

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

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

Roggo, Y., P. Chalus, L. Maurer, C. Lema-Martinez, A. Edmond, and N. Jent (2007). A review of near infrared spectroscopy and chemometrics in pharmaceutical technologies. *Journal of Pharmaceutical and Biomedical Analysis 44*(3), 683–700.

Srivastava, H. K., S. Wolfgang, and J. D. Rodriguez (2016). Expanding the analytical toolbox for identity testing of pharmaceutical ingredients: Spectroscopic screening of dextrose using portable raman and near infrared spectrometers. *Analytica Chimica Acta 914*, 91–99.

Therrien, C. W. (1975). Eigenvalue properties of projection operators and their application to the subspace method of feature extraction. *Computers, IEEE Transactions on 100*(9), 944–948.

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

Vandeginste, B. G. and D. L. Massart (1998). *Handbook of Chemometrics and Qualimetrics*. Elsevier Science.

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

Wang, Y., W. Dong, and A. Kouba (2016). Fast discrimination of bamboo species using VIS/NIR spectroscopy. *Journal of Applied Spectroscopy 83*(5), 826–831.

Weinberger, K. Q., J. Blitzer, and L. Saul (2006). Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems 18*, 1473.

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

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

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

Zhang, L., L. Zhang, D. Tao, and X. Huang (2012). On combining multiple features for hyperspectral remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing 50*(3), 879–893.

Zhang, L., L. Zhang, D. Tao, and X. Huang (2013). Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction. *IEEE Transactions on Geoscience and Remote Sensing 51*(1), 242–256.

Zhang, L., Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du (2015). Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding. *Pattern Recognition 48*(10), 3102–3112.

Zhang, L., X. Zhu, L. Zhang, and B. Du (2016). Multidomain subspace classification for hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing 54*(10), 6138–6150.

Zhou, D., B. Xiao, H. Zhou, and R. Dai (2002). Global geometry of SVM classifiers. Technical report, Technical Report 30-5-02, Institute of Automation, Chinese Academy of Sciences.

Zhou, X. and Y. Shi (2009). Nearest neighbor convex hull classification method for face recognition. In *International Conference on Computational Science*, pp. 570–577. Springer.

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