Essays on Hyperspectral Image Analysis: Classification and Target Detection

Ziyu WANG

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Security and Crime Science

Department of Statistical Science

University College London

April 4, 2017

I, Ziyu WANG, 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

Over the past a few decades, hyperspectral imaging has drawn significant attention and become an important scientific tool for various fields of real-world applications. Among the research topics of hyperspectral image (HSI) analysis, two major topics – HSI classification and HSI target detection have been intensively studied. Statistical learning has played a pivotal role in promoting the development of algorithms and methodologies for the two topics.

Among the existing methods for HSI classification, sparse representation classification (SRC) has been widely investigated, which is based on the assumption that a signal can be represented by a linear combination of a small number of redundant bases (so called dictionary atoms). By virtue of the signal coherence in HSIs, a joint sparse model (JSM) has been successfully developed for HSI classification and has achieved promising performance. However, the JSM-based dictionary learning for HSIs is barely discussed. In addition, the non-negativity properties of coefficients in the JSM are also little touched.

HSI target detection can be regarded as a special case of classification, i.e. a binary classification, but faces more challenges. Traditional statistical methods regard a test HSI pixel as a linear combination of several endmembers with corresponding fractions, i.e. based on the linear mixing model (LMM). However, due to the complicated environments in real-world problems, complex mixing effects may exist in HSIs and make the detection of targets more difficult. As a consequence, the performance of traditional LMM is limited.

In this thesis, we focus on the topics of HSI classification and HSI target detection and propose five new methods to tackle the aforementioned issues in the two tasks. For the HSI classification, two new methods are proposed based on the JSM. The first proposed method focuses on the dictionary learning, which incorporates the JSM in the discriminative K-SVD learning algorithm, in order to learn a quality dictionary with rich information for improving the classification performance. The second proposed method focuses on developing the convex cone-based JSM, i.e. by incorporating the non-negativity constraints in the coefficients in the JSM. For the HSI target detection, three approaches are proposed based on the linear mixing model (LMM). The first approach takes account of interaction effects to tackle the mixing problems in HSI target detection. The second approach called matched shrunken subspace detector (MSSD) and the third approach, called matched cone shrunken detector (MSCD), both offer on Bayesian derivatives of regularisation constrained LMM. Specifically, the proposed MSSD is a regularised subspace-representation of LMM, while the proposed MSCD is a regularised cone-representation of LMM.

All the five methods proposed in this thesis are evaluated through extensive experimental studies in the corresponding chapters.

Acknowledgements

I would like to thank the Engineering and Physical Sciences Research Council (EPSRC) and the Dean's prize from University College London (UCL) for providing the full scholarship to me. Without them, I was not be able to start my Ph.D.

I would like to express my sincere gratitude to my supervisor Dr. Jing-Hao Xue. During the four years of my Ph.D research, he has been working closely with me, providing guidance for my research direction as well as attending to the detailed problems. I would like to thank Dr Xue for his patience and help during the writing of my papers and thesis. He has always been a great advisor, mentor, and a dear friend of mine.

I would like to express my sincere gratitude to my second supervisor Prof. Thomas Fearn, who has supported my research with his immense knowledge and experience.

I would like to thank my co-authors: Prof. Kazuhiro Fukui, Dr Lefei Zhang, Dr Jianxiong Liu and Miss Rui Zhu for their selfless help and contributions to the papers we published and submitted.

My gratitude also goes to the Security Science Doctoral Research Training Centre (SECReT) in the Department of Security and Crime Science, UCL, who made my Ph.D possible. I also would like to thank the Department of Statistical Science, UCL, who hosted me for the four years of my research.

Last but not least, I would like to thank my family: my sincere gratitude to my father Jiguang Wang and my mother Fengjuan Li who have been supporting me all the time; and many thanks to my beloved husband Jianxiong Liu, who has been and will always be my great partner of life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scope of this thesis

Hyperspectral imaging, which links the remote sensing and signal processing, has been widely investigated. Hyperspectral images (HSIs), with the presentation of three-dimensional datacubes, provide rich spectral and spatial information and have been illustrated to be significantly helpful for solving real world problems. With the development of remote sensing technologies, hyperspectral imaging sensors measure the radiance of materials on the surface of the earth within a pixel area at a big range of spectral wavelength bands. An HSI pixel is then collected and formed into a high-dimensional vector, which represents the radiance at different wavelengths. The resulting high-dimensional representation by some means can provide sufficient discriminative information to identify specific materials in a scene, and is termed spectral signature [1]. On the other hand, HSI also inherits many characteristics of the two dimensional images, so that a variety of traditional image processing techniques can be immigrated and developed for HSI in terms of the spatial continuity. In short, by incorporating rich spatial information as well as spectral information, HSI analysis can provide rich information for solving real world problems.

Two research topics of hyperspectral imaging, termed *HSI classification* and *HSI target detection*, attract much attention in the research of remote sensing. The HSI classification aims to group similar HSI pixels into multiple classes. The difficulty of HSI classification is to accurately assign an unknown HSI pixel to a specific class (e.g. forest, soil), but it often enjoys relatively abundant and balanced examples for training. Typical applications include the agriculture management, surveillance and etc. In contrast, HSI target detection focuses on identifying very small objects sparsely scattered in a scene. Although it can be seen as a binary classification task, it has unique challenges such as extremely unbalanced training set and sometimes unlabelled background data. Therefore, it often requires different methodology than the HSI classification task. Typical target detection applications include mineral detection and military defence. HSI target detection can be regarded as a special case of HSI classification, i.e. a binary classification problem but with more challenges.

1.2 Contributions and outline

In this thesis, we focus on these two main directions, i.e. HSI classification and HSI target detection. For each direction, we propose several approaches. Accordingly, these approaches are based on two general models with respect to HSI classification and target detection, respectively as follows:

- joint sparse model (JSM) [4],
- linear mixing model (LMM) [5].

Firstly in Part I, we study the multi-classes classification problems of HSIs and develop two new methods based on JSM [4]. The first method focuses on the dictionary learning. Specifically, we propose to incorporate JSM [4] in the discriminative K-SVD [6] learning algorithm, in order to learn a quality dictionary with rich information for improving the classification performance. We call our proposed method joint sparse model-based D-KSVD, shortened as JSM-DKSVD. The second methods on the other hand, focuses on developing the convex cone-based JSM, which imposes the non-negativity constraints on the linear coefficients in the model. We term the proposed model C-JSM.

Secondly in Part II, we study the target detection problems of HSI. In this topic, we develop three methods based on LMM [5] for HSI target detection. We first propose a model based on matched subspace detector (MSD) [7], that in order to take into account interaction effects to tackle the mixing problems in HSI target detection. The proposed model is termed matched subspace detector with interaction effects, shortened as MSDinter. By incorporating the Tikhonov regularisation, i.e. the l_2-norm regularisation constraint into MSD, we propose another method called matched shrunken subspace detector (MSSD), which shrink the sizes of coefficients in the model for better prediction. Equally important, we analyse MSSD from the Bayesian perspective, showing that some certain prior distributions are in fact assumed in the proposed models. Moreover, we develop MSD in the non-negative coefficient space, and propose the third new method called matched shrunken cone detector (MSCD). In this cone-based analysis, we give two implementations of MSCD, which incorporate the l_1-norm regularisation term and the l_2-norm regularisation term in the cone-based representation, shortened as MSCD-l_1 and MSCD-l_2, respectively. We also derive the proposed MSCD from the Bayesian perspective, showing that two certain prior distributions of coefficients vectors are assumed in the proposed MSCD-l_1 and MSCD-l_2.

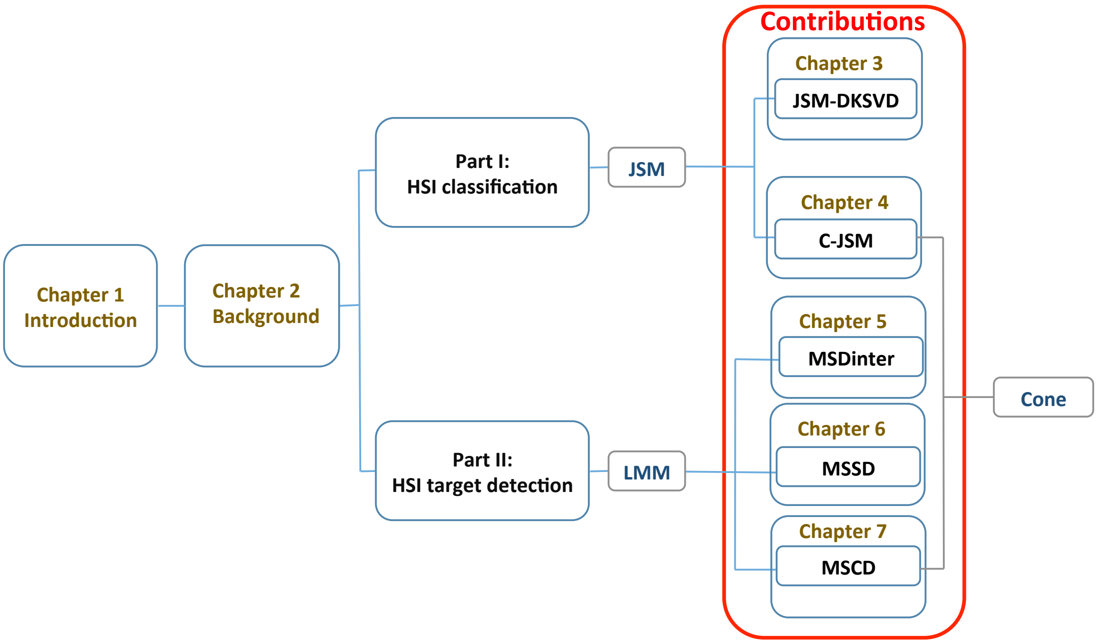


Figure 1.1: The structure of this thesis.

The main contributions of this thesis are covered in Chapter 3-7. Five papers including two publications, one revised manuscript and two submissions have been produced during the course of this thesis:

- **Ziyu Wang**, Jianxiong Liu and Jing-Hao Xue. Joint sparse model-based discriminative K-SVD for hyperspectral image classification. *Signal Processing*, 133:144-155, 2017.

- **Ziyu Wang**, Rui Zhu, Kazuhiro Fukui and Jing-Hao Xue. Cone-based joint sparse modelling for hyperspectral image classification. *IEEE Transactions on Image Processing*, 2016, submitted.

- **Ziyu Wang** and Jing-Hao Xue. The matched subspace detector with interaction effects, *Pattern recognition*, 68:24-37, 2017.

- **Ziyu Wang** and Jing-Hao Xue. Matched shrunken subspace detectors for hyperspectral target detection, *Neurocomputing*, 2016, revised.

- **Ziyu Wang**, Rui Zhu, Kazuhiro Fukui and Jing-Hao Xue. Matched Shrunken cone detector (MSCD): Bayesian derivations and case studies for hyperspectral target detection. *IEEE Transactions on Image Processing*, 2017, submitted.

The rest of thesis is organised as follows and summarised in Figure 1.1.

Background (Chapter 2)

This chapter gives a brief literature review of the relative works of HSI classification and target detection. The process of how the hyperspectral images are collected is introduced and followed by general overviews of HSI classification and HSI target detection. The limitations of the current methods are also discussed.

HSI classification: joint sparse model-based discriminative K-SVD (JSM-DKSVD) (Chapter 3)

Sparse representation classification (SRC) is being widely investigated on hyperspectral images (HSI). For SRC methods to achieve high classification performance, not only is the development of sparse representation models essential, the designing and learning of quality dictionaries also plays an important role. That is, a redundant dictionary with well-designated atoms is required in order to ensure low reconstruction error, high discriminative power, and stable sparsity. In this chapter, we propose a new method to learn such dictionaries for HSI classification. We borrow the concept of JSM [4] from SRC to dictionary learning. JSM assumes local smoothness and joint sparsity and was initially proposed for classification of HSI. We leverage JSM to develop an extension of discriminative K-SVD [6] for learning a promising discriminative dictionary for HSI. Through a semi-supervised strategy, the new dictionary learning method, termed JSM-DKSVD, utilises all spectra over the local neighbourhoods of labelled training pixels for discriminative dictionary learning. It can produce a redundant dictionary with rich spectral and spatial information as well as high discriminative power. The learned dictionary can then be compatibly used in conjunction with the established SRC methods, and can significantly improve their performance for HSI classification.

- **Ziyu Wang**, Jianxiong Liu and Jing-Hao Xue. Joint sparse model-based discriminative K-SVD for hyperspectral image classification. *Signal Processing*, 133:144-155, 2017.

HSI classification: cone-based joint sparse modelling (C-JSM) (Chapter 4)

In JSM [4], it is assumed that neighbouring hyperspectral pixels can share sparse representations. However, the coefficients of the endmembers used to reconstruct a test HSI pixel is desirable to be non-negative for the sake of physical interpretation. Hence in this chapter, we introduce the non-negativity constraint into JSM. The non-negativity constraint implies a cone-shaped space instead of the infinite sample space for pixel representation. This leads us to propose a new model called cone-based joint sparse model (C-JSM), to install the non-negativity on top of the sparse and joint modelling. To solve the C-JSM problem, we also propose a new algorithm through introducing the non-negativity constraint into the simultaneous orthogonal matching pursuit (SOMP) [8] algorithm. The new algorithm is called non-negative simultaneous orthogonal matching pursuit (NN-SOMP). Experiments and investigations show that the proposed C-JSM can produce a more stable, sparse representation and a superior classification than other methods which only ensure the sparsity, non-negativity or spatial coherence.

- **Ziyu Wang**, Rui Zhu, Kazuhiro Fukui and Jing-Hao Xue. Cone-based joint sparse modelling for hyperspectral image classification. *IEEE Transactions on Image Processing*, 2017, submitted.

HSI target detection: matched subspace detector with interaction effects (MSDinter) (Chapter 5)

In this chapter, a new hyperspectral target-detection method termed the matched subspace detector with interaction effects (MSDinter) is proposed. The MSDinter introduces "interaction effects" terms into the popular matched subspace detector (MSD [7], from regression analysis in multivariate statistics and the bilinear mixing model in hyperspectral unmixing. In this way, the interaction between the target and the surrounding background, which should have but not yet been considered by the MSD, is modelled and estimated, such that superior performance of target detection can be achieved. Besides deriving the MSDinter methodologically, we also demonstrate its superiority empirically using two hyperspectral imaging datasets.

- **Ziyu Wang** and Jing-Hao Xue. The matched subspace detector with interaction effects, *Pattern recognition*, 68: 24-37, 2017.

HSI target detection: matched shrunken subspace detectors (MSSD) (Chapter 6)

In this chapter we propose a new approach, called the matched shrunken subspace detector (MSSD), to target detection from hyperspectral images. The MSSD is developed by shrinking the abundance vectors of the target and background subspaces in the hypothesis models of the matched subspace detector (MSD) [7], a popular subspace-based approach to target detection. The shrinkage is achieved by introducing simple l_2-norm regularisation (also known as ridge regression or Tikhonov regularisation [9]). We develop two types of MSSD, one with isotropic shrinkage and thus termed MSSD-i and the other with anisotropic shrinkage and termed MSSD-a. For these two new methods, we provide both the frequentist and Bayesian derivations. Experiments on a real hyperspectral imaging dataset called Hymap demonstrate that the proposed MSSD methods can outperform the original MSD for hyperspectral target detection.

- **Ziyu Wang** and Jing-Hao Xue. Matched shrunken subspace detectors for hyperspectral target detection, *Neurocomputing*, 2017, revised.

HSI target detection: matched shrunken cone detectors (MSCD) (Chapter 7)

Hyperspectral images (HSIs) possess *non-negative* properties for both hyperspectral signatures and abundance coefficients, which can be naturally modelled using cone-based representation. However, in hyperspectral target detection, cone-based methods are barely studied. In this chapter, we propose a new regularised cone-based representation approach to hyperspectral target detection, as well as its two working models by incorporating into the cone representation l_2-norm and l_1-norm regularisations, respectively. We call the new approach the matched shrunken cone detector (MSCD). Also important, we provide principled derivations of the proposed MSCD from the Bayesian perspective: we show that MSCD can be derived by assuming a multivariate half-Gaussian distribution or a multivariate half-Laplace distribution as the prior distribution of the coefficients of the models. In the experimental studies, we compare the proposed MSCD with the subspace methods and the sparse representation-based methods for HSI target detection. Two real hyperspectral datasets are used for evaluating the detection performances on sub-pixel targets and full-pixel targets, respectively. Results show that the proposed MSCD can outperform other methods in both cases, demonstrating the effectiveness of the regularised cone-based representation.

- **Ziyu Wang**, Rui Zhu, Kazuhiro Fukui and Jing-Hao Xue. Matched Shrunken cone detector (MSCD): Bayesian derivations and case studies for hyperspectral target detection. *IEEE Transactions on Image Processing*, 2017, submitted.

Chapter 2

Background

In this chapter, the concept of hyperspectral imaging is firstly introduced. Then we give a brief literature review of the relative works of HSI classification and HSI target detection. Limitations of the current methods for solving the two problems are finally identified.

2.1 Hyperspectral imaging

In general, hyperspectral imaging is the process of taking “photos” of objects at a wide range of spectra. Different from the regular black-and-white photos, a hyperspectral image (HSI) is a collection of the objects’ radiance responses at each spectral band, typically in the number of hundreds, and therefore is a three-dimensional cube. To some extent, a colour photo can be seen as an overly simple example of an HSI, with only three spectral bands.

In a hyperspectral imaging system, four typical components are included: the illumination source, e.g. sun light, the atmospheric path, the region of interests (ROIs) and the sensor [1]. In [1], Manolakis et al. summarise the whole process as follows and illustrated in Figure 2.1: the hyperspectral sensor, typically on satellites or aircraft, collects the spectral information with three parts: the sunlight, the atmospheric attenuation, and the objects in the ROI. The energy reflected by the surface materials are different; and the sensor can detect and measure the intensity of the energy at different spectral bands. The information is then processed to be an hyperspectal dataset.

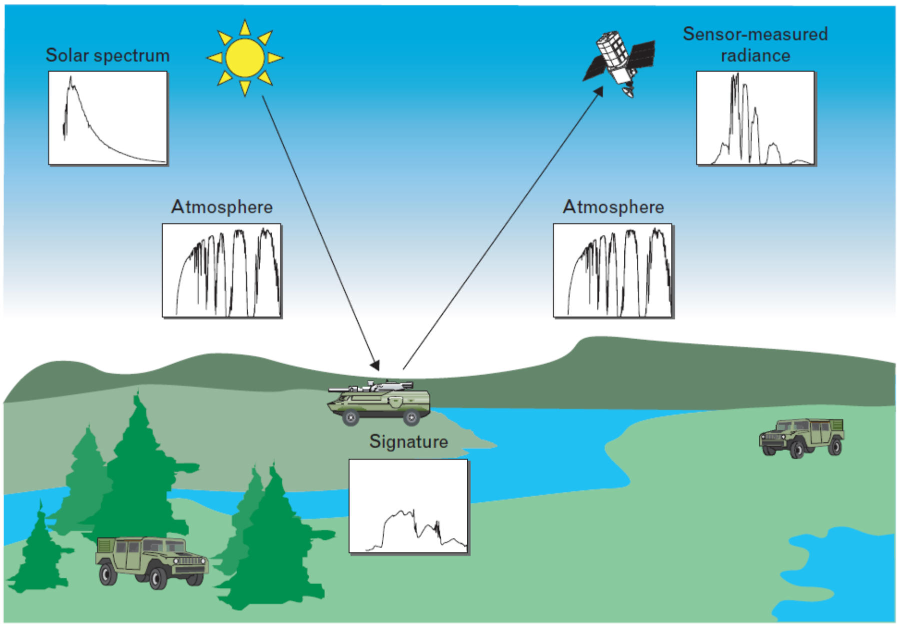


Figure 2.1: An illustration of a hyperspectral image scene [1].

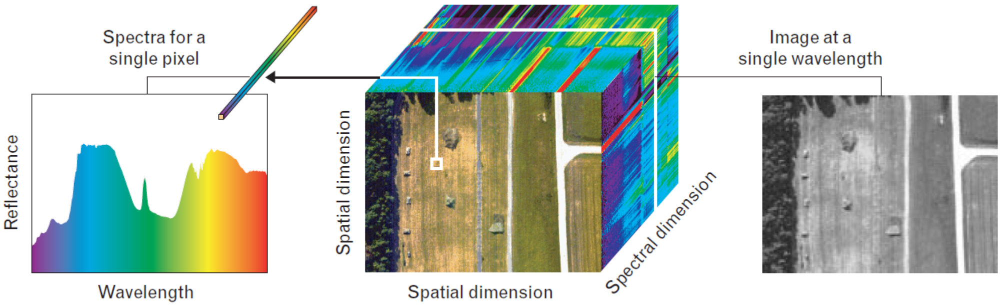


Figure 2.2: An illustration of a hyperspectral image data-cube [1].

An obtained HSI is a three-dimensional data-cube, as shown in Figure 2.2. The data-cube includes two spatial dimensions and one spectral dimension. If we regard the spectral values as a function of wavelength, the resultant high-dimensional vector is termed a spectra or a spectral signature of an HSI pixel; if we extract the pixel values at all coordinates at the same wavelength, a two-dimensional image is obtained.

HSI classification and target detection are pixel-wise problems, which aim to identify each HSI pixel in the given scene to a desired class. The resultant high-dimensional vector of the HSI pixel can provide sufficient information to identify the materials. However, due to the limitations of the sensors and the interruption of atmospheric attenuation, the obtained HSI pixel may be mixed by some uncertain factors, such as the interaction between the neighbouring materials. To tackle this issue, researchers have proposed a variety of machine learning and image processing techniques, among which some representative methods are briefly reviewed in the following sections.

2.2 HSI classification

In the HSI classification, sparse representation classification (SRC), proposed in [10], is being widely investigated on HSI. It is based on the assumption that high-dimensional data from the same class lie in a low-dimensional subspace. Therefore a signal can be represented by a linear combination of a small number of redundant bases (so-called dictionary atoms). In this thesis, we mainly focus on the SRC-based methods for HSI classification.

2.2.1 Sparse model (SM)

Suppose a B-dimensional pixel, denoted by $\mathbf{x} \in \mathbb{R}^B$, can be approximated by a linear combination of N_D training pixels:

$$\mathbf{x} \approx \mathbf{D}\alpha \tag{2.1}$$

where $\mathbf{D} \in \mathbb{R}^{B \times N_D}$ denotes a dictionary constructed by the N_D training pixels (also termed atoms), and α is the N_D-dimensional vector of coefficients in the linear combination.

In a sparse model (SM), $\mathbf{x}$ can be approximated by only a few (e.g. at most L_C) atoms in $\mathbf{D}$. That is, the coefficient vector α is sparse. The values of α can be estimated by solving the following optimisation problem:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 \ , \ s.t. \ \|\alpha\|_0 \leq L_C \tag{2.2}$$

where $\|\alpha\|_0$ denotes a l_0-pseudo-norm (i.e. the number of non-zero elements) of α, L_C ($L_C \ll N_D$) is defined as the upper bound of the sparsity level of the model. The problem in (2.2) is NP-hard, but it can be approximately solved by greedy pursuit algorithms such as orthogonal matching pursuit (OMP) [11] or be relaxed by replacing the l_0-pseudo-norm with the l_1-norm. When the problem is solved by OMP, the dictionary $\mathbf{D}$ is column-wise normalised to have unit l_2-norm.

For the SM, the class of $\mathbf{x}$ is determined by applying the obtained sparse coefficient vector $\hat{\alpha}$ from (2.2). We define the class-wise residuals as

$$r^m(\mathbf{x}) = \|\mathbf{x} - \mathbf{D}^m \hat{\alpha}^m\|_2^2,\ m = 1, \dots, M, \tag{2.3}$$

where M is the total number of classes, $\hat{\alpha}^m$ contains the N_m elements in $\hat{\alpha}$ that are associated with sub-dictionary $\mathbf{D}^m$ of the mth class, with $N = \sum_{m=1}^{M} N_m$. The label of the test pixel $\mathbf{x}$ is determined by its minimal residual over all M classes:

$$Class(\mathbf{x}) = \operatorname*{argmin}_{m=1,\dots,M} r^m(\mathbf{x}). \tag{2.4}$$

2.2.2 Joint sparse model (JSM)

In HSI, neighbouring pixels in a small area often consist of similar materials and the classes of these materials are few. Hence, local smoothness and sparsity can be assumed for HSI. In the joint sparse model (JSM) [4], it is assumed that all neighbouring pixels around a central pixel share a common sparse pattern. The modelling, learning and labelling for JSM can be described as follows.

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{T_C}]$, a $B \times T_C$ matrix, denote a small window consisting of T_C pixels and centring on a test pixel $\mathbf{x}_c$, with each pixel $\mathbf{x}_t$ represented by a B-dimensional vector for B spectral bands. The T_C pixels are approximated by sparse linear combinations of atoms from a given dictionary:

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{T_C}] \approx \mathbf{D}[\alpha_1, \dots, \alpha_{T_C}] = \mathbf{DA} \tag{2.5}$$

where $\mathbf{D} \in \mathbb{R}^{B \times N_D}$ is a dictionary with N_D known and labelled atoms, and $\mathbf{A} \in \mathbb{R}^{N_D \times T_C}$ is the matrix of unknown coefficients $[\alpha_1, \dots, \alpha_{T_C}]$. Because of the local smoothness and sparsity, we can assume that there are only L_C ($L_C \ll N_D$) non-zero rows in $\mathbf{A}$. This leads to the so-called joint sparse model (JSM), where the non-zero rows form the support shared by coefficient vectors $\{\alpha_t\}_{t=1}^{T_C}$. That is, $\{\alpha_t\}_{t=1}^{T_C}$ are sparse vectors and $\mathbf{A}$ is a sparse matrix. An illustration of the JSM equation is shown in Figure 2.3

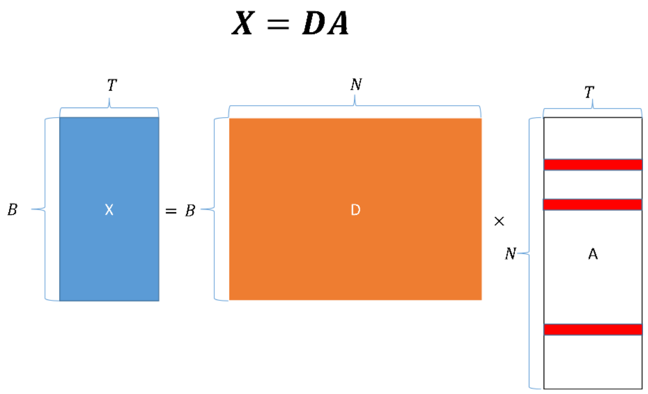


Figure 2.3: An illustration of JSM, where $\mathbf{X}$ is a $B \times T$ matrix denoting a small window consisting of T HSI pixels, $\mathbf{D}$ is a $B \times N$ matrix representing an over-complete dictionary with N atoms; and $\mathbf{A}$ is an $N \times T$ coefficient matrix with only L non-zero rows. The red lines in $\mathbf{A}$ indicate the non-zero rows and the blank areas indicate zero rows of $\mathbf{A}$.

The learning of JSM is to estimate $\mathbf{A}$, which can be achieved by solving a joint sparse recovery problem:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 \ , \ s.t. \ \|\mathbf{A}\|_{row,0} \leq L_C \tag{2.6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm; $\|\mathbf{A}\|_{row,0}$, the row-wise l_0-norm, is the number of non-zero rows of $\mathbf{A}$. As with (2.2), problem (2.6) is NP-hard and it can be approximately solved by greedy algorithms such as the Simultaneous Orthogonal Matching Pursuit algorithm (SOMP) [8] or the Simultaneous Subspace Pursuit algorithm (SSP) [4]. When solved by SOMP or SSP, the dictionary $\mathbf{D}$ is column-wise normalised to have unit l_2-norm.

For the JSM, once the sparse coefficient matrix $\hat{\mathbf{A}}$ is obtained from (2.6), we calculate the class-wise residual of the matrix $\mathbf{X}$ from its class-wise approximation similar to (2.3):

$$r^m(\mathbf{X}) = \left\|\mathbf{X} - \mathbf{D}^m\hat{\mathbf{A}}^m\right\|_F^2, \ m = 1, \ldots, M. \tag{2.7}$$

In (2.7), there are N_m rows in $\hat{\mathbf{A}}$ corresponding to a sub-dictionary $\mathbf{D}^m$ and $N_D =$

$\sum_{m=1}^{M} N_m$. Different from the SM, the label of the central test pixel $\mathbf{x}_c$ in window $\mathbf{X}$ is jointly determined by the minimal residual of $\mathbf{X}$ over all M classes, i.e.

$$Class(\mathbf{x}_c) = \underset{m=1,\dots,M}{\text{argmin}}\ r^m(\mathbf{X}). \tag{2.8}$$

2.2.3 Limitations of SRC-based HSI classification

- **Lack of good quality of dictionary**

  For the SRC-based method for HSI classification, the dictionary $\mathbf{D}$ is often constructed directly by the HSI pixels, so that the variety of the atoms are limited. To achieve higher classification performance, a well-designed dictionary would have good representation power over certain sparsity, as well as to support optimal discrimination of class [6]. However, there are limited number of works on developing the dictionary learning algorithms specifically for HSI classification problems. It is desirable to incorporate the spatially structure information into the training process to learn a more powerful dictionary. To achieve this goal, we propose a new dictionary learning method in Chapter 3.

- **Lack of non-negativity constraints on coefficients**

  In the modelling of HSI pixels, an important property of hyperspectral signals is the non-negativity, for both the signal itself and the abundance coefficients. However, research of SRC-based methods particularly the JSM-based methods have not incorporated the non-negativity properties in the HSI. The non-negative constraints on the coefficients induce a cone-shape representation [12]. To fill the gap, we replace the signal representation of JSM by cone representation, and incorporate the non-negativity constraints into the HSI classification. The proposed method is detailed in Chapter 4.

2.3 HSI target detection

HSI target detection aims to detect small objects or anomalies in a hyperspectral image. HSI target detection is essentially a binary classification problem, of which the task is to determine if an HSI pixel is a target spectrum or a background spectrum. Hence, target detection can be conducted by a binary hypothesis model with two competing hypotheses: the null hypothesis H_0 for the absence of the target; and the alternative hypothesis H_1 for the presence of the target. Binary hypothesis models for target detection have been nicely reviewed in [13, 14, 15, 16].

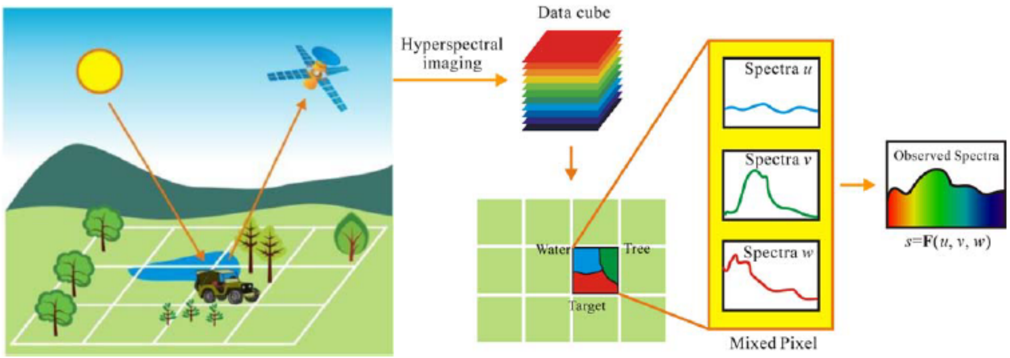


Figure 2.4: An illustration of a mixed pixel of an HSI [2].

2.3.1 Linear mixing model (LMM)

Target objects often appear as sub-pixels in an HSI. That is, the spectrum of an HSI pixel can be a mixture of different component spectra of materials, as shown in Figure 2.4. These component spectra are usually termed endmembers. To model the mixture of an HSI pixel, the linear mixing model (LMM) [5] has been widely adopted. The underlying assumption of LMM is that an HSI pixel can be approximated by a linear combination of endmembers with different fractions. The weight (abundance) of each endmember spectrum is proportional to the fraction of the pixel area covered by the endmember. If there are p spectral bands, the p-variate spectrum $\mathbf{x} = [x_1, \ldots, x_p]^T$ of a mixed pixel can be expressed as a mixture of K endmembers $\mathbf{m}_k$ with additive noise:

$$\mathbf{x} = \Sigma_{k=1}^{K} a_k \mathbf{m}_k + \mathbf{n} = \mathbf{Ma} + \mathbf{n}, \tag{2.9}$$

where $\mathbf{M}$ is a $p \times K$ matrix whose columns are the K endmember spectra $\mathbf{m}_k = [m_{k,1}, \ldots, m_{k,p}]^T$ for $k = 1, \ldots, K$, respectively; $\mathbf{a} = [a_1, \ldots, a_K]^T$ is the fraction abundance vector; and $\mathbf{n} = [n_1, \ldots, n_p]^T$ represents the additive Gaussian white noise, i.e. $\mathbf{n} \sim N(\mathbf{0}, \mathbf{C})$, where $\mathbf{C}$ is a $p \times p$ covariance matrix. Physical considerations dictate that the abundances have to satisfy 1) the non-negative constraint, i.e. $a_k \geq 0$, and 2) the sum-to-one constraint, i.e. $\Sigma_{k=1}^{K} a_k = 1$ [17]. Although the non-negative constraint and the sum-to-one constraint are quite meaningful, they are not always enforced because it significantly complicates the solving of detection problems. As explained in [5] and as usually the case, both constraints can be relaxed in target detection.

Based on LMM, several methods have been developed and can be summarised into two directions: 1) subspace-based methods and 2) sparse-representation-based methods.

2.3.1.1 Subspace-based methods

- **Matched subspace detector** (**MSD**) [7]

  In the MSD, the target spectral signatures and background spectral signatures are represented by the bases of a target subspace and the bases of a background subspace, respectively. The underlying assumption of the MSD is that each basis vector of these subspaces represents an endmember, which is formulated as follows:

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \ \mathbf{x} \text{ is a background pixel,} \\ &H_1 : \mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{n}_1, \ \mathbf{x} \text{ is a target pixel,} \end{aligned} \tag{2.10}$$

  where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}]$ is a $p \times r_t$ matrix representing the target subspace, and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$ is a $p \times r_b$ matrix representing the background subspace; $\mathbf{T}$ is derived from a training target matrix $\mathbf{M}_T \in \mathbb{R}^{p \times N_t}$ whose columns are the N_t target spectra, and $\mathbf{B}$ is derived from a training background matrix $\mathbf{M}_B \in \mathbb{R}^{p \times N_b}$ whose columns are the N_b background spectra; γ and β are the corresponding abundance vectors of the subspaces $\mathbf{T}$ and $\mathbf{B}$, respectively; and $\mathbf{n}_0$ and $\mathbf{n}_1$ are p-dimensional vectors of Gaussian white noise: $\mathbf{n}_0 \sim N(\mathbf{0}, \sigma_0^2\mathbf{I})$ and $\mathbf{n}_1 \sim N(\mathbf{0}, \sigma_1^2\mathbf{I})$, respectively.

The output detector of the MSD model is solved by least square estimates (LSE) and is given by

$$D_{\text{MSD}}(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{P}_B^{\perp} \mathbf{x}}{\mathbf{x}^T \mathbf{P}_V^{\perp} \mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu_{MSD}, \tag{2.11}$$

where $\mathbf{P}_B^{\perp} = \mathbf{I} - \mathbf{P}_B$ with $\mathbf{P}_B = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$ being the projection matrix onto the column space of $\mathbf{B}$; and $\mathbf{P}_V^{\perp} = \mathbf{I} - \mathbf{P}_V$ with $\mathbf{P}_V = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ being the projection matrix onto the column space of $\mathbf{V}$, where $\mathbf{V}$ is a $p \times (r_t + r_b)$ concatenated matrix of $\mathbf{T}$ and $\mathbf{B}$, i.e. $\mathbf{V} = [\mathbf{T}, \mathbf{B}]$.

The value of $D_{\text{MSD}}(\mathbf{x})$ is compared to a threshold ν_{MSD} to make a final decision of which hypothesis should be rejected for test pixel $\mathbf{x}$. In general, any set of orthogonal basis vectors that spans the corresponding subspace can be used as the column vectors of $\mathbf{B}$ and $\mathbf{T}$. In this thesis, the significant eigenvectors (normalised by the square roots of their corresponding eigenvalues) of the background and target covariance matrices $\mathbf{C}_b$ and $\mathbf{C}_t$ are used to create the column vectors of $\mathbf{B}$ and $\mathbf{T}$, respectively. The MSD method will be detailed in Chapter 5.

- **Orthogonal subspace projection** detector (**OSP**) [18]

  OSP aims to maximise the signal-to-noise (SNR) ratio in the subspace that is orthogonal to the background subspace. Given the spectral signature of the target material $\mathbf{t} \in \mathbb{R}^p$ and the LMM (2.9), the OSP detector is formulated as

$$D_{OSP}(\mathbf{x}) = \mathbf{t}^T \mathbf{P}_{\mathbf{B}}^{\perp} \mathbf{x}. \tag{2.12}$$

  With the same notation in MSD (2.11), $\mathbf{P}_B$ is the projection matrix derived from the background subspace $\mathbf{B}$.

- **Constrained energy minimisation** (**CEM**) [19, 20]

  For the scenario where only the spectra signature of the target is known and any background spectra are unknown, a method called constrained energy minimisation (CEM) is developed. It is a finite-impulse response filter which minimise the output energy subject to a constrained imposed by desired target spectrum $\mathbf{t}$. The solution of the constrained problem is

$$D_{CEM} = (\mathbf{t}^T \mathbf{R}_r^{-1} \mathbf{t})^{-1} \mathbf{R}_r^{-1} \mathbf{t}, \tag{2.13}$$

where $\mathbf{R}_r = (1/q) \sum_{i=1}^{q} \mathbf{r}_i \mathbf{r}_i^T$ is the data sample correlation matrix. In [19], it has been shown that the CEM and the OSP are closely related. They are essentially equivalent as long as the noise is white and its variance is negligible compared to the signals.

- **Adaptive coherence/cosine detector** (**ACE**) [21, 22]

  Adaptive coherence/cosine detector (ACE), also termed adaptive subspace detectors (ASD) is based on the following competing hypotheses:

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{n}, \text{ target absent,} \\ &H_1 : \mathbf{x} = \mathbf{T}\gamma + \sigma \mathbf{n}, \text{ target present.} \end{aligned} \tag{2.14}$$

  Different from the formulae in MSD (2.10), the test HSI pixel $\mathbf{x}$ is assume be a Gaussian white noise $\mathbf{n} \sim N(\mathbf{0}, \mathbf{C})$ in the null hypothesis H_0 and is assumed to be a linear combination of target subspace signal and a scaled background noise. Note that the background noise is assumed to have the same covariance matrix $\mathbf{C}$ under H_0 and H_1. The ACE detector is formulated as

$$D_{ACE}(\mathbf{x}) = \frac{\mathbf{x}^T \hat{\mathbf{C}}^{-1} \mathbf{T} (\mathbf{T}^T \hat{\mathbf{C}}^{-1} \mathbf{T})^{-1} \mathbf{T}^T \hat{\mathbf{C}}^{-1} \mathbf{x}}{\mathbf{x}^T \hat{\mathbf{C}}^{-1} \mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu_{ACE}, \tag{2.15}$$

  where $\hat{\mathbf{C}}$ is the maximum likelihood estimate (MSE) of the covariance $\mathbf{C}$ and ν_{ACE} is the threshold.

2.3.1.2 Sparse-representation-based methods

Sparse representation techniques have also been developed in HSI target detection with the same motivation as that of SRC for HSI classification. Given an overcomplete dictionary including sufficient background atoms and target atoms, a target HSI pixel is assumed to be represented by only a few atoms in the dictionary.

- **Sparse target detection (STD)** [23]

  Given a dictionary that consists of background endmembers (samples) and target endmembers (samples), STD employs reconstruction residuals to perform the target detection. The test HSI pixel **x** can be sparsely represented by the linear combination of all endmembers(training samples) as follows:

  $$\mathbf{x} \approx= \mathbf{D}^b \alpha^b + \mathbf{D}^t \alpha^t = \mathbf{D}\alpha, \tag{2.16}$$

  where $\mathbf{D}^b$ and $\mathbf{D}^t$ are the $p \times N_b$ background dictionary and the $p \times N_t$ target dictionary respectively; and α^b and α^t are the corresponding N_b-dimensional and N_t-dimensional sparse coefficient vectors with only a few non-zero elements, respectively. The sparse coefficient vector α can be recovered by

  $$\hat{\alpha} = \operatorname{argmin} \|\mathbf{D}\alpha - \mathbf{x}\|_2^2, \text{s.t. } \|\alpha\|_0 \leq L_0. \tag{2.17}$$

  As with the SM, $\|\cdot\|_0$ denote a l_0-pseudo-norm of α; L_0 is the upper bound of the sparsity level and can be solved by the OMP algorithm.

  Once the sparse coefficient vector $\hat{\alpha}$ is obtained, it is then decomposed into $\hat{\alpha^b}$ and $\hat{\alpha^t}$. The detection is performed based on two competing reconstructions of residuals $r_b(\mathbf{x})$ and $r_t(\mathbf{x})$ using only the background dictionary and only the target dictionary respectively:

  $$\begin{aligned} r_b(\mathbf{x}) &= \left\|\mathbf{x} - \mathbf{D}^b \hat{\alpha^b}\right\|_2^2, \\ r_t(\mathbf{x}) &= \left\|\mathbf{x} - \mathbf{D}^t \hat{\alpha^t}\right\|_2^2. \end{aligned} \tag{2.18}$$

  The label of the test HSI pixel is finally determined by

  $$D_{STD}(\mathbf{x}) = r_b(\mathbf{x}) - r_t(\mathbf{x}). \tag{2.19}$$

- **Sparse representation-based binary hypothesis model (SRBBH)** [24]

Different from STD, SRBBH adopts the binary hypothesis models and follows the same framework of MSD. When there is no target presenting, the test HSI pixel **x** is only represented by the sparsely linear combination of all background atoms in the dictionary $\mathbf{D}^b$. When a target presents, **x** is represented by both the background atoms from $\mathbf{D}^b$ and target atoms from $\mathbf{D}^t$. The models of SRBBH are given by

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{D}^b \alpha^b + \mathbf{n}_0, \text{ target absent,} \\ &H_1 : \mathbf{x} = \mathbf{D}^t \alpha^t + \mathbf{D}^b \alpha^b + \mathbf{n}_1, \text{ target present.} \end{aligned} \tag{2.20}$$

where $\mathbf{n}_0$ and $\mathbf{n}_1$ are approximated residuals. In H_0 and H_1, same upper-bound of sparsity level L are employed.

As with STD, SRBBH models are also solved by the OMP algorithm. It shall be noted that two sparse recovery problems shall be solved thus OMP shall be employed twice in SRBBH rather than once in STD. The residuals of H_0 model and H_1 model are computed as follows

$$\begin{aligned} r_0(\mathbf{x}) &= \left\| \mathbf{x} - \mathbf{D}^b \hat{\alpha^b} \right\|_2^2, \\ r_1(\mathbf{x}) &= \| \mathbf{x} - \mathbf{D}\hat{\alpha} \|_2^2, \end{aligned} \tag{2.21}$$

and the label of the test HSI pixel **x** is then determined by

$$D_{SRBBH}(\mathbf{x}) = r_0(\mathbf{x}) - r_1(\mathbf{x}). \tag{2.22}$$

2.3.2 Limitations of the LMM for HSI target detection

- **Complex mixing problems in an HSI pixel**

The underlying assumption of the LMM is that target spectral signature in the scene remains linearly mixed with the surrounding background spectra after enter the hyperspectral sensor. However this is not always true in practice. The exhibited target spectrum may be contaminated by the surrounding environments due to the multiple scattering effects during the image capturing process. As a result, the abundance vector of targets will be dependent on the characteristics of their surrounding background. It is necessary to build a new model to cope with the multiple scattering problems. To tackle this problem, we propose a new approach to account for the effect interactions for the HSI target detection and analyse it from the statistical point of view in Chapter 5.

- **High variance of coefficients estimations**

  It is known that LMM-based methods may suffer from the problem of high variance of coefficients estimations. To adjust the performance of a statistical model, some prior domain knowledge about the model, particularly the coefficients, can be incorporated by imposing regularisation, i.e. a frequentist fashion, or assuming the prior distribution, i.e. a Bayesian fashion. From the Bayesian perspective, an improper uniform prior distribution is actually assumed for the coefficients in the conventional LMM thus non-informative. It is desirable to develop the shrinkage methods [9], such as the popular lasso, i.e. l_1-norm regularisation and the Tikhonov regularisation, i.e. l_2-norm regularisation for the HSI target detection. To achieve this goal, we proposed two new approaches by imposing the regularisation terms in the LMM-based models. Particularly, we proposed a subspace-representation-based method, called matched shrunken subspace detector (MSSD) (in Chapter 6) and a cone-representation-based method, called matched shrunken cone detector (MSCD) (in Chapter 7) respectively, and provide both of the frequentist and Bayesian derivations for them.

Part I

Contributions to HSI Classification

Chapter 3

HSI Classification: Joint Sparse Modelling-based Discriminative K-SVD (JSM-DKSVD)

3.1 Introduction

Sparse representation classification (SRC), proposed in [10], is being widely investigated on hyperspectral images (HSI). It is based on the assumption that high-dimensional data from the same class lie in a low-dimensional subspace. Therefore a signal can be represented by a linear combination of a small number of redundant bases (so-called dictionary atoms). In [4], Chen et al. apply SRC and propose a joint sparse model (JSM) to HSI classification. JSM assumes that all HSI pixels in a small spatial neighbourhood can be jointly approximated by sparse linear combinations of a few common training samples, which can be solved by the simultaneous orthogonal matching pursuit (SOMP) algorithm [8]. However, in JSM all neighbouring pixels make equal contributions to the sparse recovery of the central pixel. To determine more effective neighbours for JSM, several appealing ideas have been proposed [25, 26, 27, 28, 29]. In [25], Zhang et al. introduce a non-local approach [26], which assumes that a candidate has its weight determined by the similarity between its neighbourhood and the central pixel's neighbourhood, termed non local weighting (NLW). Tang et al. propose two manifold-based l_1-norm methods, using locally linear embedding and Laplacian eigenmap to regularise local structures of pixels [27]. In [28] and [29], Fang et al. and Li et al. propose to adopt superpixel methods [30, 31] to integrate the spatial structures for JSM. The superpixel is regarded as a small spatial local region which is adaptive in shape and size.

To achieve high classification performance, not only is the development of sparse representation models essential, the designing and learning of quality dictionaries also plays an important role. A well-designed dictionary would have good representation power over a certain sparsity, as well as to support optimal discrimination of classes [6]. Previous literatures have shown that dictionary learning is beneficial to signal representation as well as to classification [6, 32, 33, 34]. In [32], Aharon et al. propose K-SVD, a generalised K-means method, to minimise the signal reconstruction error. It alternates between sparse coding by orthogonal matching pursuit (OMP) [11] and dictionary updating by singular value decomposition (SVD). For face recognition, Zhang et al. introduce into sparse representation a constraint to model classification error [6]. A K-SVD algorithm is then adopted to minimise the sum of reconstruction error and classification error, named as discriminative K-SVD (D-KSVD). In [33], Jiang et al. propose label consistent K-SVD, which incorporates a label-consistent term into D-KSVD, leading to an explicit correspondence between the dictionary atoms and labels. It also adopts the K-SVD algorithm to solve the optimisation problem. Mairal et al. propose task-driven dictionary learning (TDDL) [34], which is a general formulation for learning sparse representations tuned for specific tasks. TDDL not only can be designed for classification, but also can be designed for regression and compressive sensing.

There have been a limited number of works on developing the dictionary learning algorithms specifically for HSI classification problems. In [28], Fang et al. propose to use a modified class-labelled OMP algorithm in D-KSVD to learn a dictionary of better discriminative power. In [35], Soltani-Farani et al. partition given pixels into contextual groups, and jointly model pixels inside the same contextual group to be in a common subspace. Both methods endeavour to make a better use of the limited amount of labelled training data. Taking one step further, Wang et al. utilise spatial context of a test pixel within its local neighbourhood to develop a learning vector quantization (LVQ)-based dictionary learning method [36]. In [37], Sun et al. introduce the use of structure information into dictionary learning. They argue that the requirement of a redundant dictionary in sparse coding can be lessened if simultaneous sparse approximation is employed. Therefore they aim to produce a compact dictionary by using a joint or Laplacian sparsity prior and the TDDL framework [34]. Wang et al. follow the same TDDL framework and introduce a more explicitly formulated semi-supervised problem to the compact dictionary learning [38].

In this context, we believe that, in order to develop a dictionary with high discriminative power for HSI classification but from only a limited number of labelled training samples, it is a promising direction to utilise the structure information as much as possible. Considering the discriminative nature of D-KSVD and its imperfection of exploiting spectral signatures only, we think D-KSVD has substantial room to be explored for improvement. Furthermore, we are highly impressed by the recent progress in HSI classification made by the JSM-based algorithms from its leveraging both spectral and spatial information in the representation of HSI pixels. All these factors inspire us to develop a new dictionary learning approach for HSI classification, by enforcing the JSM assumption, of local smoothness and joint sparsity around the limited number of training sample, into D-KSVD through a semi-supervised fashion. In this chapter, we propose a new approach called JSM-DKSVD. It is able to capture and organise the rich spectral and spatial information into the learned dictionary, thus offering higher discriminative power for HSI classification tasks.

Experiment results show that, when used in conjunction with established SRC methods, the JSM-DKSVD-trained dictionary can significantly improve the SRC methods' classification performance, and can also outperform state-of-the-art dictionary learning methods for HSI classification.

The main contribution of this research is that we introduce the structure information around a limited number of training pixels into the dictionary learning for HSI, establish a new discriminative optimisation function to jointly model the enriched information, and develop a JSM-constrained D-KSVD algorithm to solve the optimisation problem and produce a desired discriminative dictionary.

3.2 Joint sparse models for HSI classification

The sparse model (SM) and the joint sparse model (JSM) are reviewed in Chapter 2. The work of this chapter mainly focuses on the JSM which is detailed in section 2.2.2, and the notations are aligned with section 2.2.2.

3.3 Discriminative dictionary learning algorithms

JSM-based classification methods introduced in Chapter 2 have achieved improved classification performance over the traditional (individual) sparse model, but most of these methods leave the dictionary $\mathbf{D}$ simply as a stack of raw labelled pixels [4, 25, 27]. On the other hand, the focus of this work is on the learning of $\mathbf{D}$. Specifically, we propose to develop a new dictionary learning algorithm, termed JSM-constrained discriminative K-SVD (JSM-DKSVD), to incorporate both the spectral and spatial information into dictionary learning and to improve the performance of HSI classification in the end.

3.3.1 K-SVD

In K-SVD [32], signals are also represented by their sparse coefficients. It aims to learn a dictionary $\mathbf{D}$ with unit atoms (bases), which minimises the reconstruction error:

$$\begin{aligned} \{\hat{\mathbf{D}}, \hat{\mathbf{A}}^{train}\} = & \underset{\mathbf{D},\mathbf{A}^{train}}{\operatorname{argmin}} \left\| \mathbf{X}^{train} - \mathbf{D}\mathbf{A}^{train} \right\|_F^2 , \\ & s.t. \ \left\| \alpha_p^{train} \right\|_0 \leq L_D, \ p = 1, \ldots, P , \end{aligned} \tag{3.1}$$

where $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_{N_D}] \in \mathbb{R}^{B \times N_D}$ is a dictionary with N_D atoms to be learned; $\mathbf{X}^{train} = [\mathbf{x}_1^{train}, \ldots, \mathbf{x}_P^{train}] \in \mathbb{R}^{B \times P}$ is a training sample set of P training samples; $\mathbf{A}^{train} = [\alpha_1^{train}, \ldots, \alpha_P^{train}] \in \mathbb{R}^{N_D \times P}$ is the corresponding sparse coefficient matrix of $\mathbf{X}^{train}$; and L_D ($L_D \ll N_D$) is upper bound of the sparsity level of the model. K-SVD consists of a sparse coding stage and a dictionary updating stage: it first solves (3.1) with $\mathbf{D}$ fixed to compute sparse coefficient matrix $\mathbf{A}^{train}$ by the OMP algorithm. Once $\mathbf{A}^{train}$ is obtained, a second stage is performed to update each dictionary atom by SVD one at a time, fixing all other atoms. The two stages are carried out iteratively till certain stopping criteria are met.

3.3.2 Discriminative KSVD (D-KSVD)

The discriminative K-SVD [6] is proposed to incorporate classification error into the optimisation problem (3.1), allowing a linear classifier and a dictionary with discriminative power to be learned at the same time.

Specifically, a classification constraint with loss function $\left\|\mathbf{H}^{train} - \mathbf{W}\mathbf{A}^{train}\right\|_F^2 + \beta \left\|\mathbf{W}\right\|_F^2$ is considered, where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{N_D}] \in \mathbb{R}^{M \times N_D}$ is an M-classes linear classifier in the atom space, $\mathbf{H}^{train} = [\mathbf{h}_1^{train}, \ldots, \mathbf{h}_P^{train}] \in \mathbb{R}^{M \times P}$ is the class matrix of P training pixels in $\mathbf{X}^{train}$, and $\left\|\mathbf{W}\right\|_F^2$ is the regularisation term. Each class vector $\mathbf{h}_p^{train} = [0, 0, \ldots, 1, \ldots, 0, 0]^T \in \mathbb{R}^M$ corresponds to the labelling of one training sample $\mathbf{x}_p^{train}$ and the non-zero position in $\mathbf{h}_p^{train}$ represents the class of $\mathbf{x}_p^{train}$. The dictionary $\mathbf{D}$ and the linear classifier $\mathbf{W}$ are jointly learned by solving the following optimisation problem:

$$\begin{aligned} \{\hat{\mathbf{D}}, \hat{\mathbf{A}}^{train}, \hat{\mathbf{W}}\} = \underset{\mathbf{D}, \mathbf{A}^{train}, \mathbf{W}}{\operatorname{argmin}} & \{\left\|\mathbf{X}^{train} - \mathbf{D}\mathbf{A}^{train}\right\|_F^2 \\ & + \gamma \left\|\mathbf{H}^{train} - \mathbf{W}\mathbf{A}^{train}\right\|_F^2 + \beta \left\|\mathbf{W}\right\|_F^2\} , \\ & s.t. \ \left\|\alpha_p^{train}\right\|_0 \leq L_D, \ p = 1, \ldots, P , \end{aligned} \tag{3.2}$$

where γ and β control the relative contributions of the corresponding terms. As described in [6], problem (3.2) can be rewritten as

$$\begin{aligned}
&\{\hat{\mathbf{D}},\hat{\mathbf{A}}^{train},\hat{\mathbf{W}}\} = \\
&\underset{\mathbf{D},\mathbf{A}^{train},\mathbf{W}}{\operatorname{argmin}} \{ \left\| \begin{pmatrix} \mathbf{X}^{train} \\ \sqrt{\gamma}\mathbf{H}^{train} \end{pmatrix} - \begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} \mathbf{A}^{train} \right\|_F^2 \\
&\quad + \beta \left\|\mathbf{W}\right\|_F^2 \} , \\
&\quad s.t. \ \left\|\alpha_p^{train}\right\|_0 \leq L_D, \ p = 1,\ldots,P .
\end{aligned} \tag{3.3}$$

Following [6], the constraint $\beta \left\|\mathbf{W}\right\|_F^2$ is omitted because during the K-SVD process the joint matrix $\begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix}$ is always column-wise normalised, i.e. the l_2-norm constraint is implicitly enforced. Now we use the following notation:

$$\mathbf{X}^* = \begin{pmatrix} \mathbf{X}^{train} \\ \sqrt{\gamma}\mathbf{H}^{train} \end{pmatrix} , \ \mathbf{D}^* = \begin{pmatrix} \mathbf{D} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} ; \tag{3.4}$$

and problem (3.3) is approximated by the following optimisation problem:

$$\begin{aligned}
\{\hat{\mathbf{D}}^*,\hat{\mathbf{A}}^{train}\} &= \underset{\mathbf{D}^*,\mathbf{A}^{train}}{\operatorname{argmin}} \left\| \mathbf{X}^* - \mathbf{D}^*\mathbf{A}^{train} \right\|_F^2 , \\
s.t. \ \left\|\alpha_p^{train}\right\|_0 &\leq L_D , \ p = 1,\ldots,P ,
\end{aligned} \tag{3.5}$$

which can then be solved by the K-SVD algorithm [32].

We note that the obtained matrix $\hat{\mathbf{D}}^*$ from K-SVD is not the actual dictionary we are looking for. To extract the actual dictionary $\mathbf{D}'$ and the classifier $\mathbf{W}'$, a final normalisation is needed. The dictionary $\mathbf{D}'$ is to be extracted from $\hat{\mathbf{D}}^*$ and column-wise normalised to have unit l_2-norm; the rest of the matrix $\hat{\mathbf{D}}^*$, namely classifier $\mathbf{W}'$, is scaled by using the same normalisation constants accordingly:

$$\begin{aligned}
\mathbf{D}' &= \left[\frac{\mathbf{d}_1}{\left\|\mathbf{d}_1\right\|_2}, \frac{\mathbf{d}_2}{\left\|\mathbf{d}_2\right\|_2}, \ldots, \frac{\mathbf{d}_{N_D}}{\left\|\mathbf{d}_{N_D}\right\|_2} \right] , \\
\mathbf{W}' &= \left[\frac{\mathbf{w}_1}{\left\|\mathbf{d}_1\right\|_2}, \frac{\mathbf{w}_2}{\left\|\mathbf{d}_2\right\|_2}, \ldots, \frac{\mathbf{w}_{N_D}}{\left\|\mathbf{d}_{N_D}\right\|_2} \right] ,
\end{aligned} \tag{3.6}$$

where $\mathbf{d}_k$ and $\mathbf{w}_k$ denote the k-th column of $\mathbf{D}$ and $\mathbf{W}$, respectively.

3.3.3 Classification approach

Given the dictionary $\mathbf{D}'$ and the linear classifier $\mathbf{W}'$, the sparse coefficient vector α^{test} of a test HSI pixel $\mathbf{x}^{test}$ is computed by solving the following problem:

$$\begin{aligned}\hat{\alpha}^{test} &= \underset{\alpha^{test}}{\operatorname{argmin}} \left\| \mathbf{x}^{test} - \mathbf{D}'\alpha^{test} \right\|_2^2 , \\ s.t. &\ \left\| \alpha^{test} \right\|_0 \leq L_C .\end{aligned} \tag{3.7}$$

By applying the linear classifier $\mathbf{W}'$ to $\hat{\alpha}^{test}$, the class label vector $\mathbf{h}^{test} = [h_1^{test}, \ldots, h_M^{test}]^T$ of $\mathbf{x}^{test}$ is obtained as

$$\hat{\mathbf{h}}^{test} = \mathbf{W}'\hat{\alpha}^{test} , \tag{3.8}$$

and the class label of $\mathbf{x}^{test}$ is determined by the position of the maximum value within $\hat{\mathbf{h}}^{test}$:

$$class(\mathbf{x}^{test}) = \underset{m=1,\ldots,M}{\operatorname{argmax}} \hat{h}_m^{test} . \tag{3.9}$$

3.4 JSM-DKSVD

Dictionary learning by K-SVD and D-KSVD only considers spectral signatures of the HSI pixels. Recent developments in JSM-related algorithms show promising results of using not only spectral but also spatial structure information in the representation of pixels. Inspired by this progress, we propose to incorporate the HSI structure information into the dictionary learning process and extend D-KSVD to HSI classification. Specifically, we enforce the assumption of local smoothness of images as well as sparsity of the representations of training HSI pixels into dictionary learning. We name this new dictionary learning approach as JSM-DKSVD.

3.4.1 Motivation of JSM-DKSVD

The core idea of JSM-DKSVD is to embed the structure information into the representation of dictionary training pixels by joint modelling. The sparse coefficients of a pixel are determined jointly with those in its local neighbourhood, which is a collection of pixels located in a small window centred on the pixel in question.

Therefore the training set $\mathbf{X}^{train} = [\mathbf{x}_1^{train}, \ldots, \mathbf{x}_P^{train}] \in \mathbb{R}^{B \times P}$ is extended as follows:

$$\mathbf{X}^{JSM} = [X_1^{JSM}, \ldots, X_P^{JSM}] \in \mathbb{R}^{B \times (T_D \times P)} \ , \tag{3.10}$$

where $X_p^{JSM} \in \mathbb{R}^{B \times T_D}$, $p = 1, \ldots, P$, denotes a small window consisting of T_D pixels and centred on the training pixel $\mathbf{x}_p^{train}$. Each of these neighbourhoods X_p^{JSM} is now as a whole to be jointly modelled, replacing the pixel $\mathbf{x}_p^{train}$ in dictionary learning. Note that, although the training sample size is effectively increased from P to $T_D \times P$, as JSM-DKSVD is working in a semi-supervised fashion, we only need P labelled training pixels, which are those central pixels; that is, our JSM-DKSVD method does not require more labelled training samples than K-SVD and D-KSVD.

The spectral and spatial structure information of a training pixel is therefore exploited by enforcing local smoothness of natural signals, i.e. nearby pixels share a common pattern. In our case, a certain degree of similarity is enforced on the sparse representation patterns of the neighbouring pixels. This forms a new constraint, and will be reflected by expanding the class matrix $\mathbf{H}^{train}$ in D-KSVD correspondingly to a larger matrix $\mathbf{H}^{JSM}$.

If, for example, the central pixel $\mathbf{x}_p^{train}$ in the neighbourhood X_p^{JSM} is labelled class #4 out of five classes, its class vector $\mathbf{h}_p$ in D-KSVD is

$$\mathbf{h}_p = [0,0,0,1,0]^T \ , \tag{3.11}$$

where the non-zero position is at the 4th element. In our JSM-DKSVD, by assuming that the neighbouring pixels share the same class label, the class matrix of the 3×3 window centred on $\mathbf{x}_p$ is as follows:

$$H_p^{JSM} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{5 \times 9} \ , \tag{3.12}$$

where the class vector of each pixel in the window shares the same non-zeros row, i.e. the 4th row of the class matrix H_p^{JSM}. Naturally, by concatenating the class matrices of all training pixels $\mathbf{X}^{JSM}$, the overall class matrix $\mathbf{H}^{JSM}$ is

$$\mathbf{H}^{JSM} = [H_1^{JSM}, \ldots, H_P^{JSM}] \in \mathbb{R}^{M \times (T_D \times P)} \ . \tag{3.13}$$

3.4.2 Formulation of JSM-DKSVD

In the proposed JSM-DKSVD, signals in a small neighbourhood are jointly represented by a common sparsity pattern, as in JSM. Meanwhile, a classification constraint with a new class matrix $\mathbf{H}^{JSM}$ is reconstructed, leading to the following optimisation problem:

$$\begin{aligned} &\{\hat{\mathbf{D}}^{JSM}, \hat{\mathbf{A}}^{JSM}, \hat{\mathbf{W}}\} = \\ &\qquad \underset{\mathbf{D}^{JSM}, \mathbf{A}^{JSM}, \mathbf{W}}{\operatorname{argmin}} \ \{ \left\| \mathbf{X}^{JSM} - \mathbf{D}^{JSM}\mathbf{A}^{JSM} \right\|_F^2 \\ &\qquad + \gamma \left\| \mathbf{H}^{JSM} - \mathbf{W}\mathbf{A}^{JSM} \right\|_F^2 + \beta \left\| \mathbf{W} \right\|_F^2 \} \ , \\ &\qquad s.t. \ \left\| A_p^{JSM} \right\|_{row,0} \leq L_D, \ \ p = 1, \ldots, P \ , \end{aligned} \tag{3.14}$$

where $\mathbf{X}^{JSM}$ and $\mathbf{H}^{JSM}$ are defined in (3.10) and (3.13), respectively; $A_p^{JSM} = [\alpha_{p,1}^{JSM}, \ldots, \alpha_{p,T_D}^{JSM}] \in \mathbb{R}^{N_D \times T_D}$ is the corresponding joint sparse coefficient matrix of a small window X_p^{JSM} as defined in (3.10), and therefore $\mathbf{A}^{JSM} = [A_1^{JSM}, \ldots, A_P^{JSM}] \in \mathbb{R}^{N_D \times (T_D \times P)}$ is the corresponding sparse coefficient matrix of $\mathbf{X}^{JSM}$; $\mathbf{D}^{JSM} \in \mathbb{R}^{B \times N_D}$ and $\mathbf{W} \in \mathbb{R}^{M \times N_D}$ are the dictionary and classifier to be learned by JSM-DKSVD. This problem can be solved by the K-SVD algorithm. Again, due to the column-wise normalisation through the K-SVD process, the constraint $\beta \left\| \mathbf{W} \right\|_F^2$ can be omitted to simplify the problem.

Similarly to (3.4) and (3.5), the optimisation problem (3.14) can be rewritten as

$$\begin{aligned}
&\{\hat{\mathbf{D}}^{JSM*}, \hat{\mathbf{A}}^{JSM}\} = \\
&\quad \underset{\mathbf{D}^{JSM*}, \mathbf{A}^{JSM}}{\operatorname{argmin}} \left\| \mathbf{X}^{JSM*} - \mathbf{D}^{JSM*}\mathbf{A}^{JSM} \right\|_F^2 , \\
&\quad s.t. \ \left\| A_p^{JSM} \right\|_{row,0} \leq L_D, \ \ p = 1, \dots, P ,
\end{aligned} \tag{3.15}$$

where

$$\mathbf{X}^{JSM*} = \begin{pmatrix} \mathbf{X}^{JSM} \\ \sqrt{\gamma}\mathbf{H}^{JSM} \end{pmatrix}, \text{ and } \mathbf{D}^{JSM*} = \begin{pmatrix} \mathbf{D}^{JSM} \\ \sqrt{\gamma}\mathbf{W} \end{pmatrix} . \tag{3.16}$$

Therefore, $\mathbf{X}^{JSM*}$ is a $(B+M) \times (T_D \times P)$ matrix and $\mathbf{D}^{JSM*}$ is a $(B+M) \times N_D$ matrix.

3.4.3 Algorithm of JSM-DKSVD

The objective function (3.14) of JSM-DKSVD can be solved by adopting the original K-SVD algorithm in [32], more specifically, by adopting the iterative updating process for $\mathbf{D}^{JSM*}$ and $\mathbf{A}^{JSM*}$.

3.4.3.1 Initialisation

It is required that the dictionary $\mathbf{D}^{JSM}$ and the classifier $\mathbf{W}$ are given initial values to enable the iterative updating process to follow. Their initial values can be as simple as randomised matrices; in this work we follow the initialisation process of D-KSVD [6] which is explained as follows.

The initial dictionary matrix is denoted by $\mathbf{D}^{(0)}$. As with in [6], $\mathbf{D}^{(0)}$ should have l_2-normalised columns. Given the number of atoms N_D, $\mathbf{D}^{(0)}$ is designed to be a $B \times N_D$ matrix, and it can be initialised by the original K-SVD algorithm with only a couple of iterations.

As a result, the coefficient matrix of $\mathbf{X}^{JSM}$ for initialisation, denoted by $\mathbf{A}^{(0)}$, is computed by solving the first objective term of (3.14):

$$\mathbf{A}^{(0)} = (\mathbf{D}^{(0)^T}\mathbf{D}^{(0)})^{-1}\mathbf{D}^{(0)^T}\mathbf{X}^{JSM} , \tag{3.17}$$

where $\mathbf{A}^{(0)}$ is an $N_D \times (T_D \times P)$ matrix.

The initial classifier, denoted by $\mathbf{W}^{(0)}$, is computed by solving the problem of $\underset{\mathbf{W}^{(0)}}{\operatorname{argmin}}\{\left\|\mathbf{H}^{JSM}-\mathbf{W}^{(0)}\mathbf{A}^{(0)}\right\|_F^2+\left\|\mathbf{W}^{(0)}\right\|_F^2\}$:

$$\mathbf{W}^{(0)}=\left((\mathbf{A}^{(0)}\mathbf{A}^{(0)T}+\mathbf{I})^{-1}\mathbf{A}^{(0)}\mathbf{H}^{JSMT}\right)^T \ , \tag{3.18}$$

where $\mathbf{W}^{(0)}$ is an $M \times N_D$ matrix.

After initialisation, we compose the objective function (3.15) with $\mathbf{X}^{JSM*} \in \mathbb{R}^{(B+M)\times(T_D\times P)}$ and $\mathbf{D}^{JSM*} \in \mathbb{R}^{(B+M)\times N_D}$, and the iterative updating process of $\mathbf{D}^{JSM*}$ and $\mathbf{A}^{JSM*}$ can start.

3.4.3.2 Iterative updating - sparse coding stage

Fixing the dictionary $\mathbf{D}^{JSM*}$, we compute the joint sparse coefficient matrix $A_p^{JSM} \in \mathbb{R}^{N_D\times T_D}$ for each training window $X_p^{JSM*} \in \mathbb{R}^{(B+M)\times T_D}$, where $p=1,\ldots,P$, by approximating the following solution:

$$\begin{aligned}\hat{A}_p^{JSM} = & \underset{A_p^{JSM}}{\operatorname{argmin}}\left\|X_p^{JSM*}-\mathbf{D}^{JSM*}A_p^{JSM}\right\|_F^2 \ ,\\ & s.t. \ \left\|A_p^{JSM}\right\|_{row,0} \le L_D \ ,\end{aligned} \tag{3.19}$$

which can be solved by the SOMP algorithm [4, 8, 39]. Then the sparse coefficient matrix $\mathbf{A}^{JSM}$ of all training window $\mathbf{X}^{JSM*}$ (3.10) is concatenated as

$$\mathbf{A}^{JSM}=[A_1^{JSM},\ldots,A_P^{JSM}]=[\alpha_1^{JSM},\ldots,\alpha_{T_D\times P}^{JSM}] \ . \tag{3.20}$$

3.4.3.3 Iterative updating - dictionary updating stage

Following the similar idea of SVD in [32], the dictionary is updated atom by atom.

In the jth iteration, for the kth atom $\mathbf{d}_k^{JSM*} \in \mathbb{R}^{B+M}$ in the dictionary $\mathbf{D}^{JSM*(j-1)}$, where $j=1,\ldots,J$ and $k=1,\ldots,N_D$; $\mathbf{D}^{JSM*(j-1)}$ is the dictionary obtained from the previous iteration $j-1$, the atom $\mathbf{d}_k^{JSM*}$ is updated to a new one, denoted by $\tilde{\mathbf{d}}_k^{JSM*}$, by the following steps:

a. define a group of the instances that use atom $\mathbf{d}_k^{JSM*}$,

$$\omega_k = \{p | 1 \leqslant p \leqslant T_D \times P, \mathbf{A}^{JSM}(k,p) \neq 0\} \ , \tag{3.21}$$

where $\mathbf{A}^{JSM}(k,p)$ denotes the kth row and pth column of $\mathbf{A}^{JSM}$;

b. compute the overall representation error $\mathbf{E}_k$ without using the atom $\mathbf{d}_k^{JSM*}$,

$$\mathbf{E}_k = \mathbf{X}^{JSM*} - \sum_{i \neq k} \mathbf{d}_i^{JSM*} \mathbf{A}^{JSM}(i, \cdot) \ , \tag{3.22}$$

where $\mathbf{A}^{JSM}(i, \cdot)$ denotes the ith row of $\mathbf{A}^{JSM}$ and $i = 1, \ldots, N_D$;

c. restrict $\mathbf{E_k}$ by choosing only the column corresponding to ω_k, and obtain $\mathbf{E}_k^R$:

$$\mathbf{E}_k^R = \mathbf{E_k}(\cdot, \omega_k) \ , \tag{3.23}$$

where $\mathbf{E_k}(\cdot, \omega_k)$ denotes the columns of $\mathbf{E_k}$ corresponding to ω_k;

d. apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\Delta\mathbf{V}^T$. The updated atom $\tilde{\mathbf{d}}_k^{JSM*}$ and its corresponding sparse coefficients in the updated coefficient matrix $\tilde{\mathbf{A}}^{JSM*}$ are solved by

$$\begin{aligned} \tilde{\mathbf{d}}_k^{JSM*} &= \mathbf{U}(\cdot, 1) \ , \\ \tilde{\mathbf{A}}^{JSM*}(k, \omega_k) &= \Delta(1,1)\mathbf{V}(\cdot, 1) \ , \end{aligned} \tag{3.24}$$

where $\mathbf{U}(\cdot, 1)$ and $\mathbf{V}(\cdot, 1)$ denotes the first column of $\mathbf{U}$ and the first column of $\mathbf{V}$, respectively.

After J iterations, the desired dictionary $\mathbf{D}'$ and the classifier $\mathbf{W}'$ learned by JSM-DKSVD should also be re-normalised as in (3.6).

Details of the proposed JSM-DKSVD are summarised in Algorithm 1.

3.4.4 Classification approach of JSM-DKSVD

Same as the D-KSVD, the dictionary $\mathbf{D}'$ and the classifier $\mathbf{W}'$ learned by JSM-DKSVD can be used together with many established HSI classification methods. By

Algorithm 1 JSM-DKSVD algorithm to solve (3.15).

```
Input:
    Training HSI pixels $\mathbf{X}^{train} \in \mathbb{R}^{B \times P}$.
    Window size $T_D$ for training.
    Control parameter $\gamma$.
    Sparsity level $L_D$.
    Number of atoms $N_D$ of the dictionary to be learned.
    Maximum iteration number $J$.
Output: $\mathbf{D}'$ , $\mathbf{W}'$.
    Initialisation:
        • Generate training sample set $\mathbf{X}^{JSM}$ by (3.10).
        • Compose class matrix $\mathbf{H}^{JSM}$ by (3.13).
        • Initialise dictionary matrix $\mathbf{D}^{(0)}$ with $l_2$-normalised columns.
        • Compute coefficient matrix $\mathbf{A}^{(0)}$ by (3.17).
        • Initialise classifier $\mathbf{W}^{(0)}$ by (3.18).
        • Compose $\mathbf{X}^{JSM*}$ and $\mathbf{D}^{JSM*}$ by (3.16).
    while $j \leqslant J$ do
        Sparse coding stage:
        Compute the sparse coefficient matrix $A_p^{JSM}$ for each training windows $X_p^{JSM*}$ by (3.19).
        Concatenate the sparse coefficient matrix $\mathbf{A}^{JSM}$ for all training windows $\mathbf{X}^{JSM*}$ by (3.20).
        Dictionary updating stage:
        for $k$ = 1:$N_D$ in $\mathbf{D}^{JSM*(j-1)}$ do
            Define the group of instances that use atom $\mathbf{d}_k^{JSM*}$ by (3.21).
            Compute the overall representation error $\mathbf{E}_k$ by (3.22).
            Restrict $\mathbf{E}_k$ to $\mathbf{E}_k^R$ by (3.23).
            Apply SVD decomposition to $\mathbf{E}_k^R$ and solve the updated atom $\tilde{\mathbf{d}}_k^{JSM*}$ and its corresponding sparse coefficients $\tilde{\mathbf{A}}^{JSM*}(k, \omega_k)$ by (3.24).
        end for
        Set $j = j + 1$.
    end while
    Compute the desired dictionary $\mathbf{D}'$ and classifier $\mathbf{W}'$ by (3.6).
```

embedding richer structure information from HSI in the dictionary and the classifier, the proposed JSM-DKSVD aims to improve the overall classification accuracy when used with these dictionary-based classification methods for HSI.

Specifically, when JSM-DKSVD is used in pair with the JSM-based SRC method, given a test matrix $\mathbf{X}^{test} = [\mathbf{x}_1^{test}, \ldots, \mathbf{x}_{T_C}^{test}]$ with T_C pixels centred on the test pixel $\mathbf{x}^{test}$, the JSM coefficient matrix $\mathbf{A}^{test} = [\alpha_1^{test}, \ldots, \alpha_{T_C}^{test}]$ is computed by solving the following problem:

$$\begin{aligned} \hat{\mathbf{A}}^{test} &= \underset{\mathbf{A}^{test}}{\operatorname{argmin}} \left\|\mathbf{X}^{test} - \mathbf{D}'\mathbf{A}^{test}\right\|_F^2 \ , \\ &s.t. \ \left\|\mathbf{A}^{test}\right\|_{row,0} \leq L_C \ . \end{aligned} \tag{3.25}$$

Then, the classifier $\mathbf{W}'$ is applied to $\hat{\mathbf{A}}^{test}$ to create the estimated class label matrix $\hat{\mathbf{H}}^{test}$ for $\mathbf{X}^{test}$:

$$\hat{\mathbf{H}}^{test} = \mathbf{W}'\hat{\mathbf{A}}^{test} \ . \tag{3.26}$$

Finally, each row of $\hat{\mathbf{H}}^{test} \in \mathbb{R}^{M \times T_C}$ is summed together as a new class label vector $\hat{\mathbf{h}}^{test} \in \mathbb{R}^M$, and the class label of the central test pixel is determined by (3.9).

3.5 Experimental studies

The experiments are carried out on two real hyperspectral datasets: the AVIRIS Indian Pines dataset and the ROSIS University of Paiva dataset, both of which are publicly available [40]. We evaluate the proposed JSM-DKSVD and compare the learned dictionary with two other types of dictionaries. The first comparison is against the dictionary constructed by original labelled training pixels, denoted by $\mathbf{D}^{raw}$, such as in [4, 25, 27]. The second comparison is against the direct application of D-KSVD [6]. Dictionaries acquired from $\mathbf{D}^{raw}$, D-KSVD and the proposed JSM-DKSVD are used with three different SRC methods: 1) SM (referred to as OMP), 2) JSM [4] (referred to as SOMP), and 3) NLW [25].

We employ three standard performance measures for HSI classification: the overall accuracy (OA), the average accuracy (AA) and kappa coefficient κ [41]. The overall accuracy is defined as the ratio of correctly-classified test pixels over all classes; the average accuracy is defined as the average value of the M accuracies of individual classes, where M is the total number of classes; and the kappa coefficient κ is defined as the percentage of classified test pixels corrected by the number of agreements that would be expected purely by chance. The OA, AA and κ are defined as follows:

$$OA = \frac{N_{corr}}{N_{test}},\ AA = \frac{1}{M}\sum_{i=1}^{M}\frac{N_i^{corr}}{N_i^{class}},\ \kappa = \frac{OA - p_e}{1 - p_e}, \tag{3.27}$$

where N_{corr} is the number of the correctly-classified test pixels, N_{test} is the total number of test pixels; N_i^{corr} is the number of the correctly-classified test pixels of class i, N_i^{class} is the total number of test pixels of class i; and $p_e = \sum_{i=1}^{M}(P_i \times P_i^t)$, in which P_i is the ratio of data assigned to class i by the classifier and P_i^t is the ratio of data that belong to class i.

The SPAMS toolbox [39] is used to execute the sparse recovery process, i.e. OMP and SOMP. MATLAB codes from [6] are used to perform the K-SVD and D-KSVD algorithms, and MATLAB codes from [25] are used to perform the NLW algorithm.

3.5.1 Parameter settings

The parameters involved in the whole evaluation process include those for both the SRC methods and the dictionary learning methods. For the SRC methods, the parameters include the sparsity level L_C and the window size T_C for SOMP and NLW. For the dictionary learning methods, the parameters include the sparsity level L_D, the number of atoms N_D, the regularisation parameter γ, the iteration number J, and finally the window size T_D for the proposed JSM-DKSVD method. It is too costly to cross-validate through the entire design space. To simplify the problem, we break it down into two steps:

- When the three SRC methods (OMP, SOMP and NLW) use $\mathbf{D}^{raw}$, we perform the leave-one-out-cross-validation (LOOCV) to tune their parameters L_C and T_C. Then the parameters of the three SRC methods are fixed and decoupled from dictionary learning, providing a relatively fair testing platform for the dictionary learning methods.

- For the D-KSVD and JSM-DKSVD dictionary learning methods which produce dictionaries that are independent of and universally applicable to different SRC methods, we define their parameters in the following way. Parameter N_D by the nature cannot be tuned by cross-validation. Therefore, we set N_D to be the maximum possible number of atoms, which is dataset-dependent (see details in the following sections). Empirically and for simplicity, the regularisation parameter γ is set to be 1 and the iteration number J is set to be 30. The sparsity level L_D for the matching pursuit algorithms is set to be 5 and 30 respectively for DKSVD and JSM-DKSVD due to the difference between OMP and SOMP. Finally, we evaluate JSM-DKSVD with two training window sizes, 3 × 3 and 5 × 5, for illustrative purposes.

3.5.2 AVIRIS dataset: Indian Pines

Table 3.1: The Indian Pines dataset: Ground-truth labels, class material, the training set and the test set. The middle two columns are for the case of 957 training pixels (9% of all pixels) and 9,409 test pixels; the rightmost two columns are for the case of 524 training pixels (5% of all pixels) and 9,842 test pixels.

Class	Material	Training	Test	Training	Test
1	Alfalfa	5	49	3	51
2	Corn-notill	132	1302	72	1362
3	Corn-mintill	77	757	42	792
4	Corn	22	212	12	222
5	Grass-pasture	46	451	25	472
6	Grass-trees	69	678	38	709
7	Grass-pasture-mowed	3	23	2	24
8	Hay-windrowed	45	444	25	464
9	Oats	2	18	1	19
10	Soybean-notill	89	879	49	919
11	Soybean-mintill	227	2241	124	2344
12	Soybean-clean	57	557	31	583
13	Wheat	20	192	11	201
14	Woods	119	1175	65	122
15	Buildings-grass-trees-drives	35	345	19	361
16	Stone-steel-towers	9	86	5	90
Total		957	9409	524	9842

The AVIRIS Indian Pines dataset consists of 145 × 145 pixels from 224 spectral bands with sixteen ground-truth labels. Similarly to [4] and [25], we use its 200

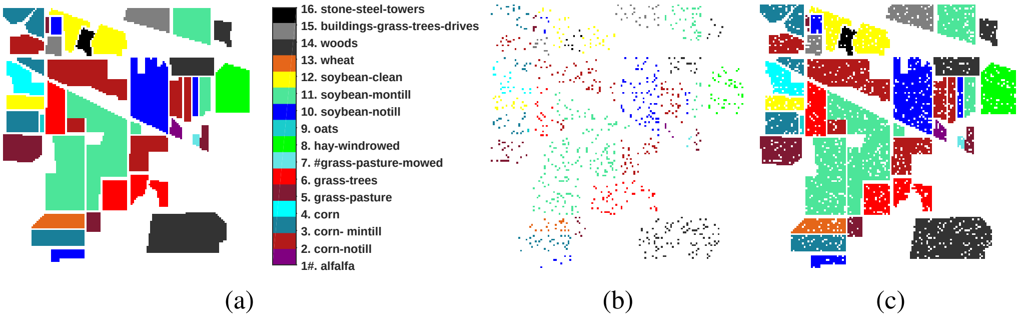


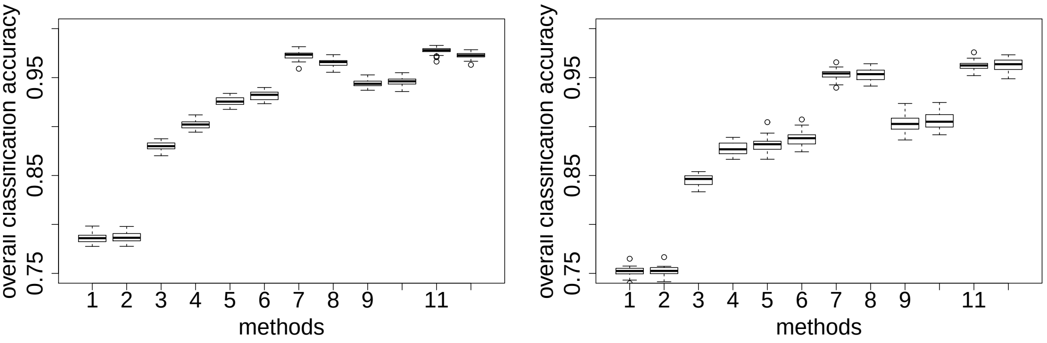
Figure 3.1: The Indian Pines dataset with 9% pixels randomly chosen for training: (a) ground-truth labels; (b) training set; (c) test set.

bands after removing the water absorption bands. Following [25], we first randomly choose 957 labelled pixels (9.23% of all pixels) for training, i.e. $\mathbf{X}^{train} \in \mathbb{R}^{200\times957}$. The rest pixels are used for testing, i.e. $\mathbf{X}^{test} \in \mathbb{R}^{200\times9409}$. A summary of the numbers of training and test pixels for individual classes is given in the middle two columns in Table 3.1. The sixteen ground-truth classes, the training set as well as the test set are shown in Figures. 3.1(a)-3.1(c).

For the three SRC methods, OMP, SOMP and NLW using $\mathbf{D}^{raw}$, the optimal parameters obtained by LOOCV are $L_C = 5$ for OMP, $L_C = 30$ and $T_C = 7\times7$ for SOMP and $L_C = 30$ and $T_C = 9\times9$ for NLW.

Regarding the number of atoms, we set $N_D = 957$ for D-KSVD. For the JSM-DKSVD dictionary learning method, due to the possible overlapping of the extended neighbourhoods, its training set $\mathbf{X}^{JSM}$, which is the extended $\mathbf{X}^{train}$, may not be full rank and as a result the K-SVD algorithm cannot be executed. The maximum possible number of atoms for JSM-DKSVD is therefore defined to be the maximum unique columns of $\mathbf{X}^{JSM}$. For the training window $T_D = 3\times3$, the unique number of atoms is 5,145; and for $T_D = 5\times5$, the unique number of atoms is 8,764. Therefore we set $N_D = 5,145$ and $N_D = 8,764$ under $T_D = 3\times3$ and $T_D = 5\times5$, respectively.

To have a reliable evaluation and fair comparison, we repeat the experiments for 20 times under these parameter settings through performing 20 random training-test splits. For each of the 12 combinations of four dictionary learning schemes and three SRC methods with their optimal parameters, all of its 20 overall classification



(a) (b)

Figure 3.2: Boxplots of the overall classification accuracies for the Indian Pines dataset, for 12 combinations indexed by the horizontal axis: (1) $\mathbf{D}^{raw}$-OMP, (2) DKSVD-OMP, (3) JSM-DKSVD-OMP under $T_D = 3 \times 3$, (4) JSM-DKSVD-OMP under $T_D = 5 \times 5$, (5) $\mathbf{D}^{raw}$-SOMP, (6) DKSVD-SOMP, (7) JSM-DKSVD-SOMP under $T_D = 3 \times 3$, (8) JSM-DKSVD-SOMP under $T_D = 5 \times 5$, (9) $\mathbf{D}^{raw}$-NLW, (10) DKSVD-NLW, (11) JSM-DKSVD-NLW under $T_D = 3 \times 3$, and (12) JSM-DKSVD-NLW under $T_D = 5 \times 5$. Each boxplot is constructed from the results of 20 experiments, with panel (a) for the case that 9% pixels are randomly chosen to train the dictionary; and panel (b) for the case that 5% pixels are randomly chosen to train the dictionary.

accuracies are recorded and box-plotted in Figure 3.2(a). Moreover, for illustrative purposes, the classification results for one experiment randomly selected from the 20 experiments are given in Table 3.2 and depicted in Figures. 3.3(a)-3.3(l), respectively.

It can be observed that, firstly, the D-KSVD method does not improve the classification performance significantly, compared with those simply using $\mathbf{D}^{raw}$ for HSI classification. Secondly, in contrast to D-KSVD, JSM-DKSVD is capable of producing a dictionary-classifier combination of much better performance than the other two dictionary learning methods in both cases of $T_D = 3 \times 3$ and $T_D = 5 \times 5$. In Table 3.2, for OMP, the overall accuracy (OA) is improved the most, with an 11% (78.63% to 89.94 %) increase under $T_D = 3 \times 3$ and with a 14% (78.63% to 92.99%) increase under $T_D = 5 \times 5$. For SOMP and NLW, OAs are also improved, by around 4% (93.85% to 97.95% and 95.00% to 98.68%) under $T_D = 3 \times 3$. JSM-DKSVD combined with NLW reaches the highest accuracies, 98.68%.

To further demonstrate the benefit of using the JSM-DKSVD-trained dictio-

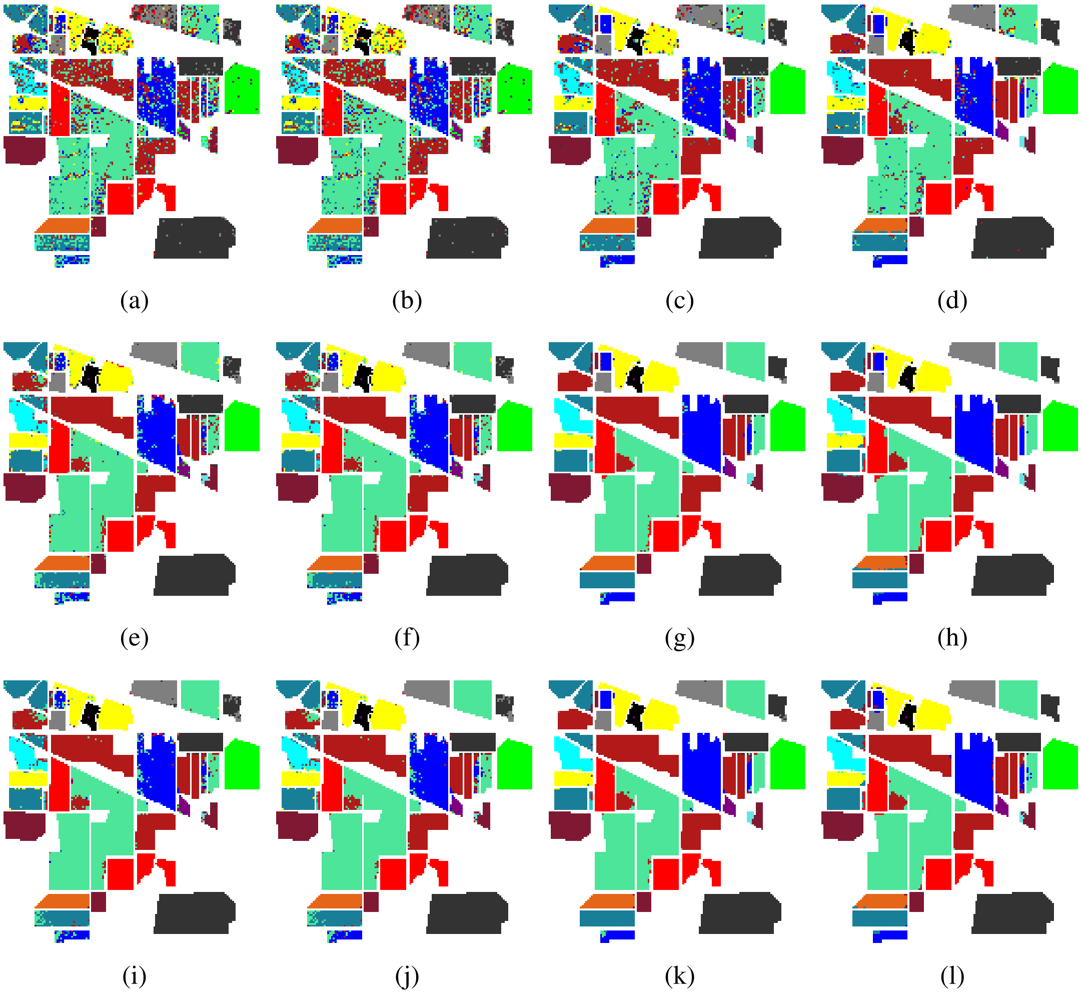


Figure 3.3: The classification maps of the Indian Pines dataset with 9% pixels randomly chosen for training: (a) $\mathbf{D}^{raw}$-OMP; (b) DKSVD-OMP (c) JSM-DKSVD-OMP (3×3); (d) JSM-DKSVD-SOMP (5×5); (e) $\mathbf{D}^{raw}$-SOMP; (f) DKSVD-SOMP (g) JSM-DKSVD-SOMP (3×3); (h) JSM-DKSVD-SOMP (5×5); (i) $\mathbf{D}^{raw}$-NLW; (j) DKSVD-NLW (k) JSM-DKSVD-NLW (3×3); (l) JSM-DKSVD-NLW (5×5).

Table 3.2: The classification accuracy (%) for the Indian Pines dataset with 957 training pixels (9% of all pixels) and 9409 test pixels, of four dictionary learning methods ($\mathbf{D}^{raw}$, DKSVD, JSM-DKSVD (3×3), and JSM-DKSVD (5×5)) for three SRC methods (OMP, SOMP, NLW). T_D: training window size; N_D: number of atoms; OA: overall accuracy (%); AA: average accuracy (%); κ: kappa coefficient.

	$\mathbf{D}^{raw}$			DKSVD			JSM-DKSVD					
T_D	N/A			N/A			3×3			5×5		
N_D	957			957			5145			8764		
	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW
1	61.22	79.59	93.88	63.27	77.55	91.84	87.76	91.84	97.96	95.92	100.00	97.96
2	70.35	92.93	94.62	70.28	92.32	94.01	87.17	97.85	98.31	96.08	98.62	98.77
3	65.65	85.73	87.71	65.39	86.79	88.38	86.92	97.62	98.55	91.94	98.02	97.49
4	54.25	89.15	85.38	53.77	89.62	84.43	73.11	99.06	98.58	84.43	97.17	97.64
5	94.24	96.45	98.45	94.46	96.67	98.67	98.23	99.78	99.78	98.89	98.23	100.00
6	94.99	99.26	99.85	94.99	99.85	99.85	97.20	99.56	99.71	97.79	98.53	99.56
7	56.52	56.52	30.43	56.52	56.52	26.09	91.30	82.61	86.96	78.26	73.91	43.48
8	96.62	100.00	100.00	96.40	100.00	100.00	99.32	100.00	100.00	99.77	100.00	100.00
9	55.56	16.67	16.67	55.56	16.67	16.67	77.78	44.44	50.00	61.11	0.00	5.56
10	63.82	77.82	82.48	63.82	79.64	82.82	85.89	93.86	96.36	86.58	94.43	94.65
11	79.43	95.67	98.39	79.70	96.97	98.62	86.93	98.30	99.06	89.60	97.72	98.88
12	72.53	93.00	97.13	72.35	96.23	97.85	86.54	97.49	99.10	91.92	95.69	97.13
13	99.48	98.96	99.48	99.48	100.00	99.48	98.96	99.48	98.96	85.94	84.90	95.31
14	94.47	97.19	97.45	94.47	97.36	97.36	97.62	99.15	99.23	98.30	99.40	100.00
15	57.39	97.68	97.97	57.68	98.55	99.71	85.22	100.00	99.71	93.91	99.71	98.84
16	82.56	98.84	98.84	83.72	98.84	98.84	89.53	93.02	97.67	81.40	83.72	87.21
OA	78.58	93.05	94.89	78.63	93.85	95.00	89.94	97.95	**98.68**	92.99	97.28	98.02
AA	74.94	85.97	86.17	75.12	86.47	85.91	89.34	93.38	**95.00**	89.49	88.75	88.28
κ	0.755	0.921	0.943	0.756	0.923	0.943	0.885	0.977	**0.985**	0.920	0.969	0.978

nary, the same test is performed again but with even fewer training pixels. For this test, only around 5% of the total pixels are chosen as training pixels, i.e. $\mathbf{X}^{train} \in \mathbb{R}^{200 \times 524}$, and the rest of the pixels are used for testing, i.e. $\mathbf{X}^{test} \in \mathbb{R}^{200 \times 9842}$. The summarised dataset is shown in the rightmost two columns in Table 3.1. The number of atoms N_D is set by the same process as that in the case of 9% training pixels, and the results are 2,919 under $T_D = 3 \times 3$ and 5,831 under $T_D = 5 \times 5$, respectively. Parameters L_C, T_C, L_D, T_D, γ and J are remained the same as those in 9% pixels for training.

We randomly split the dataset into training-test pairs for 20 times. All of the 20 overall classification accuracies are shown in Figure 3.2(b). The classification results for one experiment randomly selected from the 20 experiments are shown in Table 3.3, excluding the classification accuracies of individual classes and the classification maps to save space. Once again, JSM-DKSVD-trained dictionaries

Table 3.3: The overall classification accuracy (%) on the Indian Pines dataset with 524 training pixels (5% of all pixels) and 9842 test pixels. The notation is as for Table 3.2.

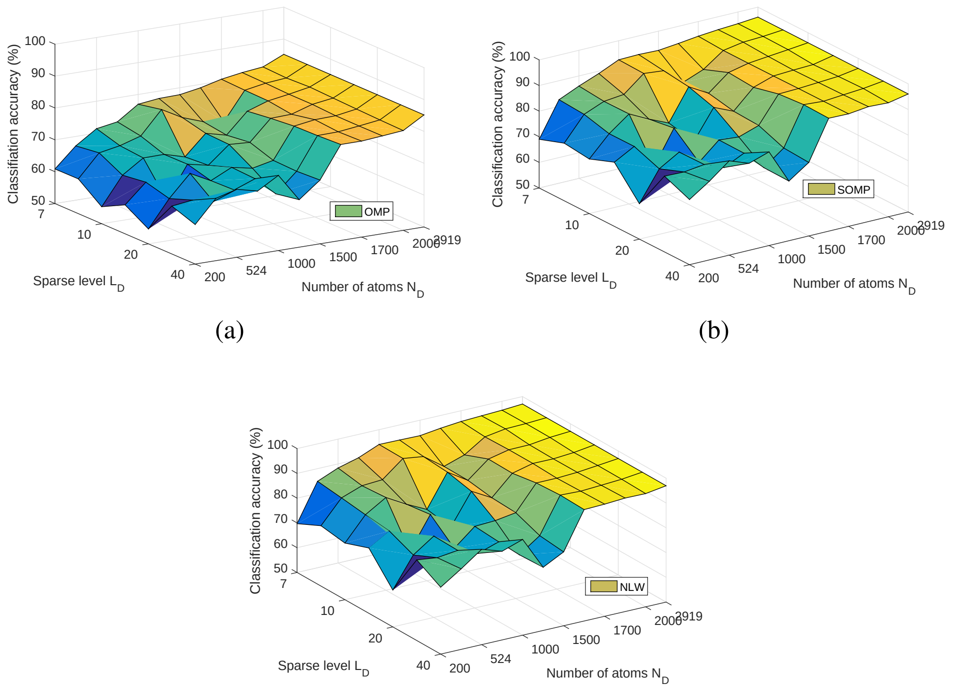
	$\mathbf{D}^{raw}$			DKSVD			JSM-DKSVD					
T_D	N/A			N/A			3×3			5×5		
N_D	524			524			2919			5831		
	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW
OA	75.75	90.46	92.35	75.67	91.15	92.48	85.17	96.08	96.97	88.41	95.74	**97.02**
AA	70.12	82.37	84.04	70.28	82.33	83.46	83.56	93.31	91.52	87.01	93.46	**93.47**
κ	0.723	0.891	0.913	0.722	0.899	0.914	0.831	0.955	0.966	0.868	0.952	**0.966**

are still capable of improving the performance of the reference SRC methods to a high standard, and can be much superior to the SRC methods with DKSVD-trained dictionaries.

We also compare our proposed JSM-DKSVD method with state-of-the-art method proposed in [37], which also incorporates the structure information into their dictionary learning processes. Referenced directly from [37], the test environment is slightly different in that 997 pixels (10.64% of all 16-classes pixels) are used for training (comparing to the 957 pixels in our case). Under the two similar test settings, our proposed JSM-DKSVD outperforms (98.68% as shown in Table 3.2) the best performance in [37] which is 94.20%. It is worth noting though: the method proposed in [37] aims to train compact dictionaries and therefore is still expected to have an edge on the computational cost.

The two parameters L_D and N_D in the dictionary learning process are essential to the quality of the resultant dictionary. To better investigate this matter for the proposed JSM-DKSVD, a sweep of the parameter space of N_D and L_D is performed, using 5% pixels for training and the training window $T_D = 3 \times 3$, for example. The classification accuracies of OMP, SOMP and NLW with JSM-DKSVD-trained dictionaries are depicted in Figure 3.4.

In all OMP (Figure 3.4(a)), SOMP (Figure 3.4(b)) and NLW (Figure 3.4(c)) settings, it can be seen that the performances of the learned dictionary are consistently maximal when the number of atoms N_D is approaching the maximum value. In these cases, the dictionary is large and flexible enough to store the rich information provided by the extended training neighbourhoods in JSM-DKSVD.



(a)

(b)

(c)

Figure 3.4: The overall classification accuracies of using JSM-DKSVD with different numbers of atoms N_D and training sparsity levels L_D. The 5% pixels randomly chosen from the Indian Pines dataset are used to train dictionaries under $T_D = 3 \times 3$. The three SRC methods for testing are (a) OMP, (b) SOMP, and (c) NLW.

When N_D drops below 1,800, the performance becomes unstable, with local maximal observed in different places depending on L_D. This is because: although the dictionaries in these cases are not big enough to support excellent representation of the training neighbourhoods themselves, when the sparsity level L_D is appropriately matched, the resultant dictionary can still achieve a relatively good performance.

Summarising all the L_D dimensions, Figure 3.5 shows the best performance that the dictionary can achieve under different N_D. It can be seen that despite of the local maximum mentioned above, the best performance remains at the places where N_D is close to the number of unique columns of $\mathbf{X}^{JSM}$.

Moreover, we can observe that the performance of the learned dictionary is not sensitive to L_D when N_D is approaching the maximum value. Therefore, based on the above discussion we can take the strategy of setting N_D to be close to the number

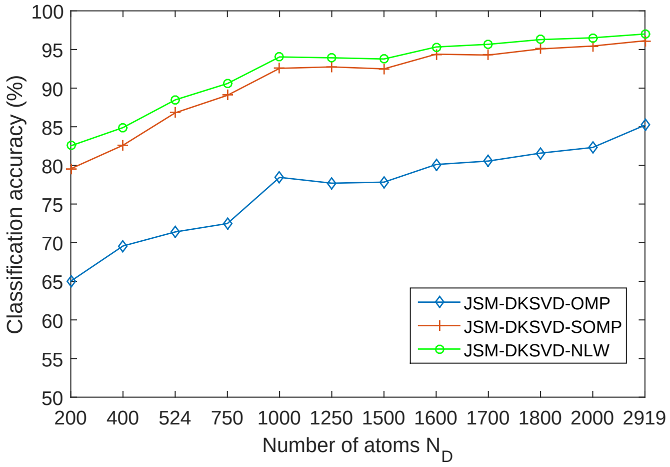


Figure 3.5: The optimal classification accuracies of OMP, SOMP, and NLW using the JSM-DKSVD-trained dictionary with different numbers of atoms N_D. The 5% pixels randomly chosen from the Indian Pines dataset are used to train the dictionaries under $T_D = 3 \times 3$.

of unique columns in $\mathbf{X}^{JSM}$ and giving L_D certain flexibility.

3.5.3 ROSIS dataset: University of Pavia

Table 3.4: The Pavia University dataset: Ground-truth labels, class material, the training set and the test set.

Class	materials	Training	Test
1	Asphalt	67	6564
2	Meadows	187	18462
3	Gravel	21	2078
4	Trees	31	3033
5	Painted metal sheets	14	1331
6	Bare soil	51	4978
7	Bitumen	14	1316
8	Self-blocking bricks	37	3645
9	Shadows	10	937
Total		432	42344

The ROSIS University of Pavia dataset consists of 610 × 340 pixels from 103 spectral bands, with nine ground-truth labels. We randomly choose only 1% of labelled samples for training, i.e. $\mathbf{X}^{train} \in \mathbb{R}^{103\times432}$ and the rest for testing, i.e. $\mathbf{X}^{test} \in \mathbb{R}^{103\times42344}$. A summary of this dataset is given in Table 3.4. The nine ground-truth classes, the training set as well as the test set are shown in Fig-

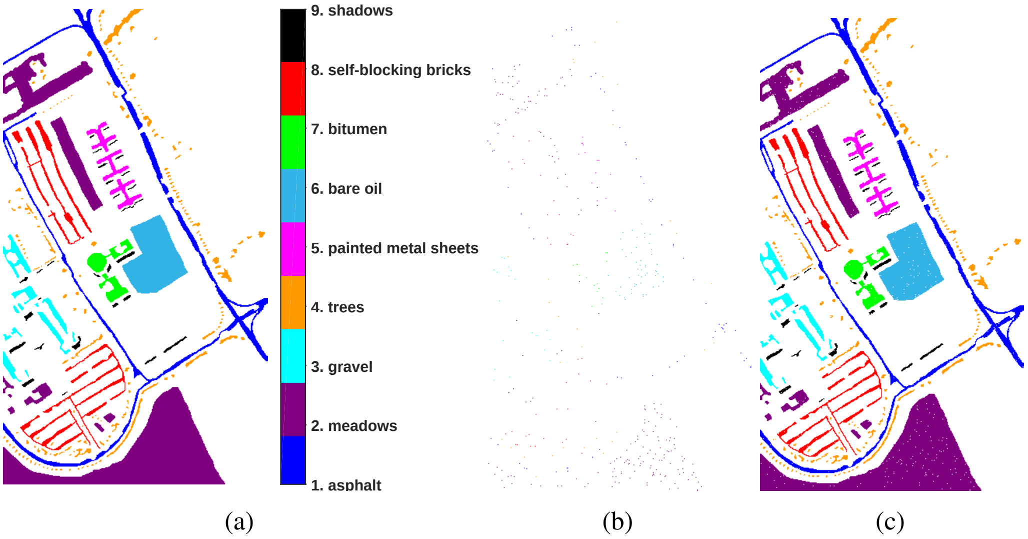


Figure 3.6: The University of Pavia dataset with 1% pixels randomly chosen for training: (a) ground-truth labels; (b) training set; (c) test set.

ures. 3.6(a)-3.6(c).

For the three SRC methods, OMP, SOMP, NLW using $\mathbf{D}^{raw}$, the optimal parameters obtained by LOOCV are $L_C = 5$ for OMP, $L_C = 10$ and $T_C = 3\times 3$ for SOMP and $L_C = 20$ and $T_C = 5\times 5$ for NLW.

For the D-KSVD and JSM-DKSVD algorithms, we set the number of atoms $N_D = 432$ for D-KSVD. For the JSM-DKSVD, the unique number of atoms is 3,604 under the training window $T_D = 3\times 3$, and 9,344 under the training window $T_D = 5\times 5$. Therefore N_D is set as 3,604 and 9,344 under $T_D = 3\times 3$ and $T_D = 5\times 5$, respectively.

As with section 3.5.2, we randomly split the dataset into training-test pairs for 20 times. All of the 20 overall classification accuracies are box-plotted in Figure 3.7. The classification results for one experiment random selected from the 20 experiments are shown in Table 3.5 and Figures. 3.8(a)-3.8(l). Once again, we can observe that the JSM-DKSVD-trained dictionary combined with the three SRC methods outperforms the other two methods ($\mathbf{D}^{raw}$ and D-KSVD) in both cases of $T_D = 3\times 3$ and $T_D = 5\times 5$.

Again, we compare our results against those in [37] which is a state-of-the-art dictionary learning method. Even with only 1% of the pixels used for training, the

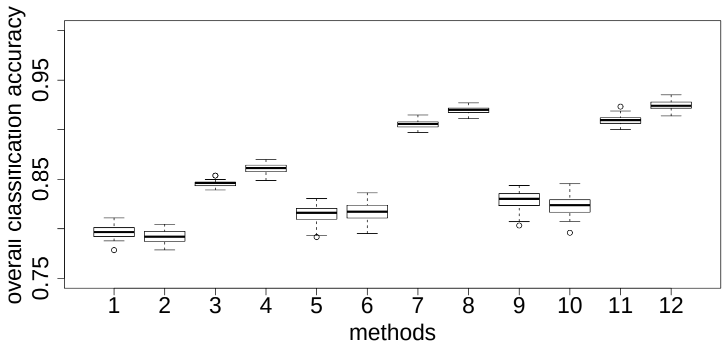


Figure 3.7: Boxplots of the overall classification accuracies the University of Pavia dataset: (1) $\mathbf{D}^{raw}$-OMP, (2) DKSVD-OMP, (3) JSM-DKSVD-OMP under $T_D = 3 \times 3$, (4) JSM-DKSVD-OMP under $T_D = 5 \times 5$, (5) $\mathbf{D}^{raw}$-SOMP, (6) DKSVD-SOMP, (7) JSM-DKSVD-SOMP under $T_D = 3 \times 3$, (8) JSM-DKSVD-SOMP under $T_D = 5 \times 5$, (9) $\mathbf{D}^{raw}$-NLW, (10) DKSVD-NLW, (11) JSM-DKSVD-NLW under $T_D = 3 \times 3$, and (12) JSM-DKSVD-NLW under $T_D = 5 \times 5$. Each boxplot is constructed from the results of 20 experiments and 1% pixels are randomly chosen to train the dictionary.

Table 3.5: The classification accuracy (%) on the University of Pavia dataset with 432 training pixels (1% of all pixels) and 42344 test pixels. The notation is as for Table 3.2.

	$\mathbf{D}^{raw}$			D-KSVD			JSM-DKSVD					
T_D	N/A			N/A			3×3			5×5		
N_D	432			432			3604			9344		
	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW	OMP	SOMP	NLW
1	74.21	78.63	83.27	78.47	86.11	90.65	83.68	90.33	93.19	85.33	90.36	93.95
2	92.32	97.82	98.02	91.12	97.36	98.02	91.63	98.00	98.44	92.40	98.34	98.40
3	52.21	63.72	61.50	51.68	61.79	56.79	65.69	76.23	77.33	71.22	79.79	83.16
4	84.54	88.39	88.99	83.42	86.84	84.01	85.86	89.45	89.55	89.28	92.78	92.09
5	99.55	100.00	100.00	95.94	99.02	99.25	98.42	99.85	99.62	98.87	99.92	99.47
6	58.54	63.00	60.81	58.28	64.34	62.56	68.74	74.93	74.89	74.21	81.86	82.74
7	73.25	88.83	89.21	72.64	88.15	86.85	72.64	88.91	87.84	77.20	90.58	91.19
8	65.54	76.19	75.23	59.34	72.18	69.66	67.54	80.80	76.87	71.22	81.59	78.68
9	81.00	81.75	81.96	92.53	97.01	94.77	95.73	98.40	98.51	92.32	97.55	97.65
OA	80.10	85.97	86.39	79.68	86.83	86.86	83.66	90.72	91.04	85.81	92.20	**92.77**
AA	75.69	82.04	82.11	75.94	83.65	82.51	81.10	88.54	88.47	83.56	90.31	**90.82**
κ	0.734	0.811	0.816	0.729	0.823	0.822	0.783	0.876	0.880	0.812	0.896	**0.903**

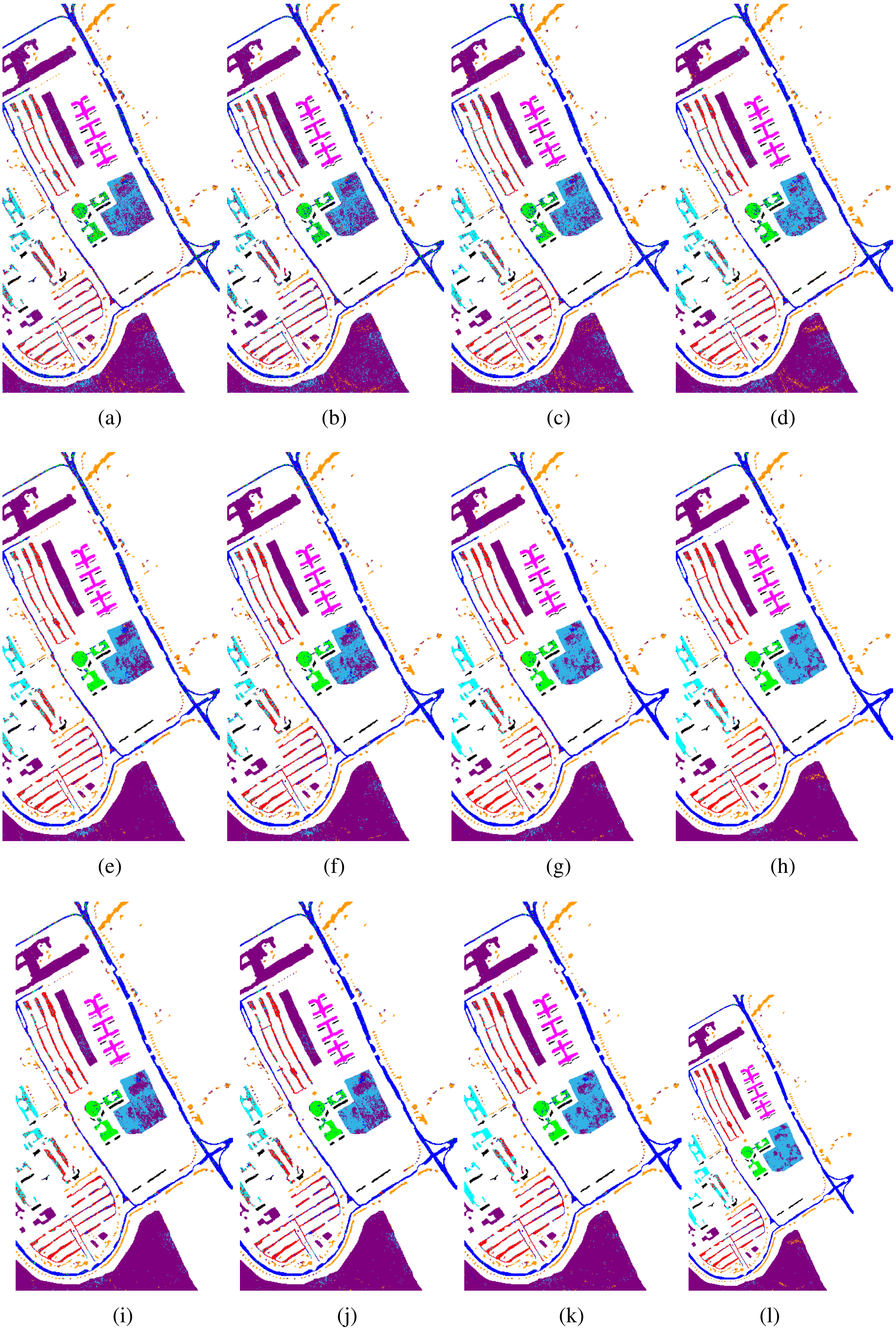


Figure 3.8: The classification maps of the University of Pavia dataset with 1% pixels randomly chosen for training: (a) $\mathbf{D}^{raw}$-OMP; (b) DKSVD-OMP (c) JSM-DKSVD-OMP (3×3); (d) JSM-DKSVD-SOMP (5×5); (e) $\mathbf{D}^{raw}$-SOMP; (f) DKSVD-SOMP (g) JSM-DKSVD-SOMP (3×3); (h) JSM-DKSVD-SOMP (5×5); (i) $\mathbf{D}^{raw}$-NLW; (j) DKSVD-NLW (k) JSM-DKSVD-NLW (3×3); (l) JSM-DKSVD-NLW (5×5).

proposed JSM-DKSVD method achieves higher OA (92.77% as shown in Table 3.5) than that reported (85.70%) in [37], which is evaluated with 10% pixels as training pixels.

3.5.4 Discussion

JSM-DKSVD utilises the JSM constraint in a fundamentally different way from JSM-based classification methods (SOMP and NLW): JSM-DKSVD applies its constraint to dictionary learning while SOMP and NLW apply their constraints to classification. Moreover, the JSM constraint in classification is used to ensure stable sparse representation for the test pixels when they are classified, while the JSM constraint in JSM-DKSVD is used to ensure richer spectral and spatial information incorporated into the learned dictionary.

As dictionary learning can be treated as a pre-processing step for the subsequent classification process and the learned dictionary can be utilised by any sparse representation-based classifiers, JSM-DKSVD can be compatibly utilised in conjunction with existing JSM-based or non-JSM-based classification methods, such as SOMP, NLW and OMP. Because of the difference in the use of the JSM constraint, such a combination of dictionary learning and classification will not introduce undesirable over-smoothness.

Nevertheless, it is worth noting that the JSM constraint itself, be it executed in the dictionary learning or classification process, is based on the grand assumption of signal continuity in natural images. This assumption may be violated in certain part of an image in practice. The violation might be caused by the low resolution of capturing devices, by the very existence of pixels near the border of object regions, or simply by the effect of random noises. This limits the performance of all dictionary learning/classification methods that are based on this assumption.

For example, as is shown in Table 3.2 for class 7 and class 9 of the Indian Pine dataset, the OMP method, which is not based on the JSM assumption, in fact achieves higher classification accuracies than SOMP and NLW, which are JSM-based. We note that both class 7 and class 9 are small regions with only 26 and 20 pixels in total, respectively (shown in Table 3.1).

In fact, such a small class is prone to violate the smoothness assumption and may prefer a small window for dictionary learning and classification. As we can observe, applying a stronger JSM-constraint during dictionary learning by switching from $T_D = 3 \times 3$ to a larger window $T_D = 5 \times 5$ actually results in a drop of performance of all three classification methods (OMP, SOMP, and NLW) for class 7 and class 9. The optimal choice of the window size (e.g. T_D) can be data-dependent, as is the case for SOMP and NLW where the JSM-constraint is employed for classification and for JSM-DKSVD where the constraint is employed for dictionary learning. It is indeed of our research interests to further investigate and make the window selection process data-adaptive for JSM-DKSVD.

Table 3.6: Execution time (sec/atom) spent on the University of Pavia dataset with 432 training pixels (1% of all pixels) for training dictionaries.

	$\mathbf{D}^{raw}$	D-KSVD	JSM-DKSVD (3×3)	JSM-DKSVD (5×5)
Time		0.014	0.245	0.512

Finally for reference purposes, we discuss the time cost for training dictionaries. All experiments are performed on Xeon E5-1650 CPU (single thread). Table 3.6 lists the execution time of training dictionaries by D-KSVD, JSM-DKSVD (3×3) and JSM-DKSVD (5×5) conducted at their optimal parameters for the Pavia dataset with 1% pixels randomly chosen for training. The execution time (sec/atom) is normalised by the numbers of trained atoms, i.e. 432 in D-KSVD and 3604 in JSM-DKSVD, respectively. Firstly, it should be noted that there is no training phase on $\mathbf{D}^{raw}$ since the atoms of the dictionary are constructed directly from the training pixels. Secondly, JSM-DKSVD spends more time than D-KSVD for both window sizes, i.e. $T_D = 3 \times 3$ and 5×5, and JSM-DKSVD (5×5) spends the most. These are expected, as the extra cost comes from the neighbours involved with JSM in the training phase. This time/dictionary quality trade-off is often preferred for offline training, which is not uncommon in the literature of the HSI classification.

3.6 Conclusion

In this chapter, we have proposed a novel dictionary learning method called JSM-DKSVD for hyperspectral image classification. Based on the concept of joint sparse modelling, we incorporate spectral and spatial structure information into the process of discriminative K-SVD, which results in a more informative and discriminative dictionary. Experiment results demonstrate that the proposed JSM-DKSVD achieves better classification performance than those using established dictionary construction methods, even when only a very small fraction (1% for example) of the pixels from the benchmark HSI are used for training.

Chapter 4

HSI Classification: Cone-based Joint Sparse Modelling (C-JSM)

4.1 Introduction

Sparse representation has been proven to be superiorly effective for a wide range of applications in computer vision, pattern recognition and signal processing [42]. It is based on the assumption that most natural signals can be compactly represented by a linear combination of only a few basis vectors (aka atoms) from an over-complete dictionary.

Recently, sparse representation has been extensively investigated in hyperspectral imaging [4]. A hyperspectral image (HSI) is a 3-dimensional data cube with two spatial dimensions and one spectral dimension. From the view of the spectral dimension, each HSI pixel is a vector, namely spectral signature whose elements correspond to reflectances at different wavelengths (spectral bands). Different classes of spectral signatures can have distinct reflectances at specific wavelengths and, as a result, the spectral signatures can provide discriminative information for classification. The sparse representation of an HSI pixel is accomplished by a linear combination of atoms in a spectral dictionary. The sparse model can be approximately solved by greedy algorithms such as orthogonal matching pursuit (OMP) [11] (l_0-norm based methods) or by convex optimisation problems such as the Lasso (l_1-norm based methods). In such sparse representation, the dictionary is usually constructed by the training spectral signatures directly from HSIs or spectral libraries. Note that, to achieve higher classification performance, dictionary learning has been also investigated for HSI analysis. Details of how to design and learn quality dictionaries for HSI classification can be found in [35, 36, 37, 38, 43], for example.

A step further, by virtue of the signal coherence in HSIs, a joint sparse model (JSM) has been successfully developed for HSI classification and has achieved promising performance [4]. The underlying assumption of JSM is that all HSI pixels in a small spatial neighbourhood can be jointly approximated by sparse linear combinations of a few common training samples, i.e. the neighbourhood shares a common sparse model. The original JSM proposed by [4] adopts a square window centred on a test pixel for joint modelling; a greedy algorithm, namely simultaneous orthogonal matching pursuit (SOMP) [8], is used to solve JSM. On top of this JSM, some extensions have been proposed to overcome the limitations of JSM [25, 27, 28, 29, 44, 45]. To extend JSM for linearly non-separable class samples, the kernel versions of SOMP have been studied in [44, 45]. To enhance JSM with a more effective neighbourhood, the adaptive versions of JSM have been proposed in [25, 27, 28, 29], which aim to produce shape/size adaptive local windows for JSM.

An important property of hyperspectral signals is the non-negativity, for both the signal itself and the abundance coefficients. It has been intensively considered for problems of HSI unmixing [46, 47, 48, 49, 50, 51, 52]. A variety of reports have been focused on the non-negative matrix factorisation (NMF), a typical decomposition method for the HSI unmixing problems [46, 47, 48]. NMF decomposes the sample data matrix into two low-dimensional matrices serving as endmembers and coefficients, both of which are enforced to be non-negative. The underlying assumption of the NMF-based unmixing is that mixed HSI pixels can be decomposed into a collection of endmembers and the corresponding proportions. Due to the physical characteristics, the endmembers, which characterise the reflected electromagnetic energy of specific materials, should be non-negative. In addition, the proportions of the underlying physical materials (endmembers) are non-negative for physical interpretations. In practice, as the standard NMF [46] is non-convex and may fall into local minima, several enhancements have been proposed. A typical constrained NMF algorithm called minimum-volume-constrained NMF [47] is proposed to combine a geometry assumption with the NMF, enforcing the minimisation of the simplex volume. Liu et al. [48] propose to add the abundance separation constraint and the smoothness constraint to the NMF to take the spatial and spectral coherence into consideration. Other constraints are also considered, such as the neighbourhood information of pixels [50] and the dissimilarity of signatures [51].

However, research of sparse representation for HSI classification, particularly the JSM-based methods in [4, 25, 27, 28, 29], have not incorporated the non-negativity properties of HSI. To fill in this gap, through replacing the signal representation of JSM by cone representation, in this chapter we incorporate non-negativity into HSI classification and propose a new HSI classification model called cone-based joint sparse model (C-JSM).

Methodologically, inspired by the NMF for HSI unmixing, we devise the non-negativity constraint on the coefficient matrix of JSM for HSI classification. Since the given atoms of a dictionary are constructed directly by the HSIs or from spectral libraries, it implies that the dictionary atoms can be regarded as endmemebers. In this fashion, both endmembers and coefficients are non-negative, and thus the proposed C-JSM considers both sparsity and non-negativity, making the joint sparsity recovery problem more realistic in terms of interpretation. It will be illustrated to have a more sparse and stable representation than the conventional JSM.

Computationally, we propose a new algorithm called non-negative simultaneous orthogonal matching pursuit (NN-SOMP) to solve the C-JSM problem. The proposed NN-SOMP algorithm is developed on the basis of the SOMP algorithm with an additional non-negative constraint on the coefficients, which will be illustrated easy to implement in this chapter.

In short, the main contribution of this chapter can be summarised as follows: 1) we incorporate the non-negativity constraints into JSM to consider more realistic physical characteristics of the spectral signals and propose a new HSI classification model called C-JSM; 2) we also propose a new NN-SOMP algorithm to solve the optimisation problem of C-JSM; and 3) C-JSM produces a stable sparse representation as well as a superior classification performance.

The rest of the chapter is organised as follows. Section 4.3 introduces the cone-based model and the cone-based sparse model. In section 4.4 and section 4.5, the proposed C-JSM, as well as the proposed algorithm NN-SOMP to solve the C-JSM problem, are detailed. Experimental studies in section 4.6 demonstrates the superior classification performance of C-JSM over the compared methods on two real hyperspectral datasets. Finally this work is discussed and concluded in section 4.7.

4.2 Joint sparse models for HSI classification

The sparse model (SM) and the joint sparse model (JSM) are reviewed in section 2.2.1 and section 2.2.2 of Chapter 2, respectively. The notations in this chapter also align with those in section 2.2.1 and section 2.2.2.

4.3 Cone-based sparse model

4.3.1 Cone-based model

A cone model (CM) to represent vectors **x** is defined as

$$\mathbf{C}:\left\{\mathbf{x}|\mathbf{x}=\sum_{i=1}^{N}\alpha_i\mathbf{d}_i=\mathbf{D}\alpha,\alpha_i\geq 0\right\}, \tag{4.1}$$

where α_i is the *non-negative* coefficient of atom $\mathbf{d}_i$, and α is an N-dimensional vector of *non-negative* coefficients.

The non-negative coefficient vector α is estimated by solving the following optimisation problem:

$$\hat{\alpha}=\underset{\alpha}{\operatorname{argmin}}\,||\mathbf{x}-\mathbf{D}\alpha||_2^2,\ \text{s.t. } \alpha\geq\mathbf{0}, \tag{4.2}$$

where $\alpha\geq\mathbf{0}$ denotes that every single element of the vector α should be non-negative. Problem (4.2) can be solved by the active-set methods, such as the typical non-negative least square method (NNLS) [53] (MATLAB function *lsqnonneg*) and its extension fast-NNLS (fnnls) [54]. In this chapter, we use the CM (4.2) as a baseline method for HSI classification. Specifically, (4.2) is used for the representation of a single test HSI pixel. The label of a test pixel is determined by (2.4), as with the rule used by the SM (2.2).

4.3.2 Cone-based sparse model

For the l_0-pseudo-norm optimisation problem, the non-negative orthogonal matching pursuit (NN-OMP) algorithm has been investigated in [55], which introduces the non-negativity constraint into the conventional OMP algorithm. Technical details of the algorithm vary, depending on different criteria such as fast implementation [56].

In [55, 56], a desired coefficient vector α is estimated by solving the following optimisation problem:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2, \text{ s.t. } \|\alpha\|_0 \leq L \text{ and } \alpha \geq \mathbf{0}, \tag{4.3}$$

which is forced to be sparse and non-negative.

In this chapter, we term model (4.3) as the cone-based sparse model (CSM). To our knowledge, CSM is first introduced and studied in this chapter for HSI classification. To align with the rule of SM (2.2), the classification of an HSI based on CSM (4.3) is also determined by (2.4).

4.4 Cone-based joint sparse model (C-JSM) for HSI classification

We notice that, on the one hand, SM (2.2), CM (4.2) and CSM (4.3) are all constructed for a single test HSI pixel and do not take the spatial coherence [57] into consideration; while on the other hand, JSM accounts for the neighbouring spatial information, but the coefficients estimated by JSM are only assumed to be sparse, not necessarily non-negative. As with the underlying assumptions made for HSI unmixing, an HSI pixel can be decomposed into a collection of endmembers with non-negative proportions. The endmembers are spectral signatures which characterise the reflect electromagnetic energy of specific materials and hence are non-negative. In the case of HSI classification, the dictionary atoms are usually constructed directly from the HSI or from the spectral libraries, so the atoms can be assumed acting as endmembers, which inspires us to devise a cone-based representation for the joint models for a more realistic interpretation.

In the same notation as aforementioned in JSM (section 2.2.2), the cone-based representation of a test window $\mathbf{X} \in \mathbb{R}^{B \times T}$ can be formulated as follows:

$$\mathbf{X} \approx \mathbf{D}\mathbf{A}, \text{ s.t. } \mathbf{A} \geq \mathbf{0}, \tag{4.4}$$

where $\mathbf{A}$ is a *non-negative* coefficient matrix and $\mathbf{A} \geq \mathbf{0}$ denotes that every element of $\mathbf{A}$ should be non-negative. To estimate $\mathbf{A}$, problem (4.4) can be reformulated as

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2, \text{ s.t. } \mathbf{A} \geq \mathbf{0}. \tag{4.5}$$

In this chapter, we term model (4.4) as the joint cone model (shortened as JCM). We also utilise it as a baseline method.

The optimisation problem (4.5) can be solved by two algorithms. Firstly, the reconstruction of each column vector $\mathbf{x}_t$ for $t = 1, \ldots, T$ can be solved independently by the conventional NNLS [53] or fast-NNLS [54]. Secondly, it can be solved by an algorithm called fast combination NNLS (FC-NNLS) [58], which is proposed to solve the large-scaled non-negativity-constrained least square problems. It solves a set of linear reconstruction for $\mathbf{X}$ in a parallel fashion instead of solving a set of single $\mathbf{x}$ in a serial fashion. Specifically, it rearranges the calculations in the standard active-set NNLS on the basis of combinational reasoning and reduces the computation burden for NNLS problems when there are a large number of observations, i.e. a large window size T in our case. The estimated coefficient matrices $\hat{\mathbf{X}}$s obtained by these two methods are the same. So we can regard FC-NNLS as a fast implementation of NNLS for solving the JCM problem (4.5).

Incorporating JCM (4.5) into the JSM for HSI classification (2.6), we propose a new method as

$$\begin{aligned} \hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{DA}\|_F^2, \\ \text{s.t. } \|\mathbf{A}\|_{row,0} \leq L \text{ and } \mathbf{A} \geq \mathbf{0}. \end{aligned} \tag{4.6}$$

We call this new model (4.6) as cone-based joint sparse model (shortened as C-JSM). In short, the proposed C-JSM incorporates the non-negative constraints into the sparse representation of a test window **X** by joint modelling. The coefficient matrix **A** of the test window **X** is not only sparse, but also forced to be non-negative. On top of these two desirable properties, the spatial coherence of HSI is also reflected in that the coefficient vector of the central test pixel $\mathbf{x}_c$ is jointly determined by those HSI pixels in its local neighbourhood with the same non-negative and sparse constraints. As a result, HSI pixels in the local window **X** share the same basis vectors of a cone, and the sparsity of the coefficients are determined only in the region of the cone.

Same as JSM, the two cone-based sparse models, JCM and C-JSM, are also joint models, hence we adopt the classification rule (2.8) for them. To solve the C-JSM problem (4.6), we propose a new algorithm and detail it in the following section 4.5.

4.5 Algorithm of NN-SOMP for solving C-JSM

We propose a new algorithm called non-negative simultaneous orthogonal matching pursuit (NN-SOMP), to solve the C-JSM problem. It combines the NNLS-based methods and the SOMP algorithm together to produce a non-negative and sparse estimation of the coefficient matrix $\hat{\mathbf{A}}$ in (4.6). Before introducing the proposed NN-SOMP, we first present the non-negative OMP to get an insight of the paradigm.

4.5.1 Algorithm of NN-OMP

The traditional SM in (2.2) with the l_0-pseudo-norm constraint on the coefficient vector is approximately solved by greedy algorithms, of which one of the most popular algorithms is called orthogonal matching pursuit (OMP) [11]. We assume

Algorithm 2 The OMP algorithm [11] to solve SM (2.2).

```
Input:  • Dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] \in \mathbb{R}^{B \times N}$ with $\|\mathbf{d}_i\|_2^2 = 1$ for $i = 1, \ldots, N$.
        • A test pixel $\mathbf{x} \in \mathbb{R}^B$.
        • Stopping criteria: sparsity level $L$ or threshold $\tau$.
Output: A sparse coefficient vector $\hat{\alpha}$.
  Initialisation:
        • The residual vector $\mathbf{r}_0 = \mathbf{x}$.
        • Sparse index set $\Lambda_0 = \varnothing$.
        • Iteration counter $j = 1$.
  while $j \leqslant L$ or $\left\|\mathbf{r}_{j-1}\right\|_2^2 < \tau$ do
      (1) Find an index $\lambda_j$ that solves the easy optimisation problem:
              $\lambda_j = \underset{i=1,\ldots,N}{\operatorname{argmax}} \left|\mathbf{d}_i^T \mathbf{r}_{j-1}\right|.$    (4.7)
      (2) Update the index set $\Lambda_j = \Lambda_{j-1} \cup \{\lambda_j\}$.
      (3) Compute the coefficient vector $\beta_j$ by the atoms of $\mathbf{D}$ indexed in $\Lambda_j$:
              $\hat{\beta}_j = (\mathbf{D}_{\Lambda_j}^T \mathbf{D}_{\Lambda_j})^{-1} \mathbf{D}_{\Lambda_j}^T \mathbf{x}$    (4.8)
          where $\mathbf{D}_{\Lambda_j} \in \mathbb{R}^{B \times j}$ consists of the $j$ atoms in $\mathbf{D}$ indexed in $\Lambda_j$.
      (4) Determine the new residual:
              $\mathbf{r}_j = \mathbf{x} - \mathbf{D}_{\Lambda_j} \hat{\beta}_j.$    (4.9)
      (5) $j \leftarrow j + 1$.
  end while
  Compute the sparse coefficient vector $\hat{\alpha}$ whose non-zero elements are indexed by
  $\Lambda$ and the corresponding $L$ elements of vector $\hat{\beta}_L$.
```

Algorithm 3 The NN-OMP algorithm to solve CSM (4.3).

```
Input:  • Dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] \in \mathbb{R}^{B \times N}$ with $\|\mathbf{d}_i\|_2^2 = 1$ for $i = 1, \ldots, N$.
        • A test pixel $\mathbf{x} \in \mathbb{R}^B$.
        • Sparsity level $L$ or threshold $\tau$.
Output: A non-negative and sparse coefficient vector $\hat{\alpha}$.
  Initialisation:
        • The residual vector $\mathbf{r}_0 = \mathbf{r}$.
        • Sparse index set $\Lambda_0 = \varnothing$.
        • Iteration counter $j = 1$.
  while $j \leqslant L$ or $\left\|\mathbf{r}_{j-1}\right\|_2^2 < \tau$ do
      (1) Find an index $\lambda_j$ that solves the easy optimisation problem:
              $\lambda_j = \underset{i=1,\ldots,N}{\operatorname{argmax}} \left|\mathbf{d}_i^T \mathbf{r}_{j-1}\right|.$        (4.10)
      (2) Update the index set $\Lambda_j = \Lambda_{j-1} \cup \{\lambda_j\}$.
      (3) Determine non-negative coefficient vector $\beta_j$ by the NNLS algorithm in the
          cone $C$ whose basis vectors are the atoms of $\mathbf{D}$ indexed in $\Lambda_j$:
              $\hat{\beta}_j = \underset{\beta_j}{\operatorname{argmin}} \left\|\mathbf{x} - \mathbf{D}_{\Lambda_j}\beta_j\right\|_2^2, \text{s.t. } \beta_j \geq \mathbf{0},$        (4.11)
          where $\mathbf{D}_{\Lambda_j} \in \mathbb{R}^{B \times j}$ consists of the $j$ atoms in $\mathbf{D}$ indexed in $\Lambda_j$.
      (4) Determine the new residual:
              $\mathbf{r}_j = \mathbf{x} - \mathbf{D}_{\Lambda_j}\hat{\beta}_j.$        (4.12)
      (5) $j \leftarrow j + 1$.
  end while
  Compute the non-negative and sparse coefficient vector $\hat{\alpha}$ whose non-zero elements are indexed by $\Lambda$ and the corresponding $L$ elements of vector $\hat{\beta}_L$.
```

that the columns (atoms) of the dictionary $\mathbf{D}$ are normalised so that $\|\mathbf{d}_i\|_2 = 1$ for $i = 1,\ldots,N$. At the beginning of the algorithm, a residual vector $\mathbf{r}_0$ is initialised to be the test HSI pixel $\mathbf{x}$. The OMP iteratively selects at each step the column of $\mathbf{D}$, i.e. the atom $\mathbf{d}_i$, which has not been selected but is most correlated with the residuals $\mathbf{r}_{j-1}$, where j is the current iteration number. The maximal correlation is calculated as $\left|\mathbf{d}_i^T \mathbf{r}_{j-1}\right|$, which is the absolute value of the projection of residual vector $\mathbf{r}_{j-1}$ onto the the atom $\mathbf{d}_i$. The selected atom $\mathbf{d}_i$ is then added into the set of selected atoms. The algorithm updates the residual vector by projecting the observed vector $\mathbf{x}$ onto the linear subspace spanned by the atoms that have already been selected, and then iterates. The termination of the OMP algorithm is either conducted by setting the iteration number, i.e. the sparsity level L, or by setting a threshold τ of the residual. The OMP algorithm is summarised in Algorithm 2.

Based on the OMP algorithm, the non-negative OMP (NN-OMP) is proposed by incorporating the non-negativity constraint on the coefficients into the iterations. The main difference between OMP and NN-OMP is the updating criteria of residual vector $\mathbf{r}_j$. In OMP, the residual vector is updated by (4.9), where the coefficient vector β_j is obtained by least squares (LS) (4.8) and has a closed-form solution. However in NN-OMP, to guarantee non-negative coefficients, the coefficient vector β_j at iteration j should be solved by NNLS-based methods instead of the LS method, which is described in (4.11). Hence there is no closed-form solution for β_j. The algorithm of NN-OMP used in this chapter is summarised in Algorithm 3.

Other versions of NN-OMP can be found in [55, 56]. We note that there is a slight difference between Algorithm 3 and the algorithms proposed in [55, 56]: we use the absolute value $\left|\mathbf{d}_i^T \mathbf{r}_{j-1}\right|$ instead of the maximal positive value $\max(\mathbf{d}_i^T \mathbf{r}_{j-1}) > 0$ used in [55, 56]. Although these two approaches may select different atoms from iteration 2 (for iteration 1, $\mathbf{r}_0$ and $\mathbf{d}_i$ both are positive so the produced results are same), the size of residuals $\left\|\mathbf{r}_j\right\|_2$ can be reduced iteratively by both, which reflects the core idea of matching pursuit algorithms. It may not be easy to claim which approach is more appropriate. To align with the original framework of OMP and for a clearer comparison, we only change the updating of the coefficients by (4.11) and adopt Algorithm 3 as a representative of NN-OMP algorithm in the following discussion.

4.5.2 Algorithm of NN-SOMP

Algorithm 4 The SOMP algorithm [4] to solve JSM (2.6).

```
Input:  • Dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_N] \in \mathbb{R}^{B \times N}$ with $\|\mathbf{d}_i\|_2^2 = 1$ for $i = 1, \dots, N$.
        • A test window $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{R}^{B \times T}$.
        • Sparsity level $L$.
Output: A non-negative and sparse coefficient matrix $\hat{\mathbf{A}}$.
  Initialisation:
    • The residual matrix $\mathbf{R}_0 = \mathbf{X}$.
    • Sparse index set $\Lambda_0 = \varnothing$.
    • Iteration counter $j = 1$.
  while $j \leqslant L$ or $\left\|\mathbf{R}_{j-1}\right\|_F^2 < \tau$ do
    (1) Find an index $\lambda_j$ that solves the following easy optimisation problem:
            $\lambda_j = \underset{i=1,\dots,N}{\operatorname{argmax}} \left\|\mathbf{R}_{j-1}^T \mathbf{d}_i\right\|_p, \; p \geq 1.$      (4.13)
    (2) Update the index set $\Lambda_j = \Lambda_{j-1} \cup \{\lambda_j\}$.
    (3) Determine coefficient matrix $\mathbf{P}_j$ by the atoms of $\mathbf{D}$ indexed in $\Lambda_j$:
            $\hat{\mathbf{P}}_j = (\mathbf{D}_{\Lambda_j}^T \mathbf{D}_{\Lambda_j})^{-1} \mathbf{D}_{\Lambda_j}^T \mathbf{X}$      (4.14)
        where $\mathbf{D}_{\Lambda_j} \in \mathbb{R}^{B \times j}$ consists of the $j$ atoms in $\mathbf{D}$ indexed in $\Lambda_j$.
    (4) Determine the new residual matrix:
            $\mathbf{R}_j = \mathbf{X} - \mathbf{D}_{\Lambda_j} \hat{\mathbf{P}}_j.$      (4.15)
    (5) $j \leftarrow j + 1$.
  end while
  Compute the sparse coefficient matrix $\hat{\mathbf{A}}$ whose non-zero rows are indexed by $\Lambda$
  and the corresponding $L$ rows of matrix $\hat{\mathbf{P}}_L$.
```

Following the derivation of NN-OMP from OMP, we propose a new algorithm called NN-SOMP, which combines the SOMP algorithm [8] and the NNLS-based methods together to solve the problem of C-JSM (4.6).

Algorithm 5 The NN-SOMP algorithm to solve C-JSM (4.6).

```
Input:  • Dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_N] \in \mathbb{R}^{B \times N}$ with $\|\mathbf{d}_i\|_2^2 = 1$ for $i = 1, \ldots, N$.
        • A test window $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{B \times T}$.
        • Sparsity level $L$.
Output: A non-negative and sparse coefficient matrix $\hat{\mathbf{A}}$.
  Initialisation:
    • The residual matrix $\mathbf{R}_0 = \mathbf{X}$.
    • Sparse index set $\Lambda_0 = \varnothing$.
    • Iteration counter $j = 1$.
  while $j \leqslant L$ or $\left\|\mathbf{R}_{j-1}\right\|_F^2 < \tau$ do
    (1) Find an index $\lambda_j$ that solves the following easy optimisation problem:
            $\lambda_j = \underset{i=1,\ldots,N}{\operatorname{argmax}} \left\|\mathbf{R}_{j-1}^T \mathbf{d}_i\right\|_p, \ p \geq 1.$    (4.16)
    (2) Update the index set $\Lambda_j = \Lambda_{j-1} \cup \{\lambda_j\}$.
    (3) Determine non-negative coefficient matrix $\mathbf{P}_j$ by the NNLS-based algorithm
        in the cone $C$ whose basis vectors are the atoms of $\mathbf{D}$ indexed in $\Lambda_j$:
            $\hat{\mathbf{P}}_j = \underset{\mathbf{P}_j}{\operatorname{argmin}} \left\|\mathbf{X} - \mathbf{D}_{\Lambda_j}\mathbf{P}_j\right\|_F^2, \text{s.t. } \mathbf{P}_j \geq \mathbf{0},$    (4.17)
        where $\mathbf{D}_{\Lambda_j} \in \mathbb{R}^{B \times j}$ consists of number of $j$ atoms in $\mathbf{D}$ indexed in $\Lambda_j$. Optimisation
        problem (4.17) can either determined in a serial fashion that each column of $\mathbf{X}$ is treated
        independently and can be approximated by NNLS [53] or fast-NNLS [54]; or in a parallel fashion
        by FC-NNLS [58]. The two approaches produce the same result.
    (4) Determine the new residual matrix:
            $\mathbf{R}_j = \mathbf{X} - \mathbf{D}_{\Lambda_j}\hat{\mathbf{P}}_j.$    (4.18)
    (5) $j \leftarrow j + 1$.
  end while
  Compute the non-negative and sparse coefficient matrix $\hat{\mathbf{A}}$ whose non-zeros rows
  are indexed by $\Lambda$ and the corresponding $L$ rows of matrix $\hat{\mathbf{P}}_L$.
```

The SOMP algorithm [8] is a generalised OMP algorithm. It aims to find a simultaneous approximation of several input signals, i.e. several columns of matrix $\mathbf{X}$, by using different linear combinations of the same atoms of the dictionary. The algorithm balances the error in approximation against the total number of atoms that participate. Specifically, the atoms supporting the sparse solution are sequentially selected from the dictionary. At each iteration, the atom that simultaneously yields the best yet simple approximation to all of the residual vectors is selected. Particularly, at the jth iteration, we calculate an $N \times T$ correlation matrix $Corr = \mathbf{D}^T \mathbf{R}_{j-1}$, where $\mathbf{R}_{j-1}$ is a residual matrix between the test window $\mathbf{X} \in \mathbb{R}^{B \times T}$ and its approximation from the last iteration. The (i,t)th entry in $Corr$ is the correlation between the ith dictionary atom $\mathbf{d}_i$ and the residual vector for $\mathbf{x}_t$, where $t = 1, \ldots, T$ at the current iteration j. In the algorithm, the l_p-norm, where $p \geq 1$, for each of the N rows of $Corr$ is computed. The row index corresponding to the largest l_p-norm is then added into the sparse index set of selected atoms. As mentioned in [4], different values of p have been adopted in literatures, such as $p = 1$ is in [8], $p = 2$ in [59] and $p = \infty$ in [60]. In this chapter we use $p = \infty$ to align with [60]. Similarly to OMP, the termination of the SOMP algorithm is either conducted by setting the iteration number, i.e. the sparsity level L, or by setting a threshold τ of the size of the residual. Details of the SOMP algorithm [4] is shown in Algorithm 4.

The proposed NN-SOMP algorithm is devised on the basis of the SOMP algorithm; it incorporates non-negative constraints in the simultaneous approximation of a test window $\mathbf{X}$. We replace the LS-based estimates of the coefficient matrix $\hat{\mathbf{P}}_j$ in (4.14) of SOMP by the NNLS-based estimates in (4.17), as detailed in Algorithm 5. We can see that the optimisation problem of (4.17) in our proposed algorithm is in fact a standard JCM problem as in (4.5).

As aforementioned, the optimisation problem (4.17) can be solved by two strategies both based on the NNLS methods. For a simple implementation, each column of $\mathbf{X}$ can be treated independently. Specifically, problem (4.17) in the Algorithm 5 is broken into T individual NNLS problems formulated by (4.2). These T problems can be solved by conventional NNLS algorithm [53] or fast-NNLS algorithm [54]. Then the coefficient matrix $\hat{\mathbf{P}}_j$ is obtained by concatenating the estimated coefficient vectors column by column. In this fashion, we need to use an inner FOR loop to compute step (3) of the NN-SOMP algorithm (Algorithm 5). The optimisation problem (4.17) can also be solved by the FC-NNLS algorithm [58], which is a generalised NNLS algorithm. It aims to solve the non-negative least squares with multiple input vectors. FC-NNLS rearranges the selection of the support set, and reduces substantially the computational burden required for the NNLS problems which have large numbers of observation vectors.

The conventional NNLS algorithm utilises the active/passive set method to solve an inequality-constrained least squares problem as a sequence of equality-constrained problems, also termed “column-serial” [58]. FC-NNLS is also based on this NNLS scheme. In general, the overall NNLS in the FC-NNLS is responsible for defining the sequence, but sequentially solving the problem tends to be computationally inefficient as it can result in redundant calculations. To this end, FC-NNLS solves the problem in a “column-parallel” fashion. Specifically, the algorithm firstly groups problems that share a common passive set and solve them together, and then recognises that the passive sets vary from iteration to iteration. Each NNLS iteration for all columns are performed in parallel rather than performing all iterations for each column in series. Note that columns will require different numbers of iterations to achieve optimality. The algorithm of FC-NNLS is detailed in Algorithm 6 in Appendix 4.8.

Although both the conventional NNLS and the FC-NNLS produce the same estimation results, for a faster computation, we adopt the FC-NNLS algorithm to solve (4.17). Details of the proposed NN-SOMP algorithm are summarised in Algorithm 5.

4.6 Experimental studies

In this section, we investigate the performance of the proposed C-JSM method on HSI classification. The experiments are carried out on two well-known real HSI datasets: the AVIRIS Indian Pines dataset and the ROSIS University of Pavia dataset, both of which can be downloaded from [40].

4.6.1 Methods compared

Table 4.1: Compared methods and their corresponding algorithms: CM – cone model; SM – sparse model; CSM – cone-based sparse model; JCM – joint cone model; JSM – joint sparse model; C-JSM – cone-based joint sparse model.

Meth.	CM	SM	CSM	JCM	JSM	C-JSM
Alg.	NNLS	OMP	NN-OMP	FC-NNLS	SOMP	NN-SOMP

We evaluate the proposed C-JSM (4.6) and compare it with five baseline methods: the sparse model (SM) (2.2), the joint sparse model (JSM) (2.6), the cone model (CM) (4.2), the cone-based sparse model (CSM) (4.3) and the joint cone model (JCM) (4.5). Corresponding algorithms used to learn these models are listed in Table 4.1: the proposed NN-SOMP (Algorithm 5), OMP (Algorithm 2), SOMP (Algorithm 4), NNLS [53], NN-OMP (Algorithm 3) and FC-NNLS (Algorithm 6), respectively.

From the point of view of models, these six methods can be grouped into two types of models: single models (CM, SM, and CSM) and joint models (JCM, JSM and C-JSM). The single models label a test HSI pixel by considering only the test pixel, i.e. a vector $\mathbf{x}$ in (2.2), (4.2) and (4.3), whereas the joint models label a central test HSI pixel $\mathbf{x}_c$ by considering a local window around it, i.e. a matrix $\mathbf{X}$ in (2.6), (4.5) and (4.6). The labelling by the single models is determined by (2.4), whereas the labelling by the joint models is determined by (2.8). The compared methods can also be grouped according to their constraints on non-negativity and sparsity: CM and JCM are only with the non-negativity constraint; SM and JSM are only with the sparsity constraint; and CSM and C-JSM both consider the non-negativity and sparsity simultaneously. Details of the relationships among the methods, algorithms, models and constraints are presented in the confusion matrices in Table 4.2 and Table 4.3.

4.6.2 Performance measures

We evaluate the performances of the compared methods by using three standard measures for HSI classification: the overall accuracy (*OA*), the average accuracy

Table 4.2: Compared methods and their groups.

	Non-negative	Sparse	Non-negative + Sparse
Single model	CM	SM	CSM
Joint model	JCM	JSM	C-JSM

Table 4.3: Compared algorithms and their groups.

	Non-negative	Sparse	Non-negative + Sparse
Single model	NNLS	OMP	NN-OMP
Joint model	FC-NNLS	SOMP	NN-SOMP

(AA) and kappa coefficient κ [41], which are widely used by the remote sensing community.

The OA, AA and κ are defined as follows:

$$OA = \frac{N_{corr}}{N_{test}},\ AA = \frac{1}{M}\sum_{m=1}^{M}\frac{N_m^{corr}}{N_m^{class}} \text{ and } \kappa = \frac{OA - p_e}{1 - p_e}. \tag{4.19}$$

In (4.19), the overall accuracy (OA) is defined as the ratio of the number of the correctly-classified test pixels N_{corr} over the total number of test pixels N_{test}. The average accuracy (AA) is defined as the average value of M accuracies of the M individual classes, where N_m^{corr} is the total number of test pixels of class m, and N_m^{class} is the number of the correctly-classified test pixels of class m. The κ coefficients measures the percentage of classified test pixels corrected by the number of agreements that would be expected purely by change [41]. In (4.19), we have $p_e = \sum_{m=1}^{M}(F_m \times F_m^t)$, where F_m is the ratio of data assigned to class m by the classifier and F_m^t is the ratio of data that belong to class m.

4.6.3 Parameter settings

Among the compared methods, in single models, only one unknown parameter needs to be determined, i.e. the sparsity level L; in joint models, two unknown parameters are involved, the sparsity level L and the window size T, except for FC-NNLS in which only the window size T is involved. The values of the parameters for all methods are determined via the leave-one-out cross validation (LOOCV) in the training phase.

4.6.4 Real dataset: Indian Pines

Table 4.4: The Indian Pines dataset: Ground-truth label, class material, training set and test set. We use around (9% of all pixels) for training and the rest for testing.

Class	Material	Training	Test
1	Alfalfa	5	49
2	Corn-notill	132	1302
3	Corn-mintill	77	757
4	Corn	22	212
5	Grass-pasture	46	451
6	Grass-trees	69	678
7	Grass-pasture-mowed	3	23
8	Hay-windrowed	45	444
9	Oats	2	18
10	Soybean-notill	89	879
11	Soybean-mintill	227	2241
12	Soybean-clean	57	557
13	Wheat	20	192
14	Woods	119	1175
15	Buildings-grass-trees-drives	35	345
16	Stone-steel-towers	9	86
Total		957	9409

The AVIRIS Indian Pines dataset consists of 145 × 145 pixels from 200 spectral bands after removing the water absorption bands. There are sixteen classes of materials in the scene. For each of the 16 ground-truth classes, we randomly choose about 9% of labelled pixels as the dictionary, i.e. $\mathbf{D} \in \mathbb{R}^{200\times 957}$. The rest pixels are used for testing, i.e. $\mathbf{X}^{test} \in \mathbb{R}^{200\times 9409}$. Similar experiment settings can also be found in [4, 35, 36, 37] and [44, 45, 25, 27, 28, 29] with different training/test samples and accordingly non-identical performance.

A summary of the numbers of training and test pixels for individual classes is given in Table 4.4. The false colour of the image averaging through all the bands, the 16 ground-truth classes, the training set and the test set are shown in Figures 4.2(a)-4.2(d).

For a more reliable evaluation, we perform the experiments by 10 times of random training/test splits. For illustration, the optimal parameters obtained by LOOCV of one random training/test split are listed in Table 4.5. Note that the NNLS has no parameter to be tuned and hence no training process is required. For the OMP and NN-OMP, the tuned value of sparsity level L is 5; for the FC-NNLS, the tuned value of window size T is 25 (5×5); for the SOMP, the values of L and

T are tuned to be 30 and 81 (9×9), respectively; and for the proposed NN-SOMP, the value of L and T are tuned to be 15 and 25 (5×5), respectively.

Table 4.5: Settings of parameters for the Indian Pines dataset in one random training/test split. The values of parameters are determined by LOOCV. "NA" stands for "not applicable".

	NNLS	OMP	NN-OMP	FC-NNLS	SOMP	NN-SOMP
L	NA	5	5	NA	30	15
T	NA	NA	NA	25	81	25

4.6.4.1 Classification performances

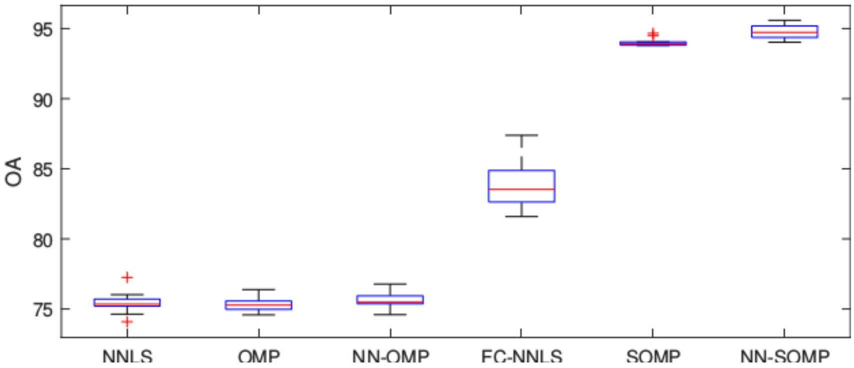


Figure 4.1: Boxplots of the overall classification accuracies (%) of 3 single models (CM (NNLS), SM (OMP), CSM (NN-OMP)) and 3 joint models (JCM (FC-NNLS), JSM (SOMP), C-JSM (NN-SOMP)) on the Indian Pines dataset.

The 10 overall classification *OA*s of all six compared methods are recorded and box-plotted in Figure 4.1. For illustration purposes, we also randomly choose one of the 10 classification results and list the *OA*, *AA* and κ coefficient of all methods in Table 4.6, and depict the classification maps of the corresponding methods in Figure 4.2(e)-4.2(j), respectively.

From Figure 4.1, we can observe two patterns. Firstly, we can observe that the proposed C-JSM (NN-SOMP) outperforms the other two joint models, JCM (FC-NNLS) and JSM (SOMP). Also, among the three single models, CSM (NN-OMP) performs the best, superior to CM (NNLS) and SM (OMP). These indicate that incorporating the non-negativity constraints into HSI classification can help to improve the performance of the sparse representation-based classifiers. Secondly, the proposed C-JSM (NN-SOMP) performs the best among all the compared methods. It indicates that combining the non-negativity constraints and the joint sparse representation can improve the classification performance the most, compared with the representation with any single constraint, i.e. joint representation, sparse representation or non-negative representation.

Table 4.6: The Indian Pines dataset: Ground-truth label and the classification accuracies (%) obtained by CM (NNLS), SM (OMP), CSM (NN-OMP), JCM (FC-NNLS), JSM (SOMP) and C-JSM (NN-SOMP), respectively. The best performance is indicated in **bold**.

Class	CM	SM	CSM	JCM	JSM	**C-JSM**
1	75.51	53.06	51.02	83.67	71.43	**85.71**
2	72.66	62.98	62.83	83.26	**94.24**	93.86
3	37.65	62.62	63.14	56.67	88.90	**92.87**
4	48.11	40.57	41.51	79.25	**92.45**	88.21
5	85.81	94.90	94.68	94.24	93.79	**98.00**
6	96.31	93.36	93.22	**99.71**	98.97	98.38
7	4.35	78.26	78.26	0	69.57	**100**
8	98.20	95.05	95.05	**100**	99.77	99.77
9	22.22	55.56	55.56	0	0	**72.22**
10	36.63	72.47	73.49	42.78	80.55	**95.45**
11	89.29	74.16	74.03	**99.06**	95.98	96.21
12	61.04	54.76	54.04	84.56	**91.38**	88.51
13	98.96	99.48	91.67	**99.48**	**99.48**	83.33
14	98.98	92.68	92.34	**99.74**	98.89	97.70
15	44.64	46.38	47.83	58.84	**99.71**	94.20
16	87.21	88.37	88.37	**100**	96.51	89.53
OA	75.42	74.79	74.83	84.88	93.79	**95.19**
AA	66.10	72.79	72.80	73.83	84.06	**92.64**
κ	0.714	0.713	0.713	0.824	0.929	**0.945**

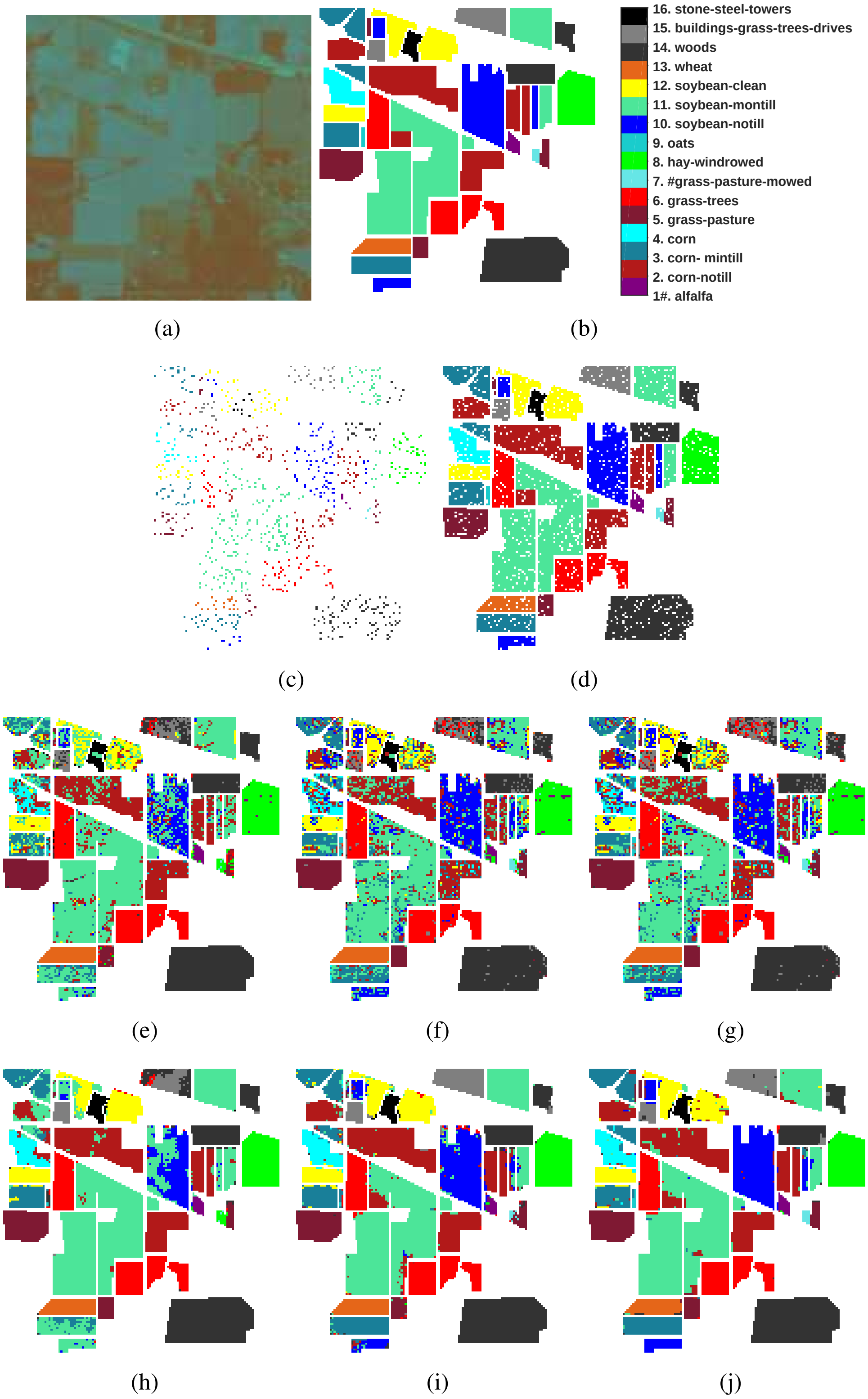
The one time classification results listed in Table 4.6 also show that the proposed C-JSM (NN-SOMP) outperforms other methods, which is aligned with our findings from the 10 times repeated random splits (Figure 4.1). We also notice two special cases, with class 7 and class 9, that the numbers of training samples are extremely small, i.e. 3 for class 7 and 2 for class 9, as listed in Table 4.4. All methods except for C-JSM (NN-SOMP) do not perform very well on classifying these two tiny classes of HSI pixels. For the single models, i.e. CM (NNLS), SM (OMP) and CSM (NN-OMP), the bad performances may be due to the lack of training samples. For the joint models of JCM (FC-NNLS) and JSM (SOMP), the performances are even worse. Particularly in class 9, the classification accuracies of both models are 0. This is because class 7 and class 9 cover narrow regions in the Indian Pines HSIs (as shown in Figure 4.2). The label of the central test pixel can be dominated by classes adjacent and thus misclassified. However, the proposed C-JSM (NN-SOMP) relives this spatial-over-smoothness caused by the local window strategy and outperforms the other five methods with substantial improvements: achieving 100% against the second best 78.26% for class 7 and achieving 72.22% against the second best 55.56% for class 9.

4.6.4.2 Effects of parameters

We further investigate the effects of tuning parameters on the performance of our proposed C-JSM (NN-SOMP). A sweep of the parameter space of sparsity level L and window size T is performed during the training phase. The sparse level L is tuned from 5 to 80 and the window size T ranges from 1 to 289 (17×17). The LOOCV result of C-JSM (NN-SOMP) is depicted in Figure 4.3(a). Within the same parameter space (L and T), we also show the LOOCV result of JSM (SOMP) in Figure 4.3(b) for comparison.

As shown in Figure 4.3(a) and Figure 4.3(b), we can easily see that the surface plot of *OA*s for C-JSM (NN-SOMP) is much smoother than that of JSM (SOMP). It implies that C-JSM (NN-SOMP) is more stable than that of JSM (SOMP) in terms of the performance sensitivity to L and T. More specifically, we split the 3-D view of the *OA* surface of C-JSM (NN-SOMP) into two 2-D views, which are shown in Figure 4.4(a) and Figure 4.4(b). It can be observed that the window size T dominates the performance of C-JSM whereas the effect of sparsity level L on the classification performance is not as sensitive as T.

To further demonstrate the effect of sparsity level L, we perform classification on one of the 10 randomly split test dataset, by fixing the window size T to be 25 (5×5) as tuned by LOOCV. This test dataset is the same as the one used in Table 4.6 and Figure 4.2. We set the level of sparsity L from 5 to 80 and depict the obtained *OA*s in Figure 4.5(a). Accordingly, we record the real sparsity L' obtained



(a) (b)

(c) (d)

(e) (f) (g)

(h) (i) (j)

Figure 4.2: The Indian Pines dataset: (a) mean image shown in the false colour; (b) ground-truth labels; (c) training set (9% pixels randomly chosen); (d) test set. Classification maps of (e) CM (NNLS), *OA* = 75.42; (f) SM (OMP), *OA* = 74.79; (g) CSM (NN-OMP), *OA* = 74.83; (h) JCM (FC-NNLS), *OA* = 84.88; (i) JSM (SOMP), *OA* = 93.79; (j) C-JSM (NN-SOMP), *OA* = 95.19.

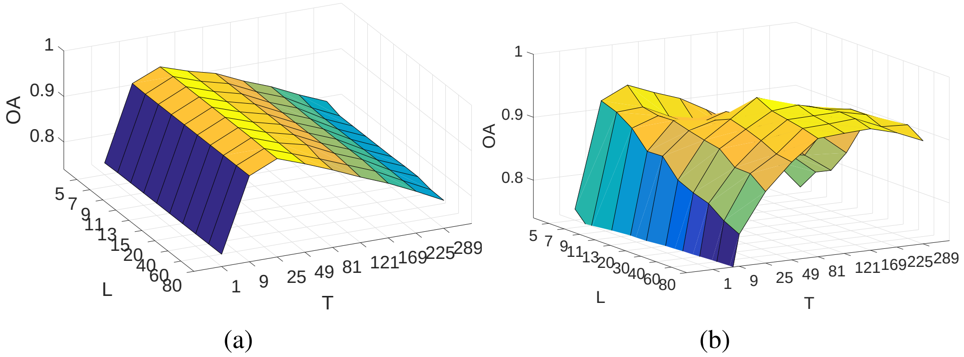


Figure 4.3: Overall classification accuracies over window size T and sparsity level L for (a) the proposed C-JSM (NN-SOMP) and (b) the JSM (SOMP) on the Indian Pines training dataset via LOOCV.

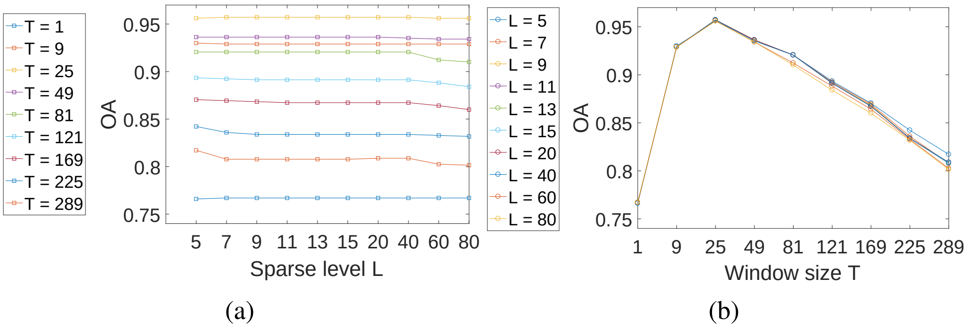
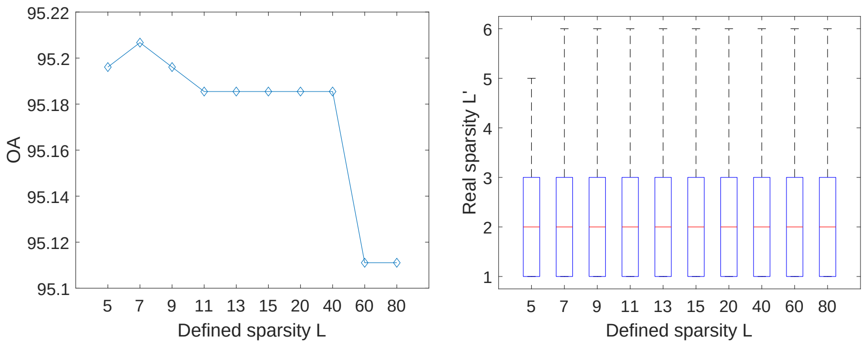


Figure 4.4: Effects of the sparsity L and window size T on the performance of the proposed C-JSM corresponding to Figure 4.3(a).

in different settings of L. Since different test HSI pixels have different real sparsities L' under a defined L, we record and box-plot them in Figure 4.5(b).

It can be seen that, although the best OA occurs at $L = 7$ when T is fixed to be 25, the performance only changes slightly with the defined sparsity L, where the OA changes only from 95.11% to 95.21%. Therefore the OA = 95.19% of C-JSM (NN-SOMP) listed in Table 4.6 with $L = 5$ is in the range of the stable performance, where the parameters are tuned by LOOCV and the testing results are reliable.

On the other hand, we can observe that the obtained sparsity L' ranges from 1 to 6, and the median value of them is around 2 no matter what values the defined sparsity L are. Furthermore, the obtained maximal sparsity L' converges to 6 when the defined sparsity L is over 6, as shown in Figure 4.5(b). This explains why the performance of C-JSM (NN-SOMP) is not so sensitive to the setting of sparsity L.



(a) (b)

Figure 4.5: Window size $T = 5$ on the test dataset of Indian Pines: (a) classification performance (overall accuracies) with sparsity level L; (b) the real sparsity level L' obtained from the test results with sparsity level L.

However, the setting of sparsity L still gives some room for each test HSI pixel to adaptively choose their optimal sparsity level and hence can achieve a stable and reliable classification performance.

4.6.4.3 Sparseness and non-negativity

We next demonstrate the effects of sparsity and non-negativity on all the compared methods by adopting a similar presentation in [61] and depicting the results in Figs. 4.7-4.8. The classification results of all methods are obtained in parameter settings listed in Table 4.5. For comparative purposes, we randomly select two test HSI pixels which belong to class 10 and are located in (48, 31) and (53, 88): one is correctly classified by all six methods and the other is only correctly classified by C-JSM (NN-SOMP).

For pixel (48, 31), the associated class-wise residuals obtained by all six methods are shown in Figure 4.6. We can observe that the pixel is correctly classified by all six methods into class 10, which has the minimum residuals and is indeed the ground-truth class. Among the six methods, CSM in Figure 4.6(c) and C-JSM in Figure 4.6(f), both of which contains both sparse and non-negative constraints, perform the best with the true class (with the smallest residual) and the most stable relative to other classes (with all large residuals).

To investigate further, we plot the obtained coefficients of this pixel in Fig-

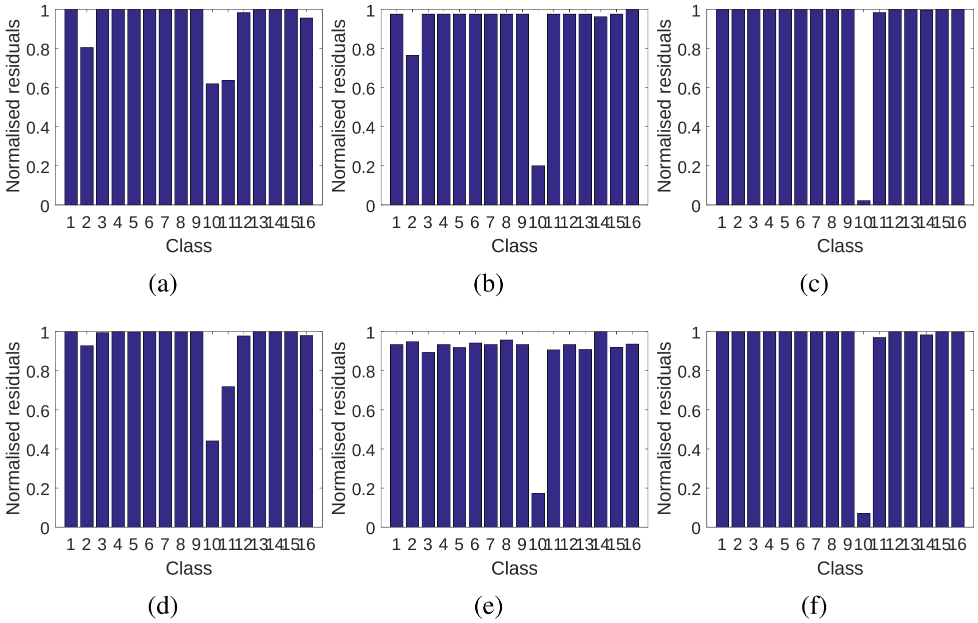


Figure 4.6: Normalised residuals for each class for the pixel located at (48, 31) by (a) CM, (b) SM, (c) CSM, (d) JCM, (e) JSM and (f) C-JSM. The ground-truth label is class 10. The test pixel is correctly identified by all six methods.

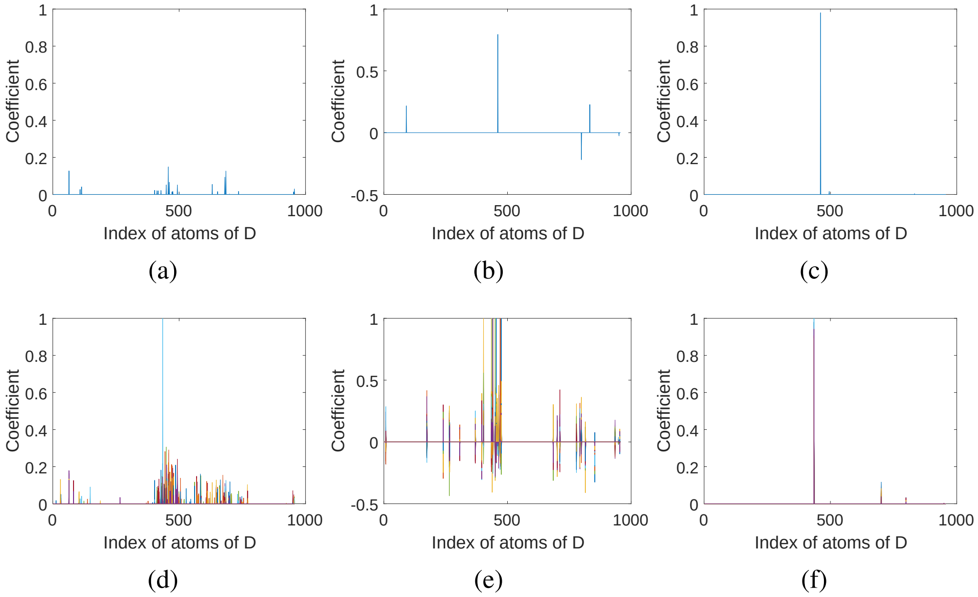


Figure 4.7: Estimated coefficients for the pixel located at (48, 31) by (a) CM, (b) SM, (c) CSM, (d) JCM, (e) JSM and (f) C-JSM. The ground-truth label is class 10. The test pixel is correctly identified by all six methods.

ure 4.7. Because for the single models (CM, SM and CSM) there is only one coefficient vector for the test HSI pixel $\mathbf{x}$, there is only one colour shown in the plots of coefficients; whereas for the joint models (JCM, JSM and C-JSM), the label of the central test pixel $\mathbf{x}_c$ is jointly determined by its local window $\mathbf{X}$, hence we plot all the coefficient vectors of the pixels in the window in different colours. In addition, since the test HSI pixel actually belongs to class 10, we expect to see that the coefficients mainly lie within the sub-dictionary of class 10, where the atom indices range from 402 to 490.

From Figure 4.7, we can observe that, although all methods can identify the correct class 10 for pixel (48, 31), the coefficient vectors obtained by different methods are remarkably different, and again, the most neat (sparse) performances are with C-JSM (Figure 4.7(f)) and CSM (Figure 4.7(c)). This also indicates that incorporating the non-negativity constraint into the sparse model is beneficial, which can produce a more sparse representation.

However, the sparse and non-negative constraints are not the only two factors that may ensure correct label identification for HSIs. As illustrated in Figure 4.8 and Figure 4.9, for a test HSI pixel located in (53, 88), only the proposed C-JSM identifies its label as class 10 correctly (Figure 4.8(f)). In C-JSM, the non-zero elements of the coefficients vectors of all pixels within the neighbourhood window mainly lie in class 10 and the label of the central test pixel is jointly determined by the minimal residuals, which belongs to class 10 (Figure 4.9(f). In contrast, although the coefficient vector obtained by CSM in Figure 4.9(c) is non-negative and most sparse, it lies in class 11, a wrong class (Figure 4.8(c)). This illustrates that the joint representation of neighbouring pixels on top of the sparsity and non-negativity can positively contribute to the classification performance for HSIs, and hence the proposed C-JSM outperforms others.

4.6.5 Real dataset: University of Pavia

The ROSIS University of Pavia dataset consists of 610×340 pixels from 103 spectral bands, with nine ground-truth labels. We randomly choose only 1% of labelled samples from each class for constructing the dictionary, i.e. $\mathbf{D} \in \mathbb{R}^{103 \times 432}$, and use

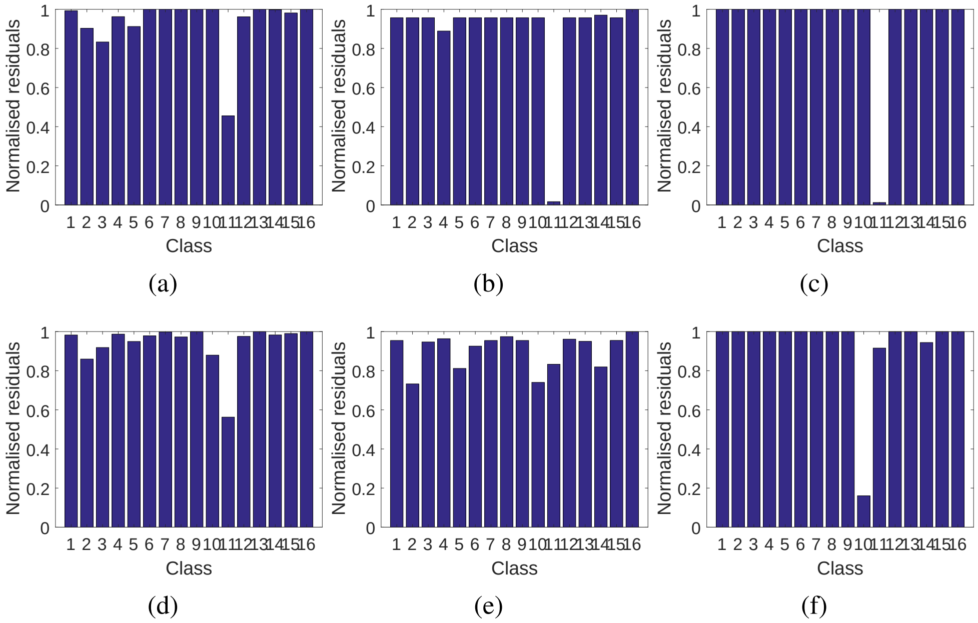


Figure 4.8: Normalised residuals for each class for the pixel located at (53, 88) by (a) CM, (b) SM, (c) CSM, (d) JCM, (e) JSM and (f) C-JSM. The ground-truth label is class 10. The test pixel is only correctly identified by our proposed C-JSM.

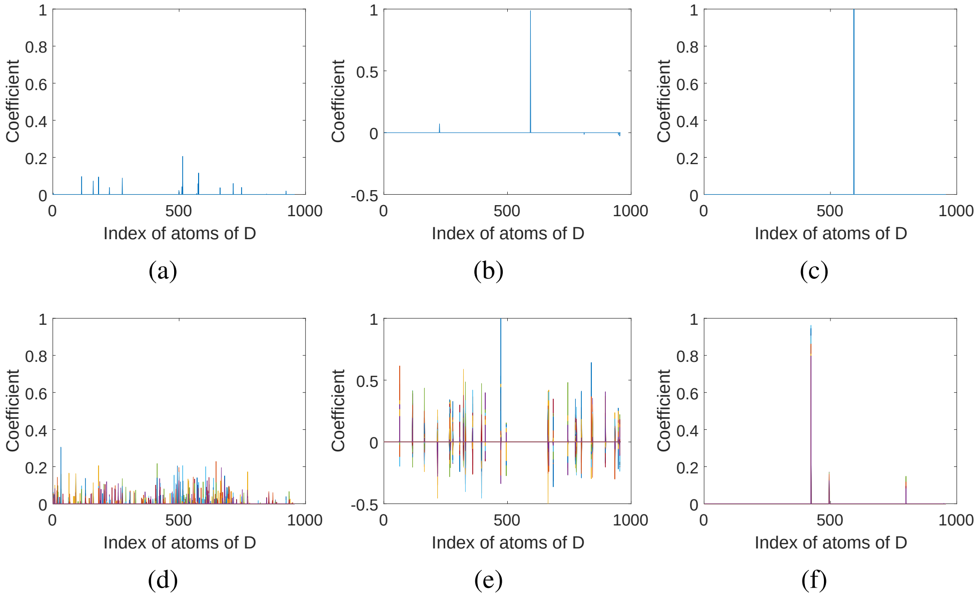


Figure 4.9: Estimated coefficients for the pixel located at (53, 88) by (a) CM, (b) SM, (c) CSM, (d) JCM, (e) JSM and (f) C-JSM. The ground-truth label is class 10. The test pixel is only correctly identified by our proposed C-JSM.

the rest HSI pixels for testing, i.e. $\mathbf{X}^{test} \in \mathbb{R}^{103\times 42344}$. A summary of this dataset is given in Table 4.7. Again, a false colour image averaging across all spectral bands, the nine ground-truth classes, the training set and the test set are shown in Figs. 4.11(a)-4.11(d).

Table 4.7: The Pavia University dataset: Ground-truth labels, class material, the training set and the test set.

Class	materials	Training	Test
1	Asphalt	67	6564
2	Meadows	187	18462
3	Gravel	21	2078
4	Trees	31	3033
5	Painted metal sheets	14	1331
6	Bare soil	51	4978
7	Bitumen	14	1316
8	Self-blocking bricks	37	3645
9	Shadows	10	937
Total		432	42344

Table 4.8: Settings of parameters for the University of Pavia dataset in one random training/test split. The values parameters are determined by LOOCV. “NA” stands for ”not applicable”.

	NNLS	OMP	NN-OMP	FC-NNLS	SOMP	NN-SOMP
L	NA	5	5	NA	10	3
T	NA	NA	NA	9	49	81

For a reliable evaluation, the experiments are also performed by 10 times random train/test splits, as with the Indian Pines dataset in section 4.6.4. For illustration, the optimal values of parameters tuned by LOOCV using one time train/test random spilt are listed in Table 4.8. The *OA*s of all six compared methods are boxplotted in Figure 4.10; we also randomly select one of 10 classification results and illustrate them in Table 4.9 and Figure 4.11(e)-4.11(j).

Once again, we can observe that the proposed C-JSM (NN-SOMP) outperforms other methods. We also note that in Figure 4.10 the performance of CM (NNLS) is better than that of SM (OMP) and of CSM (NN-OMP), a pattern different from the results shown for the Indian Pines dataset. As we have analysed in section 4.6.4.3, sparse and non-negative representations only may still be insufficient to

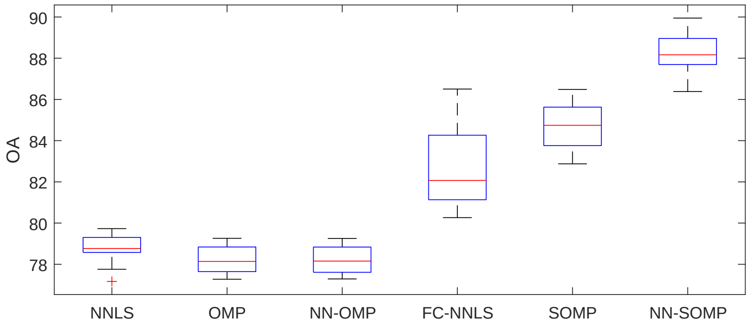


Figure 4.10: Boxplots of the overall classification accuracies (%) of CM (NNLS), SM (OMP), CSM (NN-OMP), JCM (FC-NNLS), JSM (SOMP) and C-JSM (NN-SOMP) on the University of Pavia dataset.

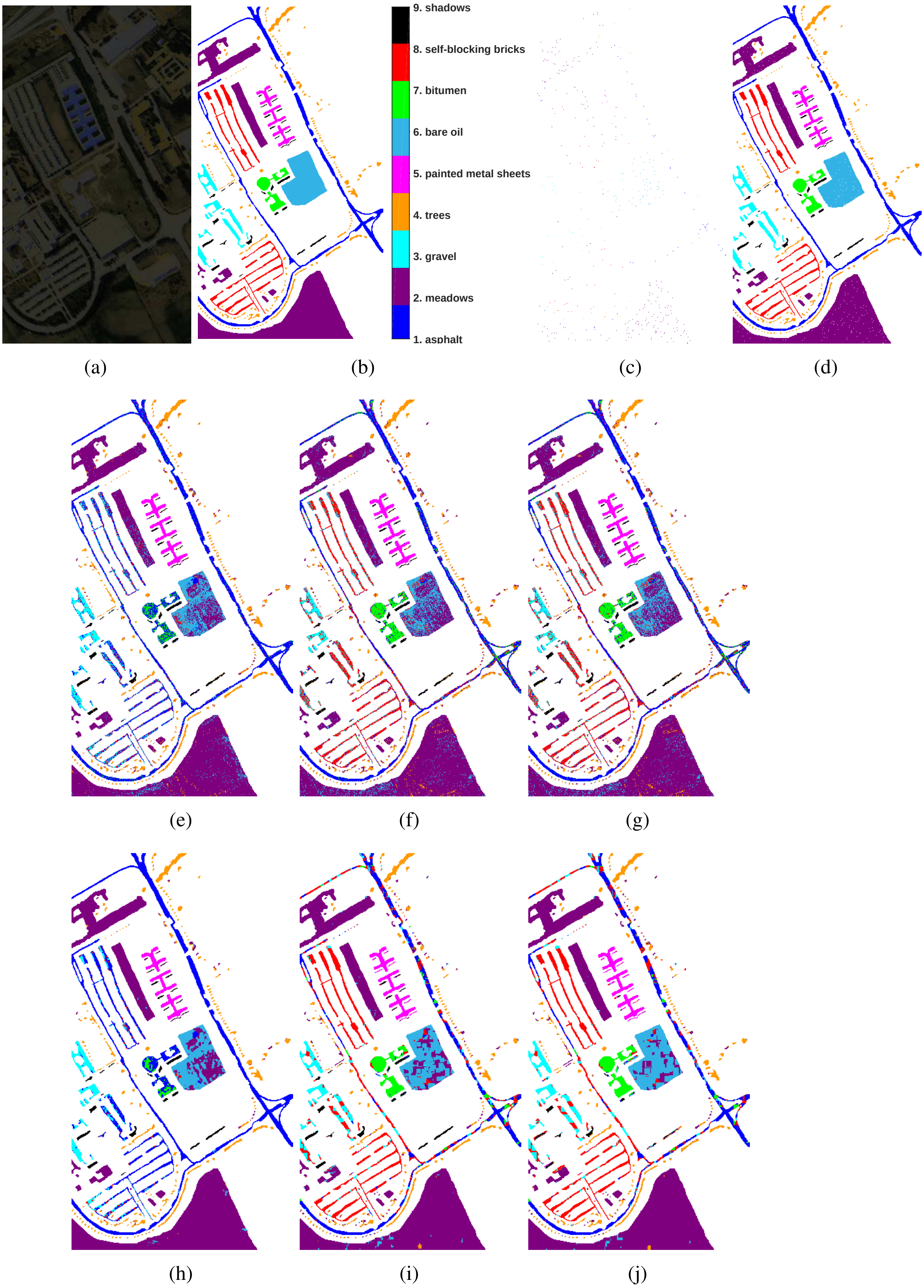
Table 4.9: The University of Pavia dataset: Ground-truth label and the classification accuracies (%) obtained by CM (NNLS), SM (OMP), CSM (NN-OMP), JCM (FC-NNLS), JSM (SOMP) and C-JSM (NN-SOMP), respectively. The best performance is indicated in **bold**.

Class	CM	SM	CSM	JCM	JSM	**C-JSM**
1	85.65	70.75	70.81	**98.92**	57.46	59.83
2	93.97	92.82	92.82	**99.40**	98.14	98.55
3	62.70	45.62	45.62	69.30	70.12	**77.48**
4	87.97	77.28	77.05	**93.14**	80.25	83.65
5	99.77	99.25	99.25	**100.00**	**100.00**	**100.00**
6	58.00	47.91	47.89	62.82	70.65	**80.63**
7	42.33	77.43	77.43	28.12	92.63	**95.74**
8	21.10	74.29	74.29	7.05	93.94	**95.34**
9	87.41	88.26	89.97	92.74	72.89	31.06
OA	78.65	78.72	78.75	82.81	84.91	**86.53**
AA	70.99	74.85	75.01	72.39	**81.79**	80.25
κ	0.712	0.714	0.714	0.765	0.798	**0.820**

produce a stable and correct classification. On the other hand, C-JSM incorporates the sparse and non-negativity constraints into the joint modelling of neighbouring pixels, and hence is capable of providing a more sparse representation and a more stable classification performance.

4.6.6 Running time comparison

We present the time costs for executing the compared algorithms. All experiments are performed by a Intel i7-3370 CPU using single thread on the platform of MATLAB R2016b. Table 4.10 shows the running time of each method for the Indian



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

(e) (f) (g)

(h) (i) (j)

Figure 4.11: The University of Pavia dataset: (a) mean image shown in the false colour; (b) ground-truth labels; (c) training set (1% pixels randomly chosen); (d) test set. Classification maps of (e) CM (NNLS), *OA* = 78.65; (f) SM (OMP), *OA* = 78.72; (g) CSM (NN-OMP), *OA* = 78.75; (h) JCM (FC-NNLS), *OA* = 82.81; (i) JSM (SOMP), *OA* = 84.91; (j) C-JSM (NN-SOMP), *OA* = 86.53.

Pines dataset. The time is recorded as second per HSI pixel.

Table 4.10: Running time (sec/pixel) spent on testing the Indian Pines dataset, settings of which are shown in Table 4.4 and Table 4.5 for 9409 test pixels.

	NNLS	OMP	NN-OMP	FC-NNLS	SOMP	NN-SOMP
Time	0.0058	0.0175	0.0017	0.1195	0.0737	0.0392

First, we can observe that, among the single models, NN-OMP takes less time than NNLS and OMP. In fact the obtained coefficients of NN-OMP are more sparse than the others, as indicated by Figure 4.7(a)-4.7(c) and Figure 4.9(a)-4.9(c). It implies that the computational burden is lessened by NN-OMP. Secondly, among the joint models, our proposed NN-SOMP is more time-efficient than FC-NNLS and SOMP. It is also because the obtained coefficients from NN-SOMP are more sparse than the others, as indicated by Figure 4.7(d)-4.7(f) and Figure 4.9(d)-4.9(f), and hence the computational costs are reduced.

4.6.7 Further remarks

It is worth noting that several literatures have studied the relationship between the sparsity and non-negativity [55, 62]. It has been shown that the non-negative least squares (NNLS) may be able to produce sufficient sparse recovery, without further imposing the sparse regularisations. However, we remark that this does not imply that the performances of NNLS and the sparsely regularised NNLS are the same, particular for the classification problems that are the focus of this chapter. That is, the distinct classification performances of the compared methods of different constraints in this chapter do not conflict the existing findings in [55, 62].

4.7 Conclusion and future work

To sum up, by considering the non-negativity of coefficients for the jointly sparse representation of HSI pixels, a new model called cone-based joint sparse model (J-CSM) has been proposed in this chapter. To solve the C-JSM, a new algorithm, called non-negative simultaneous orthogonal matching pursuit (NN-SOMP), has also been proposed. The C-JSM incorporates the non-negativity of coefficients, as well as the spatial coherence of the HSI pixels, into one model, yielding a more sparse and stable representation for the test HSI pixel whose label is jointly determined by its neighbouring pixels. As a result, the classification performance of the JSM is enhanced by the proposed C-JSM.

We notice that the proposed C-JSM may not completely solve the problems that are caused by the local window scheme. Specifically, the square shape of the window adopted in this chapter indeed introduces bias into the joint models and may cause spatial-over-smoothness. That is, the classification of the HSI pixel may not have a promising edge-preserving performance. As aforementioned in the introduction (section 4.1), several literatures have studied the improvement of the JSM by adopting size/shape adaptive windows [25, 27, 28, 29]. The proposed C-JSM can also be collaboratively conducted with the window adaptation strategies for enhancing the classification performance. On the other hand, it is also desired to exploit the non-linearity representation, such as kernelisation [44, 45], of the HSIs together with the non-negativity constraints for the joint sparse models. These two directions are our future research on the proposed C-JSM.

4.8 Appendix

The FC-NNLS algorithm is summarised in Algorithm 6.

Algorithm 6 The FC-NNLS algorithm [58] to solve (4.5).

```
Input: • Dictionary $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_N] \in \mathbb{R}^{B \times N}$.
       • A test window $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{R}^{B \times T}$.
Output: An estimated non-negative coefficient matrix $\mathbf{A}^* = [\alpha^*_{nt}] \in \mathbb{R}^{N \times T}$, where $n = 1, \dots, N$, and $t = 1, \dots, T$.
  Initialisation:
    • Initialise the columns of solution matrix: $\mathscr{M} := \{1, \dots, T\}$ and the rows of the solution matrix $\mathscr{N} := \{1, \dots, N\}$.
    • Pre-compute constant parts of the pseudo-inverse, e.g., $\mathbf{W} = [w_{nn}] = \mathbf{D}^T\mathbf{D}$ and $\mathbf{Q} = [q_{nt}] = \mathbf{D}^T\mathbf{X}$.
    • Compute
        $\hat{\mathbf{A}} = \underset{\mathbf{A}}{\text{argmin}} \left\|\mathbf{X} - \mathbf{D}\mathbf{A}\right\|_F,$    (4.20)
      where $\hat{\mathbf{A}}$ is the unconstrained estimate for $\mathbf{A}$ and $\hat{\mathbf{A}} = [\alpha_{nt}] \in \mathbb{R}^{N \times T}$.
    • Initialise the set of passive sets: $\mathbf{P} = [p_{nt}]$, where $p_{nt} = 1$ if $\alpha_{nt} > 0$ and 0 otherwise.
    • Find the set of columns that yet to be optimised: $\mathscr{F} = \{t \in \mathscr{M} : \sum_n p_{nt} \neq N\}$.
    • Compute the overwriting solution:
        $\alpha_{nt} \leftarrow \begin{cases} \alpha_{nt} \text{ if } p_{nt} = 1 \\ 0, \text{ otherwise.} \end{cases}$    (4.21)
  while $\mathscr{F} \neq \varnothing$ do
    (1) $\mathbf{P}_{\mathscr{F}} = [p_{\cdot F}] \in \mathbb{R}^{N \times |F|}$, $\mathbf{Q}_{\mathscr{F}} = [q_{\cdot F}] \in \mathbb{R}^{N \times |F|}$;
    (2) Compute $\min_{\mathbf{A}_{\mathscr{F}}} \|\mathbf{X}_{\mathscr{F}} - \mathbf{D}\mathbf{A}_{\mathscr{F}}\|_F$ using the CSSLS (Algorithm 7) and the passive set $\mathbf{P}_{\mathscr{F}}$;
    (3) Put indices of columns with negative variables into set $\mathscr{H} = \{t \in \mathscr{F} : \min_{n \in \mathscr{N}}\{\alpha_{nt}\} < 0\}$.
    while $\mathscr{H} \neq \varnothing$ do
      $\forall i \in \mathscr{H}$, select the variables to move out of the passive set $\mathbf{P}$;
      $\mathbf{P}_{\mathscr{H}} = [p_{\cdot H}] \in \mathbb{R}^{N \times |H|}$, $\mathbf{Q}_{\mathscr{H}} = [q_{\cdot H}] \in \mathbb{R}^{N \times |H|}$;
      Compute $\min_{\mathbf{A}_{\mathscr{H}}} \|\mathbf{X}_{\mathscr{H}} - \mathbf{D}\mathbf{A}_{\mathscr{H}}\|_F$ using the CSSLS (Algorithm 7) and the passive set $\mathbf{P}_{\mathscr{H}}$;
      $\mathscr{H} = \{t \in \mathscr{F} : \min_{n \in \mathscr{N}}\{\alpha_{nt}\} < 0\}$.
    end while
    $\mathbf{V} = [v_{nt}] = \mathbf{Q}_{\mathscr{F}} - \mathbf{W}\mathbf{A}_{\mathscr{F}}$, where $\mathbf{A}_{\mathscr{F}} = [\alpha^*_{\cdot F}]$.
    Record the sets of columns whose solutions are optimal: $\mathscr{Z} = \{t \in \mathscr{F} : \sum_n v_{nt}(1 - \mathbf{P}_{\mathscr{F}})_{nt} = 0\}$.
    Remove the optimised columns $\mathscr{Z}$ from $\mathscr{F}$: $\mathscr{F} \leftarrow \mathscr{F} \setminus \mathscr{Z}$.
    $p_{nt} = \left\{ \begin{array}{c} 1, \text{ if } n = \text{argmax}_n\{v_{nt}(1 - \mathbf{P}_{\mathscr{F}})_{nt}\}, \forall t \in \mathscr{F} \\ p_{nt}, \text{ otherwise} \end{array} \right\}.$
  end while
```

Algorithm 7 The CSSLS algorithm [58].

```
Input: $\mathbf{W} \in \mathbb{R}^{N\times N}$; $\mathbf{Q} \in \mathbb{R}^{N\times K}$; $\mathbf{P} \in \mathbb{R}^{N\times K}$.
Output: $\mathbf{A}$.
  $\mathscr{M} := \{1,\dots,K\}$; $\mathscr{N} := \{1,\dots,N\}$; $\mathbf{P} = [\mathbf{p}_1,\dots,\mathbf{p}_K]$.
  Find the set of $S$ unique columns in $\mathbf{P}$: $\mathbf{U} = [\mathbf{u}_1,\dots,\mathbf{u}_S] = \text{unique}\{\mathbf{P}\}$.
  Find the columns in $\mathbf{A}$ with identical passive sets $\mathbf{g}_j = \{t \in \mathscr{M} : \mathbf{p}_t = \mathbf{u}_j\}$.
  for $j = 1,\dots,S$ do
      $\mathbf{A}_{\mathbf{u}_j,\mathbf{g}_j} = \mathbf{W}^{-1}_{\mathbf{u}_j,\mathbf{u}_j}\mathbf{Q}_{\mathbf{u}_j,\mathbf{g}_j}$.
  end for
```

Part II

Contributions to HSI Target Detection

Chapter 5

HSI Target Detection: Matched Subspace Detector with Interaction Effects (MSDinter)

5.1 Introduction

Hyperpsectral target detection aims to detect small objects from the background of a hyperspectral image (HSI) by the use of known target spectra. The number of target pixels is relatively very small compared with the total number of pixels in an HSI, e.g. only a few target pixels in millions of pixels. Typical applications of the HSI target detection include the detection of specific terrain features, minerals and crops for resource management, the detection of military vehicles and aeroplanes for defence, etc. Comprehensive overviews and gentle tutorials of the HSI target detection can be found in [13, 14, 15, 16].

Target detection algorithms are typically derived from the binary hypothesis model, which consists of two competing hypotheses: the H_0 (absence of target) hypothesis and the H_1 (presence of target) hypothesis. The likelihood ratio or the generalised likelihood ratio (GLR) of functions of target and background can be used to construct a detector.

Some well-known detectors have been successfully applied to the HSI target detection, including the matched subspace detector (MSD) [7], the orthogonal subspace projection detector (OSP) [18], the spectral matched filter (SMF) [63, 64], the adaptive coherence/cosine detectors (ACEs) [21, 22] and the constrained energy minimization (CEM) [19]. Kwon et al. [65] also extend the MSD, OSP, SMF and ACEs to their corresponding kernel versions based on the kernel-based learning theory. Several methods have been developed based on the CEM specifically [66, 67, 68]. Yang et al. [66] utilise an inequality constraint on the output detector to solve the spectral variability problems, instead of the equal constraint on the CEM. A hierarchical structure of CEM [67] is proposed, which suppresses the backgrounds while preserving the target spectra to boost the performance of CEM. In a very recent work, Yang et al. [68] use total variation to constrain the spatial smoothness and show a promising detection performance when only one single target spectrum is available for training.

Sparse representation (SR)-based algorithms have also been applied to the HSI target detection [23, 24, 69, 70, 71, 72]. Chen et al. [23] propose a sparsity-based target detection (STD), linearly modelling a test pixel by the training background samples and the training target samples. Zhang et al. [24] propose an SR-based binary hypothesis model (SRBBH), which is in the similar fashion of the binary hypothesis model of the MSD. The kernel versions of the STD and SRBBH can be found in [69] and [70], respectively. Detailed reviews of SR algorithms for the HSI classification and detection can be found in [71, 72].

The assumption of these well-known detectors [7, 18, 21, 22, 23, 24, 63, 64] is the linear mixing model (LMM) [5]. The LMM assumes that the spectrum of a mixed pixel can be represented as a linear combination of component spectra (endmembers). The weight (abundance) of each endmember spectrum is proportional to the fraction of the pixel area covered by the endmember.

For the HSI target detection, the underlying physical assumption of the LMM is that each incident photon interacts with one earth surface component only and the reflected spectra do not mix before entering the sensor. Therefore, adopting the LMM in [7, 18, 21, 22, 23, 24, 63, 64] assumes that the target spectral signature in the scene remains linearly mixed with the surrounding background spectra after entering the sensor. However this is not true in practice, since the target spectral signatures captured by the hyperspectral sensor can appear significantly different from the true underlying spectrum. The exhibited target spectrum may be contaminated by the *interaction effect* of its true underlying spectrum and its surrounding environments. The reasons can be, but not limited to, that the sensor picks up the signal from multiple scattering of photons and as a result, the abundance vector of targets will be dependent on the characteristics of their surrounding background.

To cope with multiple scattering problems and to model interaction effects, the bilinear mixing model (BMM) has been proposed in the hyperspectral analysis, particularly for the unmixing applications [73, 74, 75, 76, 77, 78]. Nascimento et al. [73] and Fan et al. [74] address the HSI unmixing problem by taking into account of the second-order scattering interaction between endmembers, referred to as “Nascimento model” and “Fan model” hereafter, respectively. The two models are distinguished by different sum-to-one constraints imposed on the abundances. Halimi et al. [75] propose a generalised bilinear model (GBM) to unmix an HSI pixel and solve the problem by a hierarchical Bayesian algorithm. Practical analysis [76, 77, 78] also demonstrate impacts of different orders of interactions in real HSI mixing problems, such as tree cover estimates in orchards. It shows that the second-order interaction has the most significant effect of nonlinear mixing and the higher order interactions can be neglected. On top of the BMM, Heylen et al. [79] derive a multilinear mixing model (MLM) which extends the BMM to an infinite orders of interactions. Experimental studies in [73, 74, 75, 76, 77, 78, 79] have been carried out and shown superior performance of the above-mentioned nonlinear mixing models to conventional linear mixing models.

In this chapter, to account for the effect of interaction between the target and their surrounding background on the target spectral signature captured by the sensor, we propose to introduce interaction effects into the models for the HSI target detection. Specifically, we propose a new model, termed the matched subspace detector with interaction effects (MSDinter), by introducing the terms that describe the interaction effects between the target and its surrounding background. To our knowledge, such model is the first one proposed for the HSI target detection. The proposed MSDinter model is able to capture better the target-background mixing effects within pixel spectrum and therefore can improve the performance of target detection.

5.2 Matched Subspace Detector (MSD)

The matched subspace detector (MSD) [7] is a popular algorithm which explores the idea of the LMM binary hypothesis model (2.10). The task is to determine if a test pixel $\mathbf{x}$ contains materials characterised by exemplar target spectral signatures, i.e. whether the test pixel can be represented by a linear combination of target spectral signatures and background spectral signatures. In the MSD, the target spectral signatures and background spectral signatures are represented by the bases of a target subspace and the bases of a background subspace, respectively. The underlying assumption of the MSD in the HSI target detection is that each basis vector of these subspaces represents an endmember, which follows the assumption in the LMM (2.9).

When a target pixel presents, the spectrum of an observed pixel $\mathbf{x} \in \mathbb{R}^p$ can be decomposed into two components under the LMM assumption, as

$$\mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{n}_1, \tag{5.1}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}]$ is a $p \times r_t$ matrix representing the target subspace, and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$ is a $p \times r_b$ matrix representing the background subspace; $\mathbf{T}$ is derived from a training target matrix $\mathbf{M}_T \in \mathbb{R}^{p \times N_t}$ whose columns are the N_t target spectra $\mathbf{M}_T(\cdot, n_t)$ for $n_t = 1, \ldots, N_t$, respectively; $\mathbf{B}$ is derived from a training background matrix $\mathbf{M}_B \in \mathbb{R}^{p \times N_b}$ whose columns are the N_b background spectra $\mathbf{M}_B(\cdot, n_b)$ for $n_b = 1, \ldots, N_b$, respectively; γ and β are the corresponding abundance vectors of the subspace $\mathbf{T}$ and the subspace $\mathbf{B}$, respectively; and $\mathbf{n}_1$ is the additive Gaussian white noise.

When the target is absent, the spectrum of the observed pixel is adequately described by

$$\mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \tag{5.2}$$

which is a reduced order model. Therefore, to decide whether a given target is present or not, we can fit the full model and the reduced model to the test pixel spectrum and check which model provides a better fitting according to certain criterion. Formulated as a binary hypothesis test, the detection problem becomes a decision between the two competing hypotheses H_0 and H_1 and has been shown in (2.10),

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \text{ target absent,} \\ &H_1 : \mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{n}_1, \text{ target present.} \end{aligned}$$

Model (2.10) is defined as the MSD model. Using the generalised likelihood ratio test (GLRT) [15], the output detector of the MSD model is given by (2.11)

$$D_{\text{MSD}}(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{P}_B^{\perp} \mathbf{x}}{\mathbf{x}^T \mathbf{P}_V^{\perp} \mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu,$$

where $\mathbf{P}_B^{\perp} = \mathbf{I} - \mathbf{P}_B$ with $\mathbf{P}_B = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$ being the projection matrix onto the column space of $\mathbf{B}$; and $\mathbf{P}_V^{\perp} = \mathbf{I} - \mathbf{P}_V$ with $\mathbf{P}_V = \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$ being the projection matrix onto the column space of $\mathbf{V}$, where $\mathbf{V}$ is a $p \times (r_t + r_b)$ concatenated matrix of $\mathbf{T}$ and $\mathbf{B}$, i.e. $\mathbf{V} = [\mathbf{T}, \mathbf{B}]$.

As explained in section 2.3.1.1, the value of $D_{\text{MSD}}(\mathbf{x})$ is compared to a threshold ν to make a final decision of which hypothesis should be rejected for test pixel $\mathbf{x}$. In general, any set of orthogonal basis vectors that spans the corresponding subspace can be used as the column vectors of $\mathbf{B}$ and $\mathbf{T}$. In this chapter, the significant eigenvectors (normalised by the square roots of their corresponding eigenvalues) of the background and target covariance matrices $\mathbf{C}_b$ and $\mathbf{C}_t$ are used to create the column vectors of $\mathbf{B}$ and $\mathbf{T}$, respectively.

5.3 The Matched Subspace Detector with interaction effects (MSDinter)

The linear model (5.1) in the MSD assumes that the abundance vector γ of the target subspace $\mathbf{T}$ in composing a target pixel $\mathbf{x}$ will not change if the characteristics of the background change. Specifically, the effect of one-unit change of $\mathbf{T}$ on $\mathbf{x}$ is the marginal effect of targets $\mathbf{T}$ on $\mathbf{x}$. The marginal effect is obtained by differentiating the conditional expected value of $\mathbf{x}$ with respect to $\mathbf{T}$, i.e.

$$\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{r_t} \end{bmatrix}_{(pr_t)\times p}, \tag{5.3}$$

where

$$\Gamma_i = \begin{bmatrix} \gamma_i & 0 & \dots & 0 \\ 0 & \gamma_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma_i \end{bmatrix}_{p\times p} = \gamma_i \mathbf{I}_p,\ i = 1,\dots,r_t, \tag{5.4}$$

and $\mathbf{I}_p$ denotes the $p \times p$ identity matrix. The details of the derivation are shown in section 5.6 of Appendix.

That is, $[\Gamma_1,\dots,\Gamma_{r_t}]^T \in \mathbb{R}^{(pr_t)\times p}$ in (5.3) is the change of expected value of $\mathbf{x}$ induced by one-unit change of $\mathbf{T}$, which includes only the effect of $\mathbf{T}$ on $\mathbf{x}$, ignoring the effect of $\mathbf{B}$ on $\mathbf{x}$. In other words, no matter whether or not background spectra present in the subpixel $\mathbf{x}$ (i.e. $\beta = \mathbf{0}$ or $\beta \neq \mathbf{0}$), the marginal effect of $\mathbf{T}$ on the test pixel $\mathbf{x}$ does not depend on the values of $\mathbf{B}$.

However, in real applications of the HSI target detection, an observed HSI pixel will also receive multiple scattering of photons between its material and its neighbourhood materials, which the LMM cannot capture. The BMM has been introduced in the hyperspectral unmixing problems to accounts for the presence of multiple photon interactions [73, 74, 75, 76, 77, 78]. However, the interaction effects have not been studied in the hyperspectral target detection. To this end, we hypothesise that there are interaction effects of background spectra and the target spectrum on the composition of the spectrum of an observed target pixel. Therefore we introduce interaction terms into the LMM-based subspace model (2.9) and propose a new method called matched subspace detector with interaction effects (shortened as MSDinter).

5.3.1 The bilinear mixing model

As aforementioned, LMM (2.9) cannot deal with multiple scattering that often occurs in the real applications. To this end, the bilinear model (BMM) [73, 74, 75, 76, 77, 78] is proposed to model interaction effects of each pair of endmembers, so as to take account of the multiple scattering phenomena. A typical BMM called "Fan model" [74] is given by

$$\mathbf{x} = \mathbf{M}\mathbf{a} + \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \alpha_{i,j} \mathbf{m}_i \odot \mathbf{m}_j + \mathbf{n}, \tag{5.5}$$

where $\mathbf{M}$ is a $p \times K$ matrix whose columns are the K endmember spectra $\mathbf{m}_k \in \mathbb{R}^p$ for $k = 1, \ldots, K$; $\odot$ denotes the element-wise product operation between two vectors. It is defined as that for two vectors, $\mathbf{m}_i = [m_{i,1}, m_{i,2}, \ldots, m_{i,p}]^T$ and $\mathbf{m}_j = [m_{j,1}, m_{j,2}, \ldots, m_{j,p}]^T$ of the same length, in this case $p \times 1$, the element-wise product is still a vector of the same dimension as the operands with elements given by

$$(\mathbf{m}_i \odot \mathbf{m}_j)_l = m_{i,l} \cdot m_{j,l}, \text{ where } l = 1, \ldots, p. \tag{5.6}$$

So the element-wise product of two endmembers $\mathbf{m}_i$ and $\mathbf{m}_j$ is

$$\mathbf{m}_i \odot \mathbf{m}_j = \begin{pmatrix} m_{i,1} \\ \vdots \\ m_{i,p} \end{pmatrix} \odot \begin{pmatrix} m_{j,1} \\ \vdots \\ m_{j,p} \end{pmatrix} = \begin{pmatrix} m_{i,1} m_{j,1} \\ \vdots \\ m_{i,p} m_{j,p} \end{pmatrix}. \tag{5.7}$$

There are various BMMs with different definitions on the sum-to-one constraint to account for the hyperspectral unmixing problems. In the "Fan model" [74], it is assumed that $\sum_{k=1}^{K} a_k = 1$ and $\alpha_{i,j} = a_i a_j$, whereas in the "Nascimento model" [73], the sum-to-one constraint is based on $\sum_{k=1}^{K} a_k + \sum_{i=1}^{K-1} \Sigma_{j=i+1}^{K} \alpha_{i,j} = 1$. In the following proposed method, since we only care about the presence of the interactions terms, it does not matter whether the summation of abundance fractions is 1. Again with the explanations in the HSI target detection [5], we will relax the sum-to-one constraint as well as the non-negative constraint in the following proposed method to simplify the solution to target detection problems.

5.3.2 Formulations of MSDinter

As with the BMM (5.5), we introduce terms of the interaction between basis vectors of the background subspace **B** and the target subspace **T** into the MSD model (5.1), and then revise the alternative hypothesis H_1 of the MSD model (2.10). The proposed model with interaction effects is defined as follows:

$$\mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{H}\eta + \mathbf{n}_1, \tag{5.8}$$

where **H** is a matrix representing the interaction terms between **T** and **B**. We call the matrix **H** the interaction matrix, and η is the abundance vector for **H**.

The interaction matrix **H** is obtained by the element-wise product of each basis $\mathbf{t}_i$ and $\mathbf{b}_j$, where $i = 1, \ldots, r_t$ and $j = 1, \ldots, r_b$, of the subspace **T** and the subspace **B**, respectively. Similar to the element-wise production $\odot$ defined in (5.5), the element-wise product of two basis vectors $\mathbf{t}_i = [t_{i,1}, \ldots, t_{i,p}]^T$ and $\mathbf{b}_j = [b_{j,1}, \ldots, b_{j,p}]^T$ is defined as

$$\mathbf{t}_i \odot \mathbf{b}_j = \begin{pmatrix} t_{i,1} \\ \vdots \\ t_{i,p} \end{pmatrix} \odot \begin{pmatrix} b_{j,1} \\ \vdots \\ b_{j,p} \end{pmatrix} = \begin{pmatrix} t_{i,1} b_{j,1} \\ \vdots \\ t_{i,p} b_{j,p} \end{pmatrix}. \tag{5.9}$$

Hence, the interaction matrix **H** is formulated as

$$\mathbf{H} = [\mathbf{t}_1 \odot \mathbf{b}_1, \ldots, \mathbf{t}_1 \odot \mathbf{b}_{r_b}, \mathbf{t}_2 \odot \mathbf{b}_1, \ldots, \mathbf{t}_2 \odot \mathbf{b}_{r_b}, \ldots, \mathbf{t}_{r_t} \odot \mathbf{b}_1, \ldots, \mathbf{t}_{r_t} \odot \mathbf{b}_{r_b}], \tag{5.10}$$

which is a $p \times (r_t r_b)$ matrix. As a result, the abundance vector corresponding to **H** in (5.10) becomes

$$\eta = [\eta_{1,1}, \dots, \eta_{1,r_b}, \eta_{2,1}, \dots, \eta_{2,r_b}, \dots, \eta_{r_t,1}, \dots, \eta_{r_t,r_b}]^T, \tag{5.11}$$

which is a $(r_t r_b) \times 1$ vector.

In model (5.8), each basis vector in **T** and **B** is still assumed to represent an endmember. The column vectors in **H**, on the other hand, are assumed to represent the interactions between the corresponding basis vectors in **T** and **B**, respectively. The interaction matrix **H** in fact can be regarded as a generalisation of interaction terms $\mathbf{m}_i \odot \mathbf{m}_j$ defined in model (5.5).

Our proposed MSDinter is then modelled as follows:

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \text{ target absent,} \\ &H_1 : \mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{H}\eta + \mathbf{n}_1, \text{ target present.} \end{aligned} \tag{5.12}$$

For a simple representation, let **U** be the concatenated matrix of **T**, **B** and **H** (5.10), i.e.

$$\begin{aligned} \mathbf{U} &= [\mathbf{T}, \mathbf{B}, \mathbf{H}] \\ &= [\mathbf{t}_1, \dots, \mathbf{t}_{r_t}, \mathbf{b}_1, \dots, \mathbf{b}_{r_b}, \mathbf{t}_1 \odot \mathbf{b}_1, \dots, \mathbf{t}_{r_t} \odot \mathbf{b}_{r_b}], \end{aligned} \tag{5.13}$$

which is a $p \times (r_t + r_b + r_t r_b)$ matrix. Then the abundance vectors γ, β and η of model H_1 in (5.12) can be concatenated into a single vector, denoted as υ, i.e.

$$\upsilon = [\gamma^T, \beta^T, \eta^T]^T, \tag{5.14}$$

which is a $(r_t + r_b + r_t r_b)$-dimensional vector. Hence model H_1 in the proposed MSDinter (5.12) can be rewritten as

$$H_1 : \mathbf{x} = \mathbf{U}\upsilon + \mathbf{n}, \text{ target present,} \tag{5.15}$$

and thus the MSDinter model (5.12) becomes

$$\begin{aligned} H_0 &: \mathbf{x} = \mathbf{B}\beta + \mathbf{n}, \text{ target absent,} \\ H_1 &: \mathbf{x} = \mathbf{U}\upsilon + \mathbf{n}, \text{ target present.} \end{aligned} \tag{5.16}$$

To align with the MSD [7], we also adopt the least squares estimate (LSE) to solve the abundance vector β in H_0 and the abundance vector υ in H_1, respectively. Hence it is easily to see that the LSE of β is

$$\hat{\beta} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{x} \tag{5.17}$$

and the LSE of υ is

$$\hat{\upsilon} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{x}, \tag{5.18}$$

respectively.

Based on (5.17) and (5.18), the residual sums of squares (RSS) e_0 and e_1 given H_0 and H_1 of MSDinter (5.16) are computed as

$$H_0 : e_0 = \left\|\mathbf{x} - \mathbf{B}\hat{\beta}\right\|_2^2 = \mathbf{x}^T(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T)\mathbf{x}, \tag{5.19}$$

and

$$H_1 : e_1 = \|\mathbf{x} - \mathbf{U}\hat{\upsilon}\|_2^2 = \mathbf{x}^T(\mathbf{I} - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T)\mathbf{x}, \tag{5.20}$$

respectively, where $\mathbf{I}$ is a $p \times p$ identity matrix.

Therefore the generalised test ratio of the MSDinter model is then given by

$$D_{\text{MSDinter}}(\mathbf{x}) = \frac{e_0}{e_1} = \frac{\mathbf{x}^T(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T)\mathbf{x}}{\mathbf{x}^T(\mathbf{I} - \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T)\mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu. \tag{5.21}$$

Referring to the final results of MSD (2.11), we reformulate the output detector of the MSDinter model (5.21) by utilising the projection matrices. The numerator of (5.21) is the same as that of the MSD (2.11), where $\mathbf{P}_B = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$ is the projection matrix onto the subspace $\mathbf{B}$ spanned by the basis vectors $\mathbf{b_1}, \ldots, \mathbf{b}_{r_b}$ and $\mathbf{P}_B^{\perp} = \mathbf{I} - \mathbf{P}_B$ is the orthogonal complement of $\mathbf{P}_B$. The denominator of (5.21) can be derived in the same way, where

$$\mathbf{P}_U = \mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T \tag{5.22}$$

is the projection matrix onto the subspace **U** spanned by the column vectors in (5.13) and

$$\mathbf{P}_U^{\perp} = \mathbf{I} - \mathbf{P}_U, \tag{5.23}$$

is the orthogonal complement of $\mathbf{P}_\mathbf{U}$. Hence the final output detector of the MSDinter is formulated as

$$D_{\text{MSDinter}}(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{P}_B^{\perp}\mathbf{x}}{\mathbf{x}^T\mathbf{P}_U^{\perp}\mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu. \tag{5.24}$$

The value of $D_{\text{MSDinter}}(\mathbf{x})$ is compared with the threshold ν to make a final decision of which hypothesis should be rejected for the test pixel **x**.

5.3.3 Underlying assumption of adding interaction terms in target detection

In the proposed MSDinter model (5.12), we assume that the marginal effect of targets **T** on **x** varies in different surrounding backgrounds. Specifically, the abundance of target is not only γ when an interaction with the background presents. The abundance of the target can be decomposed into the main effect of γ plus a contribution from the interactions.

Differentiating the conditional expected value of **x** given model (5.8) with respect to **T**, we can obtain the following result:

$$\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{r_t} \end{bmatrix}_{(pr_t)\times p} + \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_{r_t} \end{bmatrix}_{(pr_t)\times p}, \tag{5.25}$$

where

$$\Pi_i = \begin{bmatrix} \mathbf{B}_{1,\cdot}^T \eta_i & 0 & \dots & 0 \\ 0 & \mathbf{B}_{2,\cdot}^T \eta_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{B}_{p,\cdot}^T \eta_i \end{bmatrix}_{p \times p} , i = 1, \dots, r_t, \tag{5.26}$$

which is a diagonal $p \times p$ matrix; η_i is an $r_b \times 1$ vector which is a segment of η (5.11) with

$$\eta = [\eta_1^T, \dots, \eta_i^T, \dots, \eta_{r_t}^T]^T \tag{5.27}$$

where

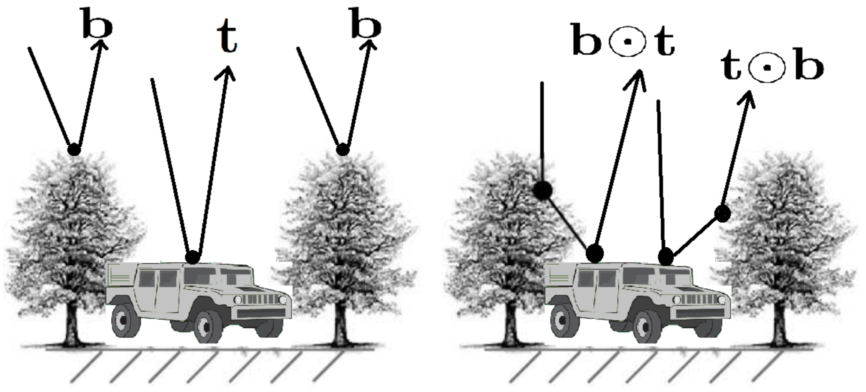
$$\eta_i = [\eta_{i,1}, \dots, \eta_{i,r_b}]^T; \tag{5.28}$$

and $\mathbf{B}_{l,\cdot}$ denotes a column vector representing the lth row of matrix $\mathbf{B}$. The details of the derivation are also presented in section 5.6 of Appendix.

In (5.25), when $\eta = \mathbf{0}$, the marginal effect of targets $\mathbf{T}$ on an observed test pixel $\mathbf{x}$ is $[\Gamma_1, \dots, \Gamma_{r_t}]^T \in \mathbb{R}^{(pr_t) \times p}$ only; when $\eta \neq \mathbf{0}$, the marginal effect is $[\Gamma_1, \dots, \Gamma_{r_t}]^T + [\Pi_1, \dots, \Pi_{r_t}]^T \in \mathbb{R}^{(pr_t) \times p}$. In other words, the abundance of targets can be variable and dependent on the values of $\mathbf{B}$, when there are interactions between target spectra and background spectra.

The underlying physical assumption of model (5.8) is that given an observed target pixel, the hyperspectral sensor will not only receive the reflectance of the target and the background independently (modelled by a linear combination of $\mathbf{T}\gamma$ and $\mathbf{B}\beta$), it will also receive the multiple scattering of the target and the background (modelled by additional interaction effects $\mathbf{H}\eta$ between the target and the background).

Similarly to the explanation of the model used for unmixing of HSIs [75], for example, we assume that there are only two components "trees" and "vehicle" presented in an observed target pixel, where the 'vehicle" is the target to be detected and "trees" are backgrounds. Illustrations of complex photons paths possible to occur are shown in Figure. 5.1.



(a) (b)

Figure 5.1: Examples of complex photon paths possible to occur: (a) LMM; (b) interaction effects.

In the assumption of LMM, the hyperspectral sensor will receive signals backscattered by the trees and the vehicle independently, which are represented by the terms $\beta\mathbf{b}$ and $\gamma\mathbf{t}$, respectively as illustrated in Figure. 5.1(a). However, if a signal is first backscattered by the vehicle to trees (or vice versa), and then backscattered to the sensor, this will result in multiple scattering and the hyperspectral sensor will receive interaction effects between endmembers "trees" and "vehicle", which we assume to be represented by the interaction term $\eta(\mathbf{t}\odot\mathbf{b})$. This multiple scattering process is illustrated in Figure. 5.1(b). It is possible that higher order interactions are also received by the hyperspectral sensor. However, as with the analysis of unmixing of HSI [75, 76, 77, 78], these higher order terms can be neglected.

5.4 Experimental studies

We conduct comparative experiments on two publicly available hyperspectral datasets. One is for synthetic target detection analysis and the other is for real target detection analysis:

1) *Synthetic targets*: the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) dataset was captured at the Lunar Crater Volcanic Field (LCVF) in northern Nye County, Nevada, USA (http://aviris.jpl.nasa.gov/data/). It has a total of 224 spectral bands covering the spectral range of 400nm-2500nm.

The dataset has been widely used for simulated HSI target detection such as in [80, 2]. We use a 200 × 200 sub-image in our experiment. There is no defined target in the scene. We manually implant target pixels into the image and simulate the target detection process, to explore the capability of the proposed method.

2) *Real targets*: the Hymap dataset contains ground-truth spectra of targets and has the targets readily deployed in the scene. It was captured at the location of a small town of Cook City, USA. This image is published by Rochester Institute of Technology (RIT), Rochester, NY, USA [3]. The dataset comes with the locations and pure spectra for all the desired targets. It has a total of 126 spectral bands and is of size 280 × 800, covering the spectral range of 453nm-2496nm. The Hymap dataset serves as standard target detection dataset and is widely used, such as in [2, 72, 80, 81, 82].

5.4.1 Synthetic targets: the AVIRIS dataset

In the AVIRIS image, five target pixels are manually implanted using two mixing models that simulate the possible linear/multi-scattering behaviour of hyperspectral sensors. This experiment focuses on exploring the capability of the proposed method in capturing the interaction effects between the target spectrum and the background spectra.

The AVIRIS image is shown in Figure. 5.2(a). The locations of the five implanted pixels are depicted in Figure. 5.2(b). The implanted target is a species of mineral called almandine, which is not from the AVIRIS dataset. As with [2], the spectrum of the target almandine is rescaled and resampled to match the AVIRIS image wavelength. The target spectrum and five background spectra originally at implanted locations are show in Figure. 5.3. In this simulation, we only conduct comparative experiments on MSD and MSDinter, to explore the potential of MSDinter.

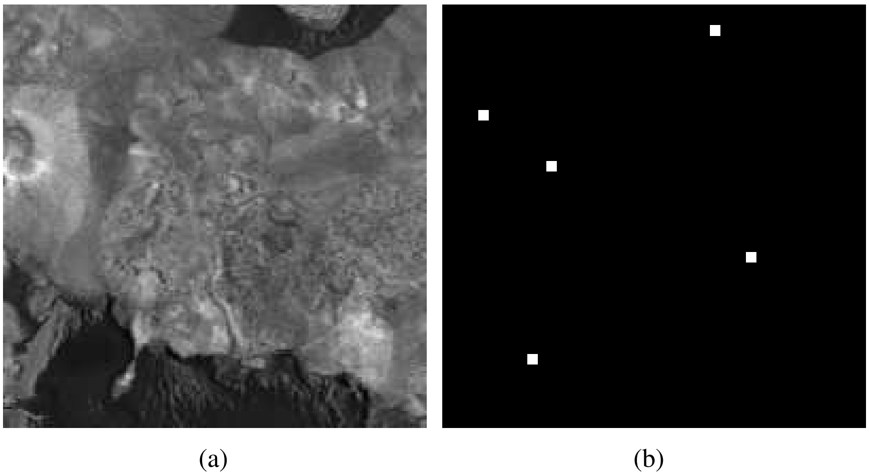


Figure 5.2: (a) The AVIRIS sub-image (200 × 200) of the third spectral band. (b) Locations of the implanted targets.

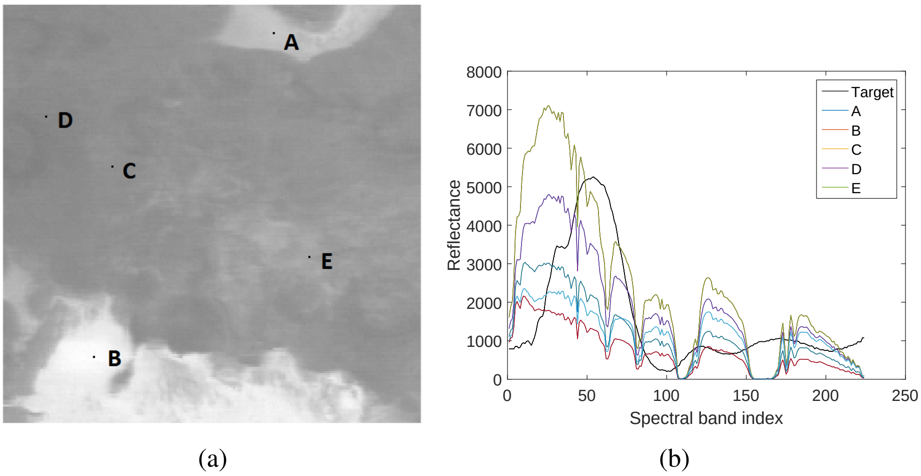


Figure 5.3: (a) The locations of the representative background spectral samples. (b) The pure target spectrum and the representative background spectra located in (a).

5.4.1.1 Experimental settings

The implanted target pixel $\mathbf{x}$ is mixed with the prior target spectrum $\mathbf{t}$ and the original background spectrum $\mathbf{b}$ at each implanted location shown in Figure. 5.2(b). Two mixing models are used:

- Linear mixing model (LMM):

$$\mathbf{x} = f_t\mathbf{t} + f_b\mathbf{b}, \tag{5.29}$$

- Bilinear mixing model (BMM):

$$\mathbf{x} = f_t\mathbf{t} + f_b\mathbf{b} + (1 - f_t - f_b)\mathbf{t} \odot \mathbf{b}, \tag{5.30}$$

where f_t and f_b are implanted fractions of the target spectrum and of the background spectrum, respectively. The fractions of all terms are sum to 1 in LMM (5.29) and BMM (5.30), respectively. We simulate four datasets for LMM and BMM, respectively, and details of the implanted fractions are shown in Table 5.1.

Table 5.1: Details of the implanted fractions for the AVIRIS dataset.

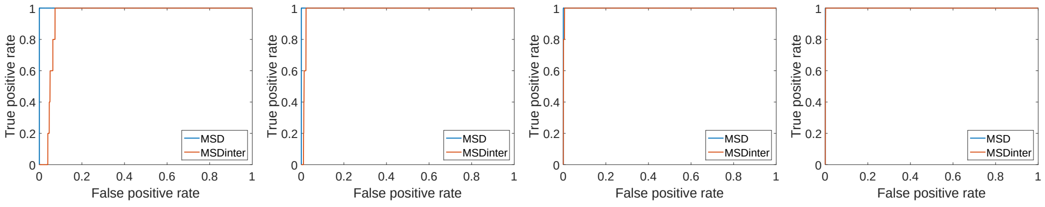
	LMM		BMM		
Fraction	f_t	f_b	f_t	f_b	$1 - f_t - f_b$
Simulation 1	5%	95%	1%	5%	94%
Simulation 2	7%	93%	1%	7%	92%
Simulation 3	9%	91%	1%	9%	90%
Simulation 4	10%	90%	1%	10%	89%

As the spectra of the mixed target pixels may appear very different from the spectra in the original image, the detection may become trivial and the performances of both detectors are not distinguishable. Therefore we randomly add white noise with mean $\mathbf{0}$ to the whole image after implanting the target pixels, which mimics the distortion caused by the sensors in real applications. In this experiment, the added white noise is measured in terms of the Signal-to-Noise Ratio (SNR). The SNR in decibels is defined as

$$\mathrm{SNR}_{dB} = 10\log_{10}\left(\frac{\sigma_i^2}{\sigma_{noise}^2}\right), \tag{5.31}$$

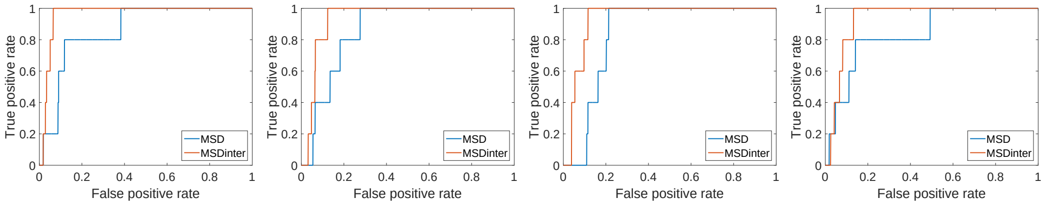
where σ_i is the standard deviation of the ith band image for $i = 1, \ldots, 224$ and σ_{noise} is the standard deviation of the noise added to each band image. We set SNR_{dB} = 20dB and therefore add white noise with $\sigma^2_{noise} = \sigma^2_i/100$ in each band image in the following simulations.

We use the single target spectrum and five background spectra shown in Figure. 5.3 as the target subspace $\mathbf{T}$ and the background subspace $\mathbf{B}$, respectively. The receiver operating characteristic (ROC) curve is adopted to measure the detection performances. The ROC is a threshold-free measurement. For each detector result, the threshold varies in a range to obtain a set of pairs of the true positive rate and the false positive rate, which is then used to plot the ROC curve. We also employ the area under curve (AUC) statistics to measure the detection performance quantitatively, in pair with the ROC curve.



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

Figure 5.4: ROC curves of detecting implanted target pixels mixed by LMM: (a) $f_t = 5\%$, $f_b = 95\%$; (b) $f_t = 7\%$, $f_b = 93\%$; (c) $f_t = 9\%$, $f_b = 91\%$; (d) $f_t = 10\%$, $f_b = 90\%$.



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

Figure 5.5: ROC curves of detecting implanted target pixels mixed by BMM: (a) $f_t = 1\%$, $f_b = 5\%$, $1 - f_t - f_b = 94\%$; (b) $f_t = 1\%$, $f_b = 7\%$, $1 - f_t - f_b = 92\%$; (c) $f_t = 1\%$, $f_b = 9\%$, $1 - f_t - f_b = 90\%$; (d) $f_t = 1\%$, $f_b = 10\%$, $1 - f_t - f_b = 89\%$.

The ROC curves of detecting the LMM-based implanted targets pixels and the

Table 5.2: AUC statistics of MSD and MSDinter for the AVIRIS dataset.

	LMM		BMM	
AUC	MSD	MSDinter	MSD	MSDinter
Simulation 1	1	0.945	0.860	0.961
Simulation 2	1	0.984	0.857	0.933
Simulation 3	1	0.998	0.839	0.931
Simulation 4	1	1	0.837	0.930

BMM-based implanted targets pixels by MSD and MSDinter are shown in Figure. 5.4 and Figure. 5.5, respectively. The AUC performances corresponding to Figure. 5.4 and Figure. 5.5 are listed in Table 5.2.

5.4.1.2 Results on LMM-mixed targets

From the results listed in Table 5.2 and shown in Figure. 5.4, where implanted target pixels are synthesised by LMM, we can observe at least two patterns. Firstly, MSD achieves perfect performance for LMM-mixed targets, i.e. AUC = 1 on detecting all implanted targets with enumerated fractions. That is, it implies that if target pixels captured by the HSI sensor are mixed by the linear combination of the target spectrum and the background spectrum, MSD can perform perfectly. Secondly, as the implanted target fraction f_t increases, e.g. slightly increasing from 5% to 10%, the detection performance of MSDinter improves from 0.945 to 1. It implies that MSDinter can also achieve nearly perfect to perfect performance even when targets are linearly mixed without any interaction effect.

5.4.1.3 Results on BMM-mixed targets

In this simulation, the implanted target fraction f_t is fixed to be 1%, and the implanted background fraction is ranged from 5% to 10%. The rest of fractions are occupied by the interaction terms $\mathbf{t} \odot \mathbf{b}$. The performances of MSD and MSDinter on detecting the BMM-based implanted targets are listed in Table 5.2 and shown in Figure. 5.5. We can observe that MSDinter outperforms MSD on detecting all BMM-based implanted targets with enumerated fractions. It reveals that if the interaction between the background spectrum and the target spectrum does exist, MSDinter can achieve better performance than that of MSD, as the latter fails to take the interaction effects into consideration.

5.4.1.4 Detection statistics of MSD and MSDinter

We further compare the test statistics of all pixels in the AIVRIS image processed by MSD and MSDinter. The test statistics of 40,000 pixels in the LMM-based simulation and BMM-based simulation are shown in Figure. 5.6 and Figure. 5.7, respectively. Due to the nature of MSD and MSDinter, the test statistics are always greater than 1 and the pixels with higher statistics are considered more likely to be targets.

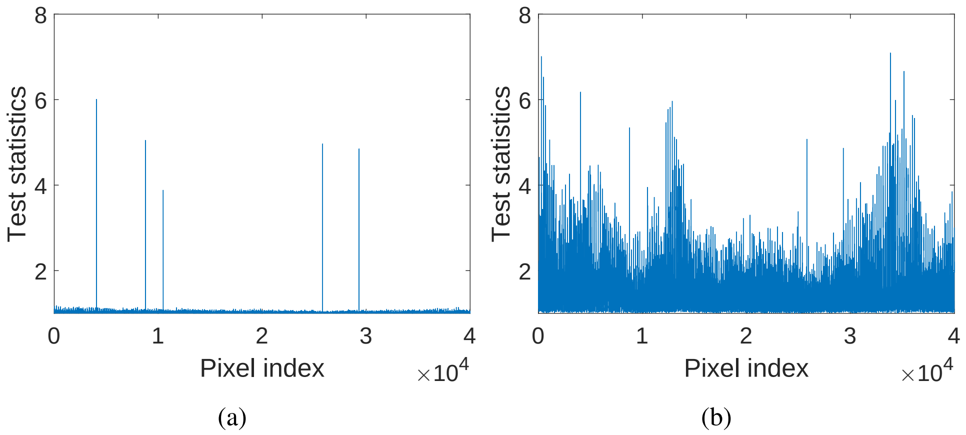


Figure 5.6: Test statistics of the AVIRIS image implanted by LMM with mixing fractions $f_t = 9\%$, $f_b = 91\%$: (a) MSD, AUC = 1; (b) MSDinter, AUC = 0.998.

In Figure. 5.6(a), we can observe that MSD has very distinguishable test statistics of the implanted targets which are linearly mixed without interaction. However in Figure. 5.7(a), the test statistics of MSD on targets not only largely decrease but also become undistinguishable when the implanted targets are bilinearly mixed with interaction, and the performance of MSD drops significantly, from AUC = 1 (5.6(a)) to AUC = 0.839 (5.7(a)). On the other hand, the test statistics of MSDinter are more stable than those of MSD, whether or not the implanted pixels are mixed by LMM or BMM, which are depicted in Figure. 5.6(b) and 5.7(b). It indicates that MSD-inter can handle both simple and complex mixing effects, with much more stable performance than MSD.

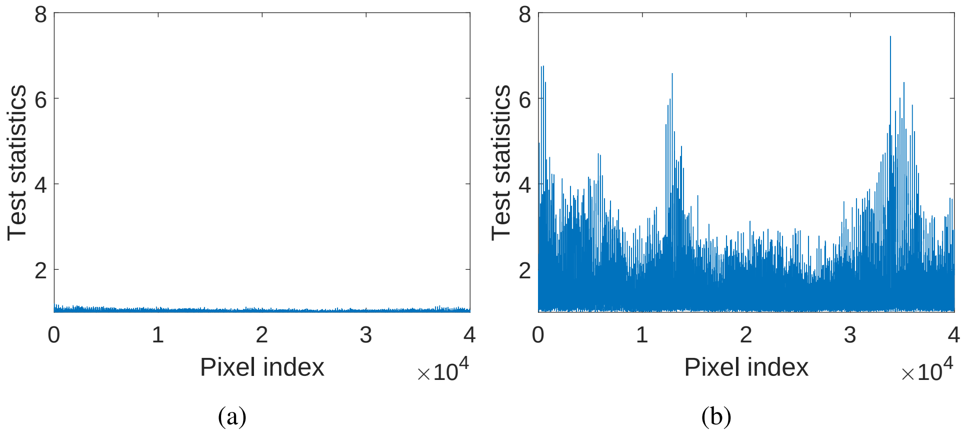


Figure 5.7: Test statistics of the AVIRIS image implanted by BMM with mixing fractions $f_t = 1\%$, $f_b = 9\%$ $1 - f_t - f_b = 90\%$: (a) MSD, AUC = 0.839; (b) MSDinter, AUC = 0.931.

5.4.2 Real targets: the Hymap dataset

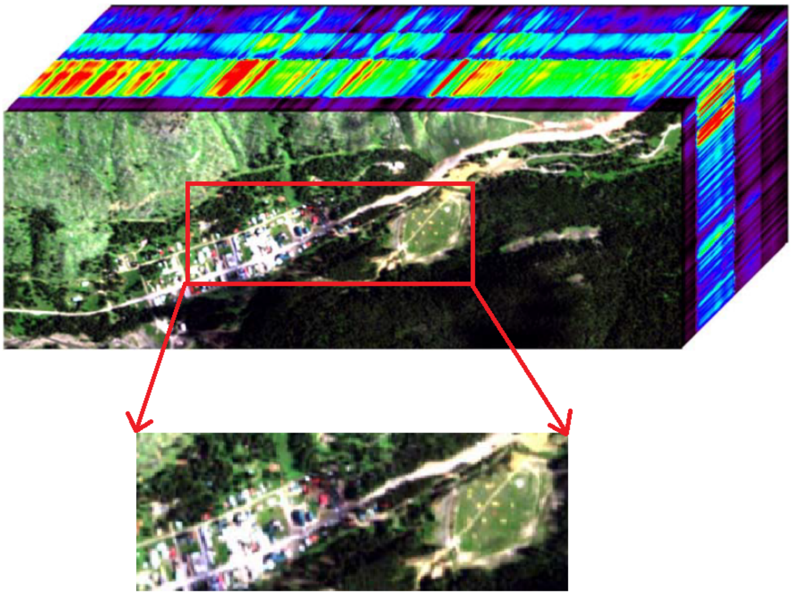


Figure 5.8: The Hymap image with a spatial size of 280×800 [3]. We cropped a spatial size of 100×300 sub-image for evaluation in this experiment.

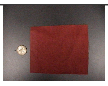
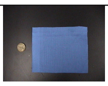
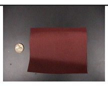
For the real hyperspectral dataset, i.e. the Hymap dataset where targets are deployed in the scene, the proposed MSDinter method is evaluated against not only MSD but some other well-known detectors, such as OSP (2.12) [18], CEM (2.13) [19] and ACE (2.14) [22]. We also compare the MSDinter method with an SR-based method termed STD (2.16) [23].

The Hymap image is shown in Figure. 5.8. As the desired targets are mainly located in the central part of the whole image and the materials lie around the margin of the image are homogeneous which are mainly composed of trees, we cropped a 100×300 sub-image from the central part of the original Hymap image for evaluating the performances of detectors. Such a sub-image setting has been widely used and well accepted by researchers, such as in [72, 83, 84]. Different experimental settings for analysing the Hymap image can also be found in [2, 66, 68, 80, 81, 82] for different illustrative purposes.

There are seven types of targets in the Hymap dataset, including four types of fabric panels (F1, F2, F3, F4) and three types of vehicles (V1, V2, V3). There are two samples with different sizes deployed in the scene for F3 and F4, termed F3a and F3b, F4a and F4b, respectively. The rest of targets, i.e. F1, F2, V1, V2 and V3, have only one sample each. When one type of target is to be detected, e.g. F3a and F3b, the other targets, i.e. F1, F2, F4a, F4b, V1, V2 and V3, are regarded as background pixels. The seven types of targets and their central coordinates of region of interests (ROIs) are shown in Table 5.3. Since the spatial resolution of the Hymap dataset is about 3m, we can infer that F1 (3m $\times$ 3m), F2 (3m $\times$ 3m) are nearly full pixels, whereas all the other targets are smaller than a pixel and appear as subpixels. Therefore a mixture model should be considered for all the targets, and the interaction effects between the target and the background are likely to occur. The cropped sub-image as well as ROIs of seven types of targets are shown in Figure 5.9(a) and Figure 5.9(b), respectively.

The spectrum of each desired target (F1-F4 and V1-V3) is provided by projected-equipped SPL files [3]. As with [2], we rescale and resample the SPL spectra according to the Hymap HSI wavelength. Preprocessed target spectra are given in Figure. 5.10. We randomly select one sample spectral signature of each target in the scene, and plot them in Figure. 5.11. Comparing Figure. 5.10 with Figure. 5.11, we can clearly see that target spectra signatures in the scene are very different from those ground-truth spectra in Figure. 5.10, and the pattern of how the

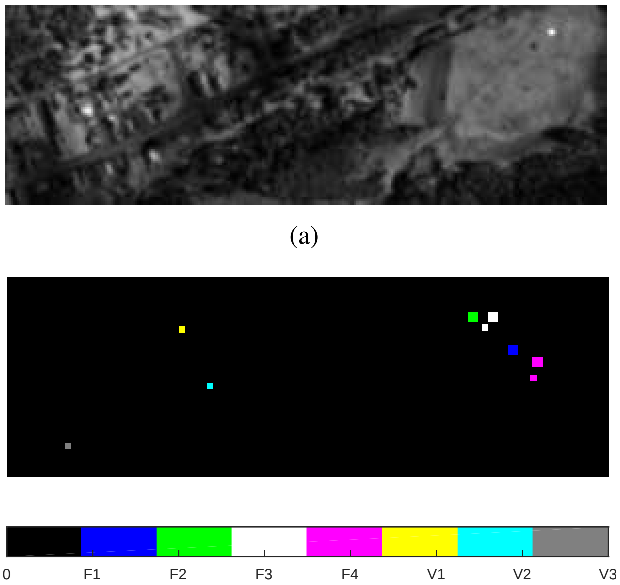
Table 5.3: List of the targets in the Hymap dataset

Target	Description and pixel size of ROI	Central coordinates of ROI	Photo
F1	Red cotton (3m × 3m) (5×5 pixels)	(138, 504)	
F2	Yellow nylon (3m × 3m) (5×5 pixels)	(122, 484)	
F3 a&b	Blue cotton (2m × 2m & 1m × 1m) (5×5 pixels &3×3 pixels)	(122, 494) & (127, 490)	
F4 a&b	Red Nylon (2m × 2m & 1m × 1m) (5×5 pixels &3×3 pixels)	(144, 516) & (152, 514)	
V1	Green Chevy Blazer (3×3 pixels)	(128, 339)	
V2	White Toyota T100 (3×3 pixels)	(156, 353)	
V3	Red Subaru GL (3×3 pixels)	(186, 282)	

sampled target spectra are mixed with the background spectra is complicated.

5.4.2.1 Experimental settings

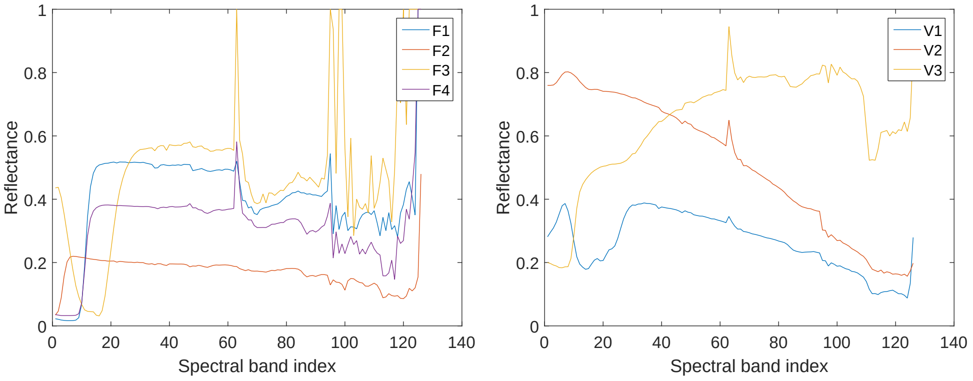
In realistic target detection problems, the background statistics are usually unknown. As explained in [1], the statistics of background can be estimated by all pixels within the area of interest when detectors are applied in a sparse target environment. In our experiment, there are 30,000 pixels in the cropped Hymap subimage and among which there is only 1 target pixel to be detected for each desired target. The number of target/image ratio is 1/30000, which means our detection environment is sufficiently sparse. Therefore we can use all pixels of the cropped Hymap image to estimate the mean μ_b and the covariance $\mathbf{C}_b$ of the background. In this way, the detector of each test pixel has global and identical background statistics (mean μ_b and covariance $\mathbf{C}_b$). In addition, detectors used in this chapter, including MSD, MSDinter, ACE, CEM, OSP, all adopt the same aforementioned background samples for fair comparison. For the SR-based method STD, the back-



(a)

(b)

Figure 5.9: (a) The Hymap sub-image (100 × 300) of the 33th spectral band; (b) ROIs of seven types of targets (F1, F2, F3, F4, V1, V2 and V3) in the Hymap sub-image. There are two samples of targets F3 and F4 each, termed F3a and F3b, and F4a and F4b, respectively. The pixel sizes of the ROI of targets F1, F2, F3a, F3b, F4a, F4b, V1, V2 and V3 are 25, 25, 25, 9, 25, 9, 9, 9 and 9, respectively. Different types of targets are shown in different colours.



(a) (b)

Figure 5.10: Rescaled prior spectra of all the targets in the SPL files: (a) fabric panels; (b) vehicles.

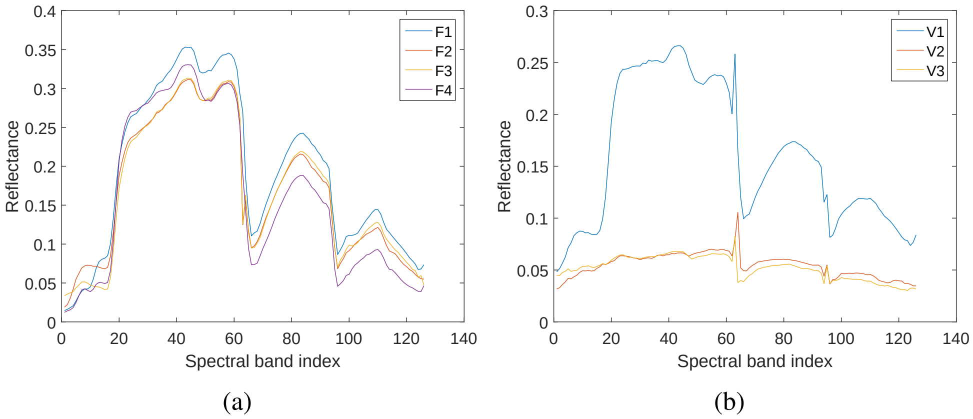


Figure 5.11: Rescaled sample spectra of all targets in the Hymap scene: (a) fabric panels; (b) vehicles. The selected sample spactra are located in the central coordinates of the ROIs of F1, F2, F3a, F4a, V1, V2 and V3, respectively, which are shown in Table 5.3.

ground dictionary for each test pixel is constructed by 29,999 pixels of the cropped image excluding the test pixel itself.

Among the compared detectors, MSD, MSDinter and OSP involve the construction of background subspace $\mathbf{B}$. We use the mean-centred HSI (removing the estimated mean μ_b from the HSI) to compute the covariance matrix $\mathbf{C}_b$ and then preserve significant eigenvectors of $\mathbf{C}_b$ to create columns of $\mathbf{B}$. For MSD and MSDinter, we should also construct target subspace $\mathbf{T}$. Since there is only one prior spectrum of each desired target $\mathbf{m}_t$, we actually do not need to do eigen-decomposition on $\mathbf{m}_t$ to obtained the target subspace $\mathbf{T}$. Instead, we subtract the background mean μ_b from the prior target spectrum $\mathbf{m}_t$, i.e. $\mathbf{m}_t - \mu_b$, and then normalise $\mathbf{m}_t - \mu_b$ to have a unit L_2-norm as the target subspace $\mathbf{T}$. As a result, the estimated background endmembers $\mathbf{b}$ and the target endmember $\mathbf{t}$ all have unit L_2-norm and are independent of each other. For STD, the union dictionary is constructed by the concatenation of 29,999 pixels and the single prior spectrum of each desired target for each test pixel. Again, each column of the dictionary is normalised to have unit L_2-norm. In this chapter, the STD method is solved by a greedy algorithm called orthogonal matching pursuit (OMP) [11].

We should note that each target deployed in the scene has an ROI [3], which means that the target may appear in any coordinates within the ROI. For example, F1 has a 5 × 5 pixels size of ROI and the central coordinates of ROI are (138, 504). It implies that if we detect at least one pixel as a target in the ROI, then this detection is regarded as a 100% correct detection. As with [2] and [84], we use the false alarm rate (FAR) to measure the detection performances of the compared methods. The FAR in this experiment is defined as the number of pixels not in the target ROI but have test statistic values equal to or greater than that of the pixel with the highest statistic value within the target ROI, normalised by the total number of pixels in the Hymap HSI (i.e. 30,000 pixels).

Among the methods to be compared, MSD, MSDinter, OSP and STD have parameters to tune. For MSD, MSDinter and OSP, the parameter is r_b, which is the number of eigenvectors to be preserved for the background subspace **B**. For STD, the parameter is the sparse level, termed L, which is the number of HSI pixels to be selected for the sparse representation. As ACE and CEM only use the target endmembers and the whole HSI to construct detectors, no tuning parameters are involved. Due to the limited number of target samples in the dataset, it is infeasible to tune parameters via cross validation. Hence as with most published works of HSI target detections conducted on the Hymap dataset such as [2, 81, 82], the parameter of each detector is manually tuned to show the optimal performance of the algorithms for illustrative purposes. The number of preserved eigenvectors r_b of the background subspace **B** for MSD, MSDinter and OSP and the sparse level L of representation for STD are listed in Table 5.4, respectively.

Table 5.4: The parameter r_b of OSP, MSD and MSDinter and the parameter L of STD.

Target	r_b			L
	OMP	MSD	MSDinter	STD
F1	9	110	5	10
F2	118	111	5	12
F3	58	11	5	12
F4	118	88	6	10
V1	91	91	6	10
V2	43	43	2	4
V3	105	106	10	12

5.4.2.2 Experimental results

The detection performances of all detectors are list in Table 5.5. We can observe that the proposed MSDinter outperforms MSD, ACE, CEM, OSP and STD in detecting all seven types of targets. Specifically, MSDinter can achieve the best detection performance on detecting F1, F2, F3 with FAR equal to 0. Compared with MSD, MSDinter significantly improves FARs for all targets. It implies that these observed target pixels captured by the HSI sensor are more likely to contain the interaction of background spectra and target spectra. In this sense, as MSDinter models the interaction effects, it achieves better performance than MSD, which fails to model the interaction effects.

Table 5.5: FAR under 100% detection of ACE, CEM, OSP, MSD, STD and MSDinter for the Hymap dataset. Boldface indicates the best performance.

FAR	ACE	CEM	OSP	MSD	STD	MSDinter
F1	1.02e-02	1.19e-02	0.01e-02	0.76e-02	0.06e-02	**0.00e-02**
F2	8.55e-02	1.11e-02	0.01e-02	0.14e-02	0.53e-02	**0.00e-02**
F3	0.57e-02	1.35e-02	0.27e-02	0.0057e-02	0.08e-02	**0.00e-02**
F4	0.21e-02	0.51e-02	0.08e-02	0.0037e-02	0.31e-02	**0.0027e-02**
V1	1.37e-02	1.41e-02	0.86e-02	0.62e-02	24.76e-02	**0.0013e-02**
V2	1.34e-02	2.22e-02	0.85e-02	0.40e-02	0.52e-02	**0.31e-02**
V3	19.94e-02	24.87e-02	1.82e-02	1.48e-02	11.36e-02	**0.54e-02**

For illustration purposes, we select one of the seven types of targets, i.e. F1, and plot prediction maps resulted from all compared methods. The prediction maps are shown in Figure. 5.12, in which the test statistic of each HSI pixel is colour coded. We can observe that the proposed MSDinter produces the most distinguishable detection results, as shown in Figure. 5.12(c). In the MSDinter prediction map (Figure. 5.12(c)), the test statistics of pixels within the ROI of F1 have the highest values compared with the statistics of all the other pixels, which result in the best detection performance with FAR equal to 0. On the other hand, the prediction maps of MSD, ACE, CEM, OSP and STD are not easy to distinguish F1 and the background, and their detection performances are not as good as that of MSDinter. In addition, comparing the prediction maps of MSD and MSDInter shown in Figure. 5.12(b) and Figure. 5.12(c), we can see that MSDinter eliminates the high statistics of background pixels and thus reduces FAR, which indicates that taking the target-background interaction effects into account can significantly improve the performance of the HSI target detection.

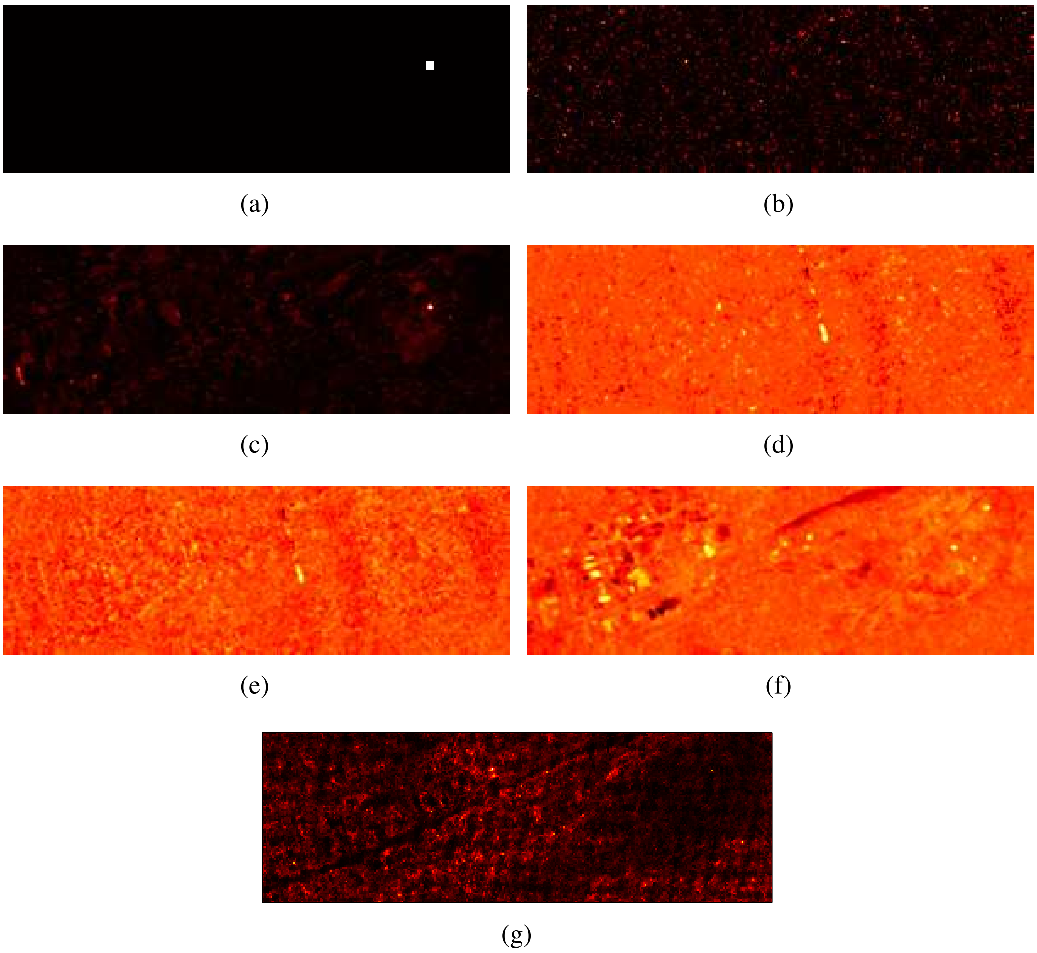


Figure 5.12: Test statistics for detecting F1 in the Hymap image. Brighter pixels have higher test statistics and therefore are more likely to be targets. (a) Ground-truth labels of F1; (b) MSD, FAR = 0.76e-02; (c) MSDinter, FAR = 0.00e-02; (d) ACE, FAR = 1.02e-02; (e) CEM, FAR = 1.19e-02; (f) OSP, FAR = 0.01e-02; (g) STD, FAR = 0.06e-02.

5.5 Conclusion

In this chapter we have proposed a new method called MSDinter for the hyperspectral target detection. The MSDinter method introduces interaction terms into the popular MSD to model and capture the interaction between target and background spectra. Compared with MSD, the proposed MSDinter method produces superior detection performance on the synthetic dataset of AVIRIS and the real dataset of Hymap, demonstrating the benefit of taking target-background interaction into modelling for target detection.

It is worthwhile to mention that, besides the platform of MSD, the proposed concept of *interaction effects* can also be applied to other target detection methods which have not yet considered target-background interaction. It is of our research interests to further work in this direction to investigate its potential of improving other established algorithms of target detection from hyperspectral images.

5.6 Appendix

This section describes in detail how to differentiate the conditional expected value of **x** with respect to **T**, i.e. $\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}}$, for model (5.1) and model (5.8), respectively.

To start with, assume that matrix **T** contains only one vector **t**. Then the model (5.1) of **x** is simplified as

$$\mathbf{x} = \mathbf{B}\beta + \mathbf{t}\gamma + \mathbf{n}, \tag{5.32}$$

where γ is a scalar. It follows that the derivative $\frac{\partial E[\mathbf{x}|\mathbf{t},\mathbf{B}]}{\partial \mathbf{t}}$ effectively measures the impact on the expected value of **x** from one-unit change of each element in **t**. According to the definition of the Jacobian matrix, the resultant derivative of $\frac{\partial E[\mathbf{x}|\mathbf{t},\mathbf{B}]}{\partial \mathbf{t}}$ will be a $p \times p$ matrix, given a $p \times 1$ vector **x** and a $p \times 1$ vector **t**. That is:

$$\frac{\partial E[\mathbf{x}|\mathbf{t},\mathbf{B}]}{\partial \mathbf{t}} = \begin{bmatrix} \gamma & 0 & \dots & 0 \\ 0 & \gamma & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \gamma \end{bmatrix}_{p \times p} = \gamma \mathbf{I}_p, \tag{5.33}$$

which turns out to be a diagonal $p \times p$ matrix $\gamma\mathbf{I}_p$, where $\mathbf{I}_p$ denotes the $p \times p$ identity matrix.

When matrix **T** contains multiple vectors $\mathbf{t}_i$ for $i = 1, \dots, r_t$, which is the case of model (5.1), the derivative of $\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}}$ measures the impact on the expected value of **x** from one-unit change of each element in **T**. Let us rewrite model (5.1) as

$$\mathbf{x} = \mathbf{B}\beta + \mathbf{T}\gamma + \mathbf{n} = \mathbf{B}\beta + [\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}]\gamma + \mathbf{n}, \tag{5.34}$$

where γ is an r_t-variate vector. Then the resultant derivative $\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}}$ will be a $(pr_t) \times p$ matrix, with **x** being a $p \times 1$ vector and **T** being a $p \times r_t$ matrix.

Based on the results in (5.33) and letting Γ_i denote the $p \times p$ diagonal matrix with γ_i on the diagonal, i.e.

$$\Gamma_i = \begin{bmatrix} \gamma_i & 0 & \ldots & 0 \\ 0 & \gamma_i & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \gamma_i \end{bmatrix}_{p \times p} = \gamma_i \mathbf{I}_p, \tag{5.35}$$

it follows that the derivative in the case of model (5.1) is

$$\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{r_t} \end{bmatrix}_{(pr_t) \times p}, \tag{5.36}$$

which is a concatenated matrix.

For model (5.8), the addition of interaction term $\mathbf{H}\eta$ introduces complexity to the computation, but due to the nature of linear algebra, the derivative can be found in a similar fashion. With the added interaction term, the model (5.8) of **x**,

$$\mathbf{x} = \mathbf{B}\beta + \mathbf{T}\gamma + \mathbf{H}\eta + \mathbf{n}, \tag{5.37}$$

has the derivative as

$$\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{r_t} \end{bmatrix}_{(pr_t)\times p} + \frac{\partial \mathbf{H}\eta}{\partial \mathbf{T}}. \tag{5.38}$$

For the derivation $\frac{\partial \mathbf{H}\eta}{\partial \mathbf{T}}$, we can follow the same steps by which we get results (5.33) and (5.36). Firstly, recall that the interaction matrix $\mathbf{H}$ has been expanded in (5.10):

$$\mathbf{H} = [\mathbf{t}_1 \odot \mathbf{b}_1, \ldots, \mathbf{t}_1 \odot \mathbf{b}_{r_b}, \mathbf{t}_2 \odot \mathbf{b}_1, \ldots, \mathbf{t}_2 \odot \mathbf{b}_{r_b}, \ldots, \mathbf{t}_{r_t} \odot \mathbf{b}_1, \ldots, \mathbf{t}_{r_t} \odot \mathbf{b}_{r_b}].$$

Thus $\frac{\partial \mathbf{H}\eta}{\partial \mathbf{t}_i}$, where $i = 1, \ldots, r_t$, can be written as

$$\begin{aligned} \frac{\partial \mathbf{H}\eta}{\partial \mathbf{t}_i} &= \begin{bmatrix} \sum_{j=1}^{r_b} b_{j,1}\eta_{i,j} & 0 & \cdots & 0 \\ 0 & \sum_{j=1}^{r_b} b_{j,2}\eta_{i,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{j=1}^{r_b} b_{j,p}\eta_{i,j} \end{bmatrix}_{p\times p} \\ &= \begin{bmatrix} \sum_{j=1}^{r_b} \mathbf{B}_{1,j}\eta_{i,j} & 0 & \cdots & 0 \\ 0 & \sum_{j=1}^{r_b} \mathbf{B}_{2,j}\eta_{i,j} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum_{j=1}^{r_b} \mathbf{B}_{p,j}\eta_{i,j} \end{bmatrix}_{p\times p} \\ &= \begin{bmatrix} \mathbf{B}_{1,\cdot}^T\eta_i & 0 & \cdots & 0 \\ 0 & \mathbf{B}_{2,\cdot}^T\eta_i & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{B}_{p,\cdot}^T\eta_i \end{bmatrix}_{p\times p}, \end{aligned} \tag{5.39}$$

which is a diagonal $p \times p$ matrix, where η_i is a segment of η with

$$\eta = [\eta_{1,1}, \ldots, \eta_{i,j}, \ldots, \eta_{r_t,r_b}]^T = [\eta_1^T, \ldots, \eta_i^T, \ldots, \eta_{r_t}^T]^T, \tag{5.40}$$

and $\mathbf{B}_{l,\cdot}$ denotes a column vector representing the lth row of matrix $\mathbf{B}$.

Let Π_i denote the resultant derivative with respect to $\mathbf{t}_i$ in (5.39):

$$\Pi_i = \begin{bmatrix} \mathbf{B}_{1,\cdot}^T\eta_i & 0 & \dots & 0 \\ 0 & \mathbf{B}_{2,\cdot}^T\eta_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{B}_{p,\cdot}^T\eta_i \end{bmatrix}_{p\times p} . \tag{5.41}$$

The derivative of $\frac{\partial \mathbf{H}\eta}{\partial \mathbf{T}}$ is then the concatenation of Π_i:

$$\frac{\partial \mathbf{H}\eta}{\partial \mathbf{T}} = \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_{r_t} \end{bmatrix}_{(pr_t)\times p} . \tag{5.42}$$

By substituting (5.42) back to (5.38), the derivative of the expected value of $\mathbf{x}$ given the interaction model (5.8) is then

$$\frac{\partial E[\mathbf{x}|\mathbf{T},\mathbf{B}]}{\partial \mathbf{T}} = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{r_t} \end{bmatrix}_{(pr_t)\times p} + \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \vdots \\ \Pi_{r_t} \end{bmatrix}_{(pr_t)\times p} . \tag{5.43}$$

Chapter 6

HSI Target Detection: Matched Shrunken Subspace Detectors (MSSD)

6.1 Introduction

Target detection is an important task of hyperspectral image (HSI) analysis [14, 71, 15]. To target detection, the matched subspace detector (MSD) [7] is one of the most widely-used subspace-based approaches, underlying which is the idea of the linear mixing model (LMM) [5] shown in (2.9).

To achieve an HSI target detection, the MSD determines whether a test pixel can be represented by a linear combination of target spectral signatures and background spectral signatures. To this end, two subspaces are constructed: the target subspace and the background subspace. In each subspace, the MSD assumes that each basis vector represents an endmember, which is in line with the assumption of the LMM for HSI analysis.

To construct the two subspaces, the MSD usually acquires their basis vectors from the eigen-decomposition of covariance matrices of the training samples [14, 85]. The eigenvectors with dominant eigenvalues, termed leading eigenvectors, are selected as bases to span the subspaces, while those with small eigenvalues are discarded. This is essentially a scheme of basis selection, or say 0/1 weighting, which extracts a subspace out of the full eigenspace.

In fact, the 0/1 weighting scheme of the MSD implicitly imposes a sparseness constraint or say an l_0-norm regularisation while building its LMM. However, it is well known that such a "hard" selection may exhibit high variance on the selected leading eigenvectors. Alternatively, explicit sparse representation (SR)-based techniques have also been developed in hyperspectral target detection [23, 24, 86], with selection of a small number of atoms from a large dictionary. That is, these SR methods model a test HSI pixel as a linear combination of only few atoms from an over-complete dictionary; atoms in the dictionary are usually also samples, hence these SR methods can be viewed as being developed in the original sample space. Regarding the construction of the dictionary, [23] propose to construct a background spectra dictionary and a target spectra dictionary separately; on the other hand, [24, 86] propose to construct an over-complete dictionary including both background spectra and target spectra.

To avoid the problem of high variance from such a "hard" selection, shrinkage methods [9] have been developed in statistical learning, mainly due to such a problem in regression analysis. Among the shrinkage methods, the most popular one is called ridge regression, also known as Tikhonov regularisation [87] in other disciplines; it shrinks the regression coefficients through imposing an l_2-norm constraint. In this way, the estimates of the coefficients become more stable and therefore can improve the performance of regression.

The l_2-norm regularisation has been investigated for analysing hyperspectral imagery [61, 72, 83, 88, 89, 90]. For the HSI classification, [61] and [88] assume that a test pixel can be collaboratively represented by raw spectral signatures. It is shown that l_2-norm constraints can actually improve the classification, instead of the "competitive" nature imposed by sparseness constraints (as l_1-norm or l_0-norm regularisation). For the HSI target detection, [72, 83, 89, 90] add a scaled identity matrix to the background clutter covariance matrix before inverting it, in order to avoid an ill-conditioned problem. It is worth noting that these l_2-norm regularisation methods are developed in the original sample space, rather than in the eigenspace as this work.

In this chapter, focusing on the popular MSD, we propose a new approach, called the matched shrunken subspace detector (MSSD), to target detection from hyperspectral images. Our MSSD is developed by shrinking the abundance vectors of the target and background subspaces in the hypothesis models of the MSD. The shrinkage is simply achieved by introducing l_2-norm regularisation into the MSD. We develop two types of the MSSD, one with isotropic shrinkage (and thus termed MSSD-i) and the other with anisotropic shrinkage (and termed MSSD-a). For these two new methods, we provide both the frequentist and Bayesian derivations. Experiments on a real hyperspectral imaging dataset called Hymap demonstrate that the proposed MSSD-i and MSSD-a can outperform the original MSD for hyperspectral target detection.

The main contributions of this chapter are two-fold. 1) Through introducing the l_2-norm regularisation terms into the MSD, we shrink the abundance vectors so that the variance in each basis direction of the subspaces is also reduced, leading to a more stable estimation. 2) We derive the proposed MSSD-i and MSSD-a from both the frequentist and Bayesian perspectives, with the latter showing how the proposed methods preserve Gaussian prior distributions of the abundance vectors, instead of the uniform prior distribution which is implicitly imposed by the original MSD.

The rest of this chapter is organised as follows. Section 6.2 reviews the original MSD. In section 6.3.1 and section 6.3.2, detailed formulation of the two proposed method, MSSD-i and MSSD-a, are introduced. Then the two proposed methods are derived from the Bayesian perspective and shown in section 6.4. The links of MSD, MSSD-i and MSSD-a are discussed in section 6.5. Section 6.6 presents the experimental results, with the whole work concluded in section 6.7.

6.2 Matched subspace detector (MSD)

6.2.1 Overview of the binary hypothesis testing model

From a statistical perspective, target detection is typically derived from a binary hypothesis testing problem [15]. It is based on the likelihood ratio of the conditional probability density functions (pdfs) of two competing hypotheses, given that the spectral signature of an HSI pixel $\mathbf{x} \in \mathbb{R}^p$ is treated a continuous random vector:

$$\begin{aligned} &H_0 : \mathbf{x} \text{ is a background pixel,} \\ &H_1 : \mathbf{x} \text{ is a target pixel,} \\ &\Rightarrow D(\mathbf{x}) = \frac{f_{\mathbf{x}|H_1}(\mathbf{x})}{f_{\mathbf{x}|H_0}(\mathbf{x})} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \end{aligned} \tag{6.1}$$

where $f_{\mathbf{x}|H_0}(\mathbf{x})$ and $f_{\mathbf{x}|H_1}(\mathbf{x})$ are two conditional pdfs of $\mathbf{x}$ under the null hypothesis H_0 and the alternative hypothesis H_1, respectively; ν is the detection threshold; and $D(\mathbf{x})$ is an output detector. In reality, the conditional pdfs are usually not available and expressed parametrically. Hence, the generalised likelihood ratio test (GLRT) [91] is commonly used to replace the unknown parameters by their maximum likelihood estimates (MLEs):

$$\begin{aligned} D_{GLRT}(\mathbf{x}) &= \frac{f_{\mathbf{x}|H_1}(\mathbf{x}; \hat{\omega}_1)}{f_{\mathbf{x}|H_0}(\mathbf{x}; \hat{\omega}_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \nu \\ &= \frac{\max_{\omega_1}\{f_{\mathbf{x}|H_1}(\mathbf{x}; \omega_1)\}}{\max_{\omega_0}\{f_{\mathbf{x}|H_0}(\mathbf{x}; \omega_0)\}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \end{aligned} \tag{6.2}$$

where ω_0 and ω_1 are unknown parameters of pdf $f_{\mathbf{x}|H_0}(\mathbf{x}; \omega_0)$ and pdf $f_{\mathbf{x}|H_1}(\mathbf{x}; \omega_1)$, respectively; and $\hat{\omega}_0$ and $\hat{\omega}_1$ are their MLEs. In this chapter, "^" denotes the estimates of unknown parameters.

6.2.2 Formulation of the matched subspace detector (MSD)

Following the idea of LMM (2.9) [5], the MSD models a test pixel by a linear combination of target spectral endmembers and background spectral endmembers, and these endmembers are represented by the basis vectors of the target subspace and the background subspace, respectively.

That is, derived from the binary hypothesis model (6.1), the MSD model [7] is formulated in (2.10) as follows:

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \ \mathbf{x} \text{ is a background pixel,} \\ &H_1 : \mathbf{x} = \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{n}_1, \ \mathbf{x} \text{ is a target pixel,} \end{aligned}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}]$ is a $p \times r_t$ matrix representing the target subspace, and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$ is a $p \times r_b$ matrix representing the background subspace; $\mathbf{T}$ is derived from a training target matrix $\mathbf{M}_T \in \mathbb{R}^{p \times N_t}$ whose columns are the N_t target spectra, and $\mathbf{B}$ is derived from a training background matrix $\mathbf{M}_B \in \mathbb{R}^{p \times N_b}$ whose columns are the N_b background spectra; γ and β are the corresponding abundance vectors of the subspaces $\mathbf{T}$ and $\mathbf{B}$, respectively; and $\mathbf{n}_0$ and $\mathbf{n}_1$ are p-dimensional vectors of Gaussian white noise: $\mathbf{n}_0 \sim \mathscr{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$ and $\mathbf{n}_1 \sim \mathscr{N}(\mathbf{0}, \sigma_1^2\mathbf{I})$, respectively.

In general, a set of orthogonal basis vectors that spans the corresponding subspace are used as the column vectors of $\mathbf{T}$ or $\mathbf{B}$. In common practice, the leading eigenvectors of the target covariance matrix $\mathbf{C}_T$ and those of the background covariance matrix $\mathbf{C}_B$ are used as the columns of $\mathbf{T}$ and $\mathbf{B}$, respectively, as with [85][14]. In other words, when the test pixel $\mathbf{x}$ is a target pixel, it is decomposed into two components by linear combinations of the bases of $\mathbf{B}$ and $\mathbf{T}$, denoted by model H_1. When $\mathbf{x}$ is a background pixel, it is adequately described by model H_0, which is a reduced order model.

Let $\mathbf{V}$ be the concatenated matrix of $\mathbf{T}$ and $\mathbf{B}$, i.e. $\mathbf{V} = [\mathbf{T}\,\mathbf{B}] = [\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}, \mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$, then the abundance vectors γ and β of model H_1 can be concatenated into a single vector, denoted as α, i.e. $\alpha = \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = [\gamma_1, \ldots, \gamma_{r_t}, \beta_1, \ldots, \beta_{r_b}]^T$. Hence model H_1 can be written as

$$\begin{aligned} H_1 : \mathbf{x} &= \mathbf{T}\gamma + \mathbf{B}\beta + \mathbf{n}_1 \\ &= \begin{bmatrix} \mathbf{T} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} + \mathbf{n}_1 \\ &= \mathbf{V}\alpha + \mathbf{n}_1, \end{aligned} \tag{6.3}$$

and thus the MSD model (2.10) becomes

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n}_0, \ \mathbf{x} \text{ is a background pixel,} \\ &H_1 : \mathbf{x} = \mathbf{V}\alpha + \mathbf{n}_1, \ \mathbf{x} \text{ is a target pixel,} \end{aligned} \tag{6.4}$$

where now the unknown parameters are β, α, and those of $\mathbf{n}_0$ and $\mathbf{n}_1$.

The corresponding estimate of the likelihood ratio is the generalised likelihood ratio (GLR) of the MSD, formulated as

$$\begin{aligned}\hat{l}(\mathbf{x}) &= \frac{l(\hat{\alpha}, \hat{\sigma}_1^2; \mathbf{x})}{l(\hat{\beta}, \hat{\sigma}_0^2; \mathbf{x})} \\ &= \left(\frac{\hat{\sigma}_1^2}{\hat{\sigma}_0^2}\right)^{-p/2} \exp\left\{-\frac{1}{2\hat{\sigma}_1^2}\|\hat{\mathbf{n}}_1\|_2^2 + \frac{1}{2\hat{\sigma}_0^2}\|\hat{\mathbf{n}}_0\|_2^2\right\}.\end{aligned} \tag{6.5}$$

The MLEs $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$ are equal to $\frac{1}{p}\|\hat{\mathbf{n}}_0\|_2^2$ and $\frac{1}{p}\|\hat{\mathbf{n}}_1\|_2^2$, respectively. Taking the $2/p$ power of (6.5), we have the following GLR of the MSD:

$$\begin{aligned}L_{MSD}(\mathbf{x}) &= (\hat{l}(\mathbf{x}))^{2/p} \\ &= \left(\frac{\hat{\sigma}_1^2}{\hat{\sigma}_0^2}\right)^{-1} = \frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2} \\ &= \frac{\|\hat{\mathbf{n}}_0\|_2^2}{\|\hat{\mathbf{n}}_1\|_2^2} = \frac{\left\|\mathbf{x} - \mathbf{B}\hat{\beta}\right\|_2^2}{\|\mathbf{x} - \mathbf{V}\hat{\alpha}\|_2^2}.\end{aligned} \tag{6.6}$$

The MLEs of β and α in (6.6) are given by

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}}\left\{f_{\mathbf{x}|H_0}(\mathbf{x}; \beta, \sigma_0^2)\right\} = \underset{\beta}{\operatorname{argmin}}\left\{\frac{1}{2\sigma_0^2}\|\mathbf{x} - \mathbf{B}\beta\|_2^2\right\} \tag{6.7}$$

and

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}}\left\{f_{\mathbf{x}|H_1}(\mathbf{x}; \alpha, \sigma_1^2)\right\} = \underset{\alpha}{\operatorname{argmin}}\left\{\frac{1}{2\sigma_1^2}\|\mathbf{x} - \mathbf{V}\alpha\|_2^2\right\}, \tag{6.8}$$

and thus

$$\hat{\beta} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{x} = \mathbf{B}^T\mathbf{x} \tag{6.9}$$

and

$$\hat{\alpha} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{x}. \tag{6.10}$$

It is to be noted that the bases $[\mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$ of $\mathbf{B}$ are orthogonal, therefore $(\mathbf{B}^T\mathbf{B})^{-1}$ is an identity matrix and $\hat{\beta}$ can be simplified to $\mathbf{B}^T\mathbf{x}$, but the bases $[\mathbf{t}_1, \ldots, \mathbf{t}_{r_t}, \mathbf{b}_1, \ldots, \mathbf{b}_{r_b}]$ of $\mathbf{V}$ are not orthogonal to each other.

Based on (6.9) and (6.10), the residual sums of squares (RSS) e_0 and e_1 given model H_0 and model H_1 are computed as

$$H_0 : e_0 = ||\hat{\mathbf{n}}_0||_2^2 = \left\|\mathbf{x} - \mathbf{B}\hat{\beta}\right\|_2^2 = \mathbf{x}^T(\mathbf{I} - \mathbf{B}\mathbf{B}^T)\mathbf{x}, \tag{6.11}$$

and

$$H_1 : e_1 = ||\hat{\mathbf{n}}_0||_2^2 = ||\mathbf{x} - \mathbf{V}\hat{\alpha}||_2^2 = \mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T)\mathbf{x}, \tag{6.12}$$

where $\mathbf{I}$ is a $p \times p$ identity matrix. The final GLRT detector of the MSD model is then given by

$$D_{MSD}(\mathbf{x}) = \frac{e_0}{e_1} = \frac{\mathbf{x}^T(\mathbf{I} - \mathbf{B}\mathbf{B}^T)\mathbf{x}}{\mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T)\mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu. \tag{6.13}$$

Equation (6.13) is the same as equation (2.11) shown in Chapter 2. The value of D_{MSD} is compared to a threshold ν to make the final decision of which hypothesis should be rejected for the test pixel $\mathbf{x}$. Two tuning parameters should be determined for the MSD, which are the numbers of leading eigenvectors to be preserved in the subspace $\mathbf{B}$ and $\mathbf{T}$, i.e. r_b and r_t, respectively.

6.3 Matched shrunken subspace detector (MSSD)

In the MSD, the eigenvectors spanning the eigenspace are either preserved or discarded to build the subspaces. Rather than applying this selection scheme, it is desirable to adopt shrinkage schemes to reduce the variance induced by selection [9], in order to develop a more stable statistical method like the MSD, in particular for high-dimensional data like hyperspectral pixels. In the l_2-norm regularised shrinkage methods, all the available features/eigenvectors are preserved and their coefficients are shrunk. In other words, r_b and r_t are fixed to the maximal numbers of available features/eigenvectors. We propose to introduce l_2-norm regularisation into the MSD, to shrink the abundance vectors of the target and background subspaces in the hypothesis models of the MSD. We call this approach the matched shrunken subspace detector (MSSD).

It is worth noting that, in the hyperspectral target detection practice, we often have only one target spectrum as a priori information for training, and this single target spectrum usually comes from the spectrum library. If this is the case, the target training sample $\mathbf{M}_T$ is a single vector, not a matrix, and thus the typical eigen-decomposition cannot be applied on $\mathbf{M}_T$ to get $\mathbf{T}$. To this end and as usually the case, we use the normalised mean-corrected target spectrum as the only basis vector of the target subspace $\mathbf{T}$. As a result, we have $r_t = 1$ and $\mathbf{T} \in \mathbb{R}^{p\times 1}$, and the MSD does not discard this basis vector. Similarly, we do not shrink the abundance γ for the target subspace $\mathbf{T}$ when there is only one target spectrum available in practice, as also discussed in section 6.6.

In the following sections, we shall develop two types of the MSSD, MSSD-i with isotropic shrinkage and MSSD-a with anisotropic shrinkage, and provide both the frequentist and Bayesian derivations of them.

6.3.1 MSSD with isotropic shrinkage (MSSD-i)

While persevering all available eigenvectors, we introduce l_2-norm regularisation terms $\theta_0 \|\beta\|_2^2$ and $\theta_1 \|\alpha\|_2^2$ as constraints to the hypothesis models H_0 and H_1 of the MSD, respectively. The shrunken estimates of β and α now become

$$\hat{\beta}_{iso} = \underset{\beta}{\operatorname{argmin}}\{\|\mathbf{x} - \mathbf{B}\beta\|_2^2 + \theta_0 \|\beta\|_2^2\} \tag{6.14}$$

and

$$\hat{\alpha}_{iso} = \underset{\alpha}{\operatorname{argmin}}\{\|\mathbf{x} - \mathbf{V}\alpha\|_2^2 + \theta_1 \|\alpha\|_2^2\}, \tag{6.15}$$

where θ_0 and θ_1 are the parameters that control the degree of shrinkage imposed on the size of abundance vectors β and α, respectively. In this sense, the same shrinkage degree is applied to all eigenvectors, as done in (6.14) and (6.15), and we call this new method the MSSD with isotropic shrinkage, shortened as MSSD-i.

The test likelihood ratio of the MSSD-i is thus given by

$$L_{MSSD_{iso}}(\mathbf{x}) = \frac{\min_\beta\{\|\mathbf{x} - \mathbf{B}\beta\|_2^2 + \theta_0 \|\beta\|_2^2\}}{\min_\alpha\{\|\mathbf{x} - \mathbf{V}\alpha\|_2^2 + \theta_1 \|\alpha\|_2^2\}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \tag{6.16}$$

and the estimates of β and α in the MSSD-i are readily given as

$$\hat{\beta}_{iso} = ((1+\theta_0)\mathbf{I}_0)^{-1}\mathbf{B}^T\mathbf{x} \tag{6.17}$$

and

$$\hat{\alpha}_{iso} = (\mathbf{V}^T\mathbf{V} + \theta_1\mathbf{I}_1)^{-1}\mathbf{V}^T\mathbf{x}, \tag{6.18}$$

where $\mathbf{I}_0$ is a $r_b \times r_b$ identity matrix and $\mathbf{I}_1$ is $(r_t + r_b) \times (r_t + r_b)$ identity matrix. Hence the RSS e_0 and e_1 given models H_0 and H_1 are computed as

$$H_0 : e_0^{iso} = \left\|\mathbf{x} - \mathbf{B}\hat{\beta}_{iso}\right\|_2^2 = \mathbf{x}^T\left(\mathbf{I} - \mathbf{B}(1+\theta_0)^{-1}\mathbf{B}^T\right)\mathbf{x}, \tag{6.19}$$

and

$$H_1 : e_1^{iso} = \|\mathbf{x} - \mathbf{V}\hat{\alpha}_{iso}\|_2^2 = \mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V} + \theta_1\mathbf{I}_1)^{-1}\mathbf{V}^T)\mathbf{x}. \tag{6.20}$$

As with (6.13), the detector of the MSSD-i model is finally given by

$$D_{MSSD_{iso}}(\mathbf{x}) = \frac{e_0^{iso}}{e_1^{iso}} = \frac{\mathbf{x}^T(\mathbf{I} - \mathbf{B}\left((1+\theta_0)\mathbf{I}_0)^{-1}\mathbf{B}^T\right)\mathbf{x}}{\mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V} + \theta_1\mathbf{I}_1)^{-1}\mathbf{V}^T)\mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \tag{6.21}$$

To be noticed, the MSSD-i also has two tuning parameters, but not the r_b and r_t of the MSD: this time the tuning parameters are the shrinkage parameters θ_0 and θ_1.

6.3.2 MSSD with anisotropic shrinkage (MSSD-a)

Besides the directions represented by eigenvectors, the values of eigenvalues also reflect the information about distributions, in particular variances, of the data in the background and target subspaces. Therefore in addition to the MSSD-i, we propose another new method which preserves not just the useful information from all the available eigenvectors, but also the information of all the eigenvalues, while constructing the l_2-norm regularisation terms for the MSD.

Let $\Lambda_\mathbf{B}$ denote the background eigenvalue matrix with the eigenvalues of the background eigenvectors $\lambda_1^b, \ldots, \lambda_{r_b}^b$ on the diagonal, i.e. $\Lambda_\mathbf{B} = \text{diag}([\lambda_1^b, \ldots, \lambda_{r_b}^b]^T)$; and let $\Lambda_\mathbf{T}$ denote the target eigenvalue matrix with the eigenvalues of the target eigenvectors $\lambda_1^t, \ldots, \lambda_{r_t}^t$ on the diagonal, i.e. $\Lambda_\mathbf{T} = \mathrm{diag}([\lambda_1^t, \ldots, \lambda_{r_t}^t]^T)$.

It is known that small eigenvalues correspond to the eigenvectors having small variances, therefore we aim to shrink these directions the most. To this end, we can add the inverse of the eigenvalue matrix, $\Lambda_\mathbf{B}^{-1}$, to the regularisation term $\beta^T\beta$, for example. The shrunken estimates of β and α now become

$$\hat{\beta}_{aniso} = \underset{\beta}{\mathrm{argmin}} \left\{ (\mathbf{x} - \mathbf{B}\beta)^T(\mathbf{x} - \mathbf{B}\beta) + \theta_0 \beta^T \Lambda_\mathbf{B}^{-1} \beta \right\} \tag{6.22}$$

and

$$\hat{\alpha}_{aniso} = \underset{\alpha}{\mathrm{argmin}} \left\{ (\mathbf{x} - \mathbf{V}\alpha)^T(\mathbf{x} - \mathbf{V}\alpha) + \theta_1 \alpha^T \Lambda_\mathbf{V}^{-1} \alpha \right\}, \tag{6.23}$$

where θ_0 and θ_1 are again the parameters for the shrinkage degrees, and $\Lambda_\mathbf{V}$ is a concatenated matrix formed as:

$$\Lambda_\mathbf{V} = \begin{bmatrix} \Lambda_\mathbf{T} & \mathbf{0} \\ \mathbf{0} & \Lambda_\mathbf{B} \end{bmatrix}. \tag{6.24}$$

Compared with (6.14) and (6.15) which shrink isotropically over features in MSSD-i, both (6.22) and (6.23) shrink anisotropically over features. Hence we call this new method the MSSD with anisotropic shrinkage, shortened as MSSD-a.

As with (6.16), the test likelihood ratio of the MSSD-a is given by

$$L_{MSSD_{aniso}}(\mathbf{x}) = \frac{\min_\beta\{\|\mathbf{x} - \mathbf{B}\beta\|_2^2 + \theta_0 \beta^T \Lambda_\mathbf{B}^{-1} \beta\}}{\min_\alpha\{\|\mathbf{x} - \mathbf{V}\alpha\|_2^2 + \theta_1 \alpha^T \Lambda_\mathbf{V}^{-1} \alpha\}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \tag{6.25}$$

and the estimates of β_{aniso} and α_{aniso} are

$$\hat{\beta}_{aniso} = (\mathbf{I}_0 + \theta_0 \Lambda_\mathbf{B}^{-1})^{-1} \mathbf{B}^T \mathbf{x} \tag{6.26}$$

and

$$\hat{\alpha}_{aniso} = (\mathbf{V}^T\mathbf{V} + \theta_1 \Lambda_\mathbf{V}^{-1})^{-1} \mathbf{V}^T \mathbf{x}. \tag{6.27}$$

The RSS e_0^{aniso} and e_1^{aniso} given models H_0 and H_1 are then computed as

$$\begin{aligned} H_0 : e_0^{aniso} &= \left\| \mathbf{x} - \mathbf{B}\hat{\beta}_{aniso} \right\|_2^2 \\ &= \mathbf{x}^T(\mathbf{I} - \mathbf{B}(\mathbf{I}_0 + \theta_0 \Lambda_{\mathbf{B}}^{-1})^{-1}\mathbf{B}^T)\mathbf{x} \end{aligned} \tag{6.28}$$

and

$$\begin{aligned} H_1 : e_1^{aniso} &= \|\mathbf{x} - \mathbf{V}\hat{\alpha}_{aniso}\|_2^2 \\ &= \mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V} + \theta_1 \Lambda_{\mathbf{V}}^{-1})^{-1}\mathbf{V}^T)\mathbf{x}. \end{aligned} \tag{6.29}$$

As with (6.13) and (6.21), the detector of the MSSD-a model can be written as

$$\begin{aligned} D_{MSSD_{aniso}}(\mathbf{x}) &= \frac{e_0^{aniso}}{e_1^{aniso}} \\ &= \frac{\mathbf{x}^T(\mathbf{I} - \mathbf{B}(\mathbf{I}_0 + \theta_0 \Lambda_{\mathbf{B}}^{-1})^{-1}\mathbf{B}^T)\mathbf{x}}{\mathbf{x}^T(\mathbf{I} - \mathbf{V}(\mathbf{V}^T\mathbf{V} + \theta_1 \Lambda_{\mathbf{V}}^{-1})^{-1}\mathbf{V}^T)\mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu, \end{aligned} \tag{6.30}$$

Similar to MSSD-i, only two tuning parameters are need to be determined in the proposed MSSD-a: the shrinkage parameters θ_0 and θ_1.

6.4 Bayesian derivations of MSSD-i and MSSD-a

From the Bayesian perspective, the estimation of parameters β and α in the MSSD-i and the MSSD-a can be translated as the maximisation of a posteriori probability (MAP). Taking β for example, Bayes' theorem [9] says

$$f(\beta|\mathbf{x}) = \frac{f(\mathbf{x}|\beta)f(\beta)}{f(\mathbf{x})}, \tag{6.31}$$

where $f(\mathbf{x}|\beta)$ is a likelihood function of $\mathbf{x}$ and $f(\beta)$ is a prior distribution of β. Therefore the MAP estimate of β is

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} f(\beta|\mathbf{x}) \propto \underset{\beta}{\operatorname{argmax}} f(\mathbf{x}|\beta)f(\beta). \tag{6.32}$$

As the noise term $\mathbf{n}_0$ is assumed to be a multivariate Gaussian distribution $\mathbf{n}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$ in the LMM [5] and the MSD [7], the likelihood function $f(\mathbf{x}|\beta)$

can be formulated as

$$f(\mathbf{x}|\beta) \propto \exp\left\{ -\frac{1}{2\sigma_0^2} ||\mathbf{x} - \mathbf{B}\beta||_2^2 \right\}. \tag{6.33}$$

In the conventional MSD, an improper uniform (non-informative) prior distribution is actually assumed for parameter β of the selected leading eigenvectors. In the proposed MSSD-i and MSSD-a, adding l_2-norm regularisation in fact imposes Gaussian prior distributions on β.

6.4.1 Prior distributions of β and α in MSSD-i

For the MSSD-i, the prior distribution of β is in fact assumed to be

$$\beta \sim \mathscr{N}(\mathbf{0}, \sigma_B^2 \mathbf{I}_0), \tag{6.34}$$

with equal variance σ_B^2 in each element β_i of β for $i = 1, \ldots, r_b$. Thus $f(\beta)$ is given by

$$f(\beta) \propto \exp\left\{ -\frac{1}{2\sigma_B^2} ||\beta||_2^2 \right\}. \tag{6.35}$$

Placing (6.33) and (6.35) into (6.32) and taking logarithm, we have

$$\begin{aligned} \hat{\beta}_{iso} &= \underset{\beta}{\operatorname{argmax}} \log\{f(\beta|\mathbf{x})\} \\ &= \underset{\beta}{\operatorname{argmax}} \log\{f(\mathbf{x}|\beta) f(\beta)\} \\ &= \underset{\beta}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma_0^2} ||\mathbf{x} - \mathbf{B}\beta||_2^2 - \frac{1}{2\sigma_B^2} ||\beta||_2^2 \right\} \\ &= \underset{\beta}{\operatorname{argmin}} \left\{ ||\mathbf{x} - \mathbf{B}\beta||_2^2 + \theta_0 ||\beta||_2^2 \right\}, \end{aligned} \tag{6.36}$$

where $\theta_0 = \sigma_0^2 / \sigma_B^2$. The estimate of β in (6.36) is exactly the same as the MSSD-i estimate in (6.14). In this fashion, parameter θ_0 effectively controls the degree of shrinkage through the ratio of two variances σ_0^2 and σ_B^2.

Similarly, the prior distribution of γ is in fact assumed to be

$$\gamma \sim \mathscr{N}(\mathbf{0}, \sigma_T^2 \mathbf{I}_t), \tag{6.37}$$

where $\mathbf{I}_t$ is a $r_t \times r_t$ identity matrix and therefore it results in a zero mean distribution of α with an $(r_t + r_b) \times (r_t + r_b)$ diagonal covariance matrix

$$\begin{bmatrix} \sigma_T^2 \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \sigma_B^2 \mathbf{I}_0 \end{bmatrix}. \tag{6.38}$$

Then $f(\alpha)$ is given by

$$f(\alpha) \propto \prod_{i=1}^{r_t+r_b} \exp\left\{-\frac{1}{2\sigma_i^2}\alpha_i^2\right\}, \tag{6.39}$$

where $\sigma_i = \sigma_T$ for $i = 1,\ldots,r_t$ and $\sigma_i = \sigma_B$ for $i = r_t + 1,\ldots,r_t + r_b$.

When $\sigma_T = \sigma_B$ and we let both of them to be σ_α, (6.39) can be simplified to

$$f(\alpha) \propto \exp\left\{-\frac{1}{2\sigma_\alpha^2} \|\alpha\|_2^2\right\}. \tag{6.40}$$

Then the MAP estimate of α is then given by

$$\hat{\alpha}_{iso} = \underset{\alpha}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{V}\alpha\|_2^2 + \theta_1 \|\alpha\|_2^2 \right\}, \tag{6.41}$$

where $\theta_1 = \sigma_1^2/\sigma_\alpha^2$ is the shrinkage parameter. This is also in the same form of the MSSD-i estimate of α in (6.15), in particular if we assume $\sigma_T = \sigma_B$.

We can further generalise (6.41) to a slightly-adaptive shrinkage model:

$$\hat{\alpha}_{iso} = \underset{\alpha}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{V}\alpha\|_2^2 + \sum_{i=1}^{r_t+r_b} \theta_{1i}\alpha_i^2 \right\}. \tag{6.42}$$

In (6.42), when $i = 1,\ldots,r_t$, we have $\theta_{1i} = \sigma_1^2/\sigma_T^2$, and when $i = r_t + 1,\ldots,r_t + r_b$, we have $\theta_{1i} = \sigma_1^2/\sigma_B^2$.

6.4.2 Prior distributions of β and α in MSSD-a

For MSSD-a, the prior distribution of β is in fact assumed to be

$$\beta \sim \mathcal{N}(\mathbf{0}, \theta_B \Lambda_B), \tag{6.43}$$

where Λ_B is a $r_b \times r_b$ diagonal matrix with eigenvalues $\lambda_1^b, \dots \lambda_{r_b}^b$ on the diagonal, and θ_B is a parameter scaling the eigenvalue matrix Λ_B. It means that each β_i, for $i = 1, \dots, r_b$, is assumed to have its own variance instead of an equal variance assumed in the MSSD-i. Then $f(\beta)$ in MSSD-a is given by

$$f(\beta) \propto \exp\left\{-\frac{1}{2}\beta^T(\theta_B\Lambda_B)^{-1}\beta\right\}. \tag{6.44}$$

Placing (6.33) and (6.44) into (6.32) and taking logarithm, we have the MAP estimator of β in MSSD-a:

$$\hat{\beta}_{aniso} = \underset{\beta}{\operatorname{argmin}}\left\{(\mathbf{x}-\mathbf{B}\beta)^T(\mathbf{x}-\mathbf{B}\beta) + \theta_0\beta^T\Lambda_B^{-1}\beta\right\}, \tag{6.45}$$

where $\theta_0 = \sigma_0^2/\theta_B$. This is the same as the MSSD-a estimate of β in (6.22).

The prior distribution of γ is assumed to be

$$\gamma \sim \mathcal{N}(\mathbf{0}, \theta_T\Lambda_\mathbf{T}), \tag{6.46}$$

where $\Lambda_\mathbf{T}$ is a $r_t \times r_t$ diagonal matrix with different eigenvalues $\lambda_1^t, \dots \lambda_{r_t}^t$ on the diagonal. Therefore the distribution of α is a zero mean distribution with an $(r_t + r_b) \times (r_t + r_b)$ diagonal covariance matrix

$$\begin{bmatrix} \theta_T\Lambda_T & \mathbf{0} \\ \mathbf{0} & \theta_B\Lambda_B \end{bmatrix}, \tag{6.47}$$

i.e. $\alpha \sim \mathcal{N}(\mathbf{0}, \theta_v\Lambda_\mathbf{V})$, when $\theta_T = \theta_B$ and both of them are equal to θ_v; and $\Lambda_\mathbf{V} = \begin{bmatrix} \Lambda_T & \mathbf{0} \\ \mathbf{0} & \Lambda_B \end{bmatrix}$.

Similar to (6.39), $f(\alpha)$ is given by

$$f(\alpha) \propto \exp\left\{-\frac{1}{2}\alpha^T(\theta_v\Lambda_V)^{-1}\alpha\right\}, \tag{6.48}$$

Then the MAP estimate of α becomes

$$\hat{\alpha}_{aniso} = \underset{\alpha}{\operatorname{argmin}} \left\{ (\mathbf{x} - \mathbf{V}\alpha)^T (\mathbf{x} - \mathbf{V}\alpha) + \theta_1 \alpha^T \Lambda_V^{-1} \alpha \right\}, \tag{6.49}$$

where $\theta_1 = \sigma_1^2 / \theta_v$. This is also exactly the same as the MSSD-a estimate of α in (6.23).

To sum up, in contrast to the improper uniform distributions assumed in the MSD, two different prior distributions are assumed by the proposed MSSD-i and MSSD-a for the abundance vectors β and γ for the background and target subspaces. In the MSSD-i, a common variance is assumed on each coefficient in the form of an scaled identity matrix (see (6.35) and (6.37)). In the MSSD-a, unequal variances are assumed for individual coefficients in the form of an scaled eigenvalue matrix (see (6.44) and (6.46)).

6.5 Underlying links among MSD, MSSD-i and MSSD-a

The conventional MSD preserves the leading eigenvectors to form the subspaces **B** and **T**, which is essentially a basis selection process. Specifically, it drops eigenvectors of small eigenvalues, effectively forcing these eigenvalues to be 0. At the same time, eigenvalues of the preserved eigenvectors are effectively forced to be equal to each other. The proposed MSSD-i and MSSD-a on the other hand, preserve all available eigenvectors and control the degrees of shrinkage of abundance by imposing l_2-norm regularisation. Specifically, MSSD-i imposes an isotropic shrinkage over the full eigenspace, while MSSD-a is anisotropic using eigenvalues to adapt the shrinkage for different directions.

From the Bayesian perspective, the conventional MSD implies a non-informative uniform distribution for the coefficient vectors over infinite interval. Different from the MSD, the proposed MSSD-i and MSSD-a imply Gaussian prior distributions for the coefficient vectors. MSSD-i assumes an equal variance for each coefficient, while MSSD-a assumes different variances for different coefficients which are based on eigenvalues.

Nevertheless, it is readily seen that the MSSD-i method is equivalent to a ridge regression on the eigenspace. Also, as a kind of dual representation, the proposed MSSD-a can also be derived as a ridge regression on the original sample space. Specifically regarding this derivation of MSSD-a, if we apply the LMM in the original N_b-dimensional sample space of the $p \times N_b$ training sample matrix $\mathbf{M}_B$ under model H_0 with mean-corrected measurement, i.e. $\mathbf{M}_B$ is a mean-corrected matrix and pixel $\mathbf{x}$ is represented as a linear mixture of N_b samples, we have

$$\mathbf{x} = \mathbf{M}_B \mathbf{a} + \mathbf{n}, \tag{6.50}$$

where $\mathbf{a}$ is an $N_b \times 1$ coefficient vector, and the ridge regression problem becomes

$$\hat{\mathbf{a}}_{iso} = \underset{\alpha}{\operatorname{argmin}} \{ \|\mathbf{x} - \mathbf{M}_B \mathbf{a}\|_2^2 + \theta_M \|\mathbf{a}\|_2^2 \}, \tag{6.51}$$

where $\hat{\mathbf{a}}_{iso}$ is the shrunken estimator of $\mathbf{a}$ and θ_M is the parameter controlling the shrinkage. The solution of $\hat{\mathbf{a}}_{iso}$ is

$$\hat{\mathbf{a}}_{iso} = (\mathbf{M}_B^T \mathbf{M}_B + \theta_M \mathbf{I}_b)^{-1} \mathbf{M}_B^T \mathbf{x}, \tag{6.52}$$

where $\mathbf{I}_b$ is a $N_b \times N_b$ identity matrix.

Following the notation in [9], if we perform the singular value decomposition (SVD) on $\mathbf{M}_B$, saying $p < N_b$, we obtain

$$\mathbf{M}_B = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{6.53}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $p \times p$ and $N_b \times N_b$ orthogonal matrices, with columns of $\mathbf{U}$ spanning the column space of $\mathbf{M}_B$ and columns of $\mathbf{V}$ spanning the row space of $\mathbf{M}_B$; and $\mathbf{D}$ is a $p \times N_b$ rectangular diagonal matrix with singular values of $\mathbf{M}_B$ on the diagonal in descending order. Based on the relationship between this SVD and the eigen-decomposition of covariance matrix $\mathbf{C}_B$ in MSSD-a, we have:

1) $\mathbf{U} = \mathbf{B}$ ($r_b = p$ in this case) and

2) $\mathbf{D}_p^2 = N_b \Lambda_{\mathbf{B}}$,

where $\mathbf{D}_p$ is a $p \times p$ diagonal matrix of the first p columns of $\mathbf{D}$. Then the solution of $\mathbf{M}_B \hat{\mathbf{a}}_{iso}$ has the following form:

$$\begin{aligned}
\mathbf{M}_B \hat{\mathbf{a}}_{iso} &= \mathbf{M}_B(\mathbf{M}_B^T \mathbf{M}_B + \theta_M \mathbf{I}_b)^{-1} \mathbf{M}_B^T \mathbf{x} \\
&= \mathbf{U}\mathbf{D}_p(\mathbf{D}_p^2 + \theta_M \mathbf{I}_p)^{-1} \mathbf{D}_p \mathbf{U}^T \mathbf{x} \\
&= \mathbf{B}(N_b \Lambda_{\mathbf{B}})(N_b \Lambda_{\mathbf{B}} + \theta_M \mathbf{I}_p)^{-1} \mathbf{B}^T \mathbf{x} \\
&= \mathbf{B}(\mathbf{I}_p + \frac{\theta_M}{N_b} \Lambda_{\mathbf{B}}^{-1})^{-1} \mathbf{B}^T \mathbf{x} \\
&= \mathbf{B}(\mathbf{I}_p + \theta_0 \Lambda_{\mathbf{B}}^{-1})^{-1} \mathbf{B}^T \mathbf{x},
\end{aligned} \tag{6.54}$$

where $\mathbf{I}_p$ is a $p \times p$ identity matrix and $\theta_0 = \frac{\theta_M}{N_b}$. This is indeed the same as the solution of $\mathbf{B}\hat{\beta}_{aniso}$, where $\hat{\beta}_{aniso}$ is given by (6.26) in the MSSD-a method. Similar derivation can also be obtained for model H_1, which we omit here.

6.6 Experimental studies

In the experimental studies, we compare the performances of the MSSD-i, MSSD-a and MSD methods by applying them to a real HSI dataset call Hymap image, which has been used for evaluation in section 5.4.2 of Chapter 5. To measure the detection performances of the three methods, the receiver operating characteristic (ROC) curve is used, in which a good detection curve should lie near to the top left. In pair with ROC curve, we also employ the area under curve (AUC) statistics to measure the detection results quantitatively.

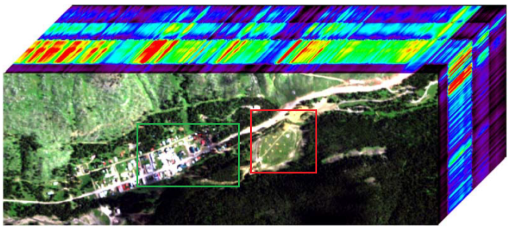


Figure 6.1: The Hymap scene. Two sub-images are cropped for evaluation.

Details of the Hymap dataset have been introduced in section 5.4.2. As shown in Figure 6.1 in this chapter, we cropped two regions of interests (ROIs) into two separate HSI cubes, with the pixel size of 100 × 120 and 100 × 150, respectively. The ROIs of fabric panels (F1, F2, F3 and F4) and their corresponding target locations are shown in Figure 6.2, and the ROIs of three vehicles (V1, V2 and V3) and their corresponding target locations are shown in Figure 6.3.

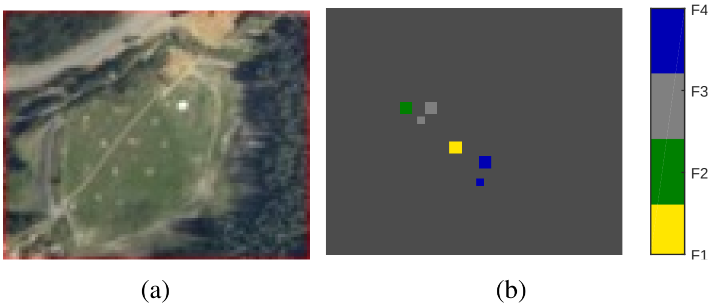


Figure 6.2: Target F1, F2, F3 and F4: (a) Hymap image scene of fabric panels; (b) locations of fabric panels. Pixels in different colours indicate different targets. The pixels sizes of ROIs of F1, F2, F3 and F4 are 25, 25, 34 and 34, respectively.

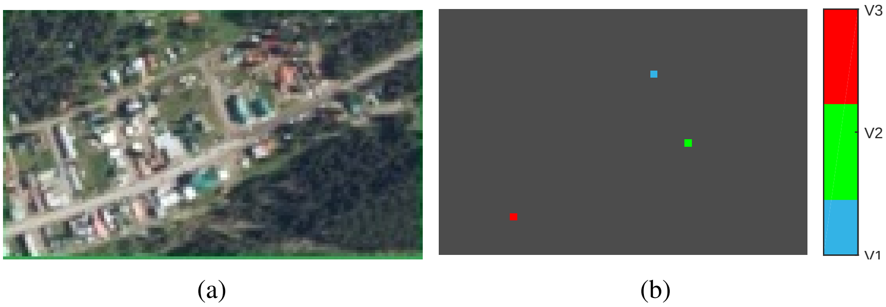


Figure 6.3: Target V1, V2, V3: (a) Hymap image scene of vehicles; (b) locations if vehicles. Pixels in different colours indicate different targets. The pixels sizes of ROIs of V1, V2,and V3 are 9, 9 and 9, respectively.

There are two widely accepted experiment settings regarding the target pixels in the Hymap scene: 1) In [81, 2, 82, 92], only one target pixel of each desired target is assumed to be in the HSI as we have done in Chapter 5; 2) whereas in [80], pixels within the ROIs of desired targets are all regarded as target pixels. In the setting 1), no target pixels are available for training. As a consequence, the parameters of the models have to be manually set. While in the setting 2), the target pixels can be randomly split into a training set and a test set and we can tune parameters for models. The setting 2) is believed to be a tougher condition for target detection than the setting 1). In this chapter, we adopt the setting 2) in the evaluation of the compared methods for fair comparison.

We randomly choose 2-3 labelled target pixels for training and the rest target pixels for testing; and randomly choose around 10% background pixels for training and the rest background pixels for testing. Summaries of the numbers of training and test pixels of sub-images which are used for detecting fabrics and vehicles are given in Table 6.1 and Table 6.2, respectively.

Table 6.1: Target fabrics: the number of target pixels for training and test in the sub-image shown in Figure 6.2.

	Target pixels			Background pixels		
Target	training	test	total	training	test	total
F1	2	23	25	1197	10778	11975
F2	2	23	25	1197	10778	11975
F3	3	31	34	1196	10770	11966
F4	3	31	34	1196	10770	11966

Table 6.2: Target vehicles: the number of target pixels for training and for test in the sub-image shown in Figure 6.3.

	Target pixels			Background pixels		
Target	training	test	total	training	test	total
V1	2	7	9	1499	13492	14991
V2	2	7	9	1499	13492	14991
V3	2	7	9	1499	13492	14991

6.6.1 Parameter settings

In real target detection problems, training examples of background pixels are not available. It is often assumed that the target presence in the scene is so sparse that if we extract neighbourhood pixels around a test pixel but not close to the test pixel, this neighbourhood can be seen as a replacement for background samples. Therefore as with [83, 15, 23, 24], we adopt the double concentric sliding window [23], a local and adaptive approach to extract the background pixels from the neighbourhood of each test pixel. Specifically, the concentric window separates the local area around each pixel into two regions, an inner window region (IWR) and an outer window region (OWR). The IWR is used to enclose the target of interest to be detected. The OWR is used to model the local backgrounds around the target region. An illustration of the double concentric window is shown in Figure 6.4. The determination of the window sizes is difficult therefore as with [24][93], the window sizes are set empirically. In our cases, the sizes of OWR and of IWR are set as 17×17 and 7×7 for detecting fabrics panels, and 15×15 and 5×5 for detecting vehicles, respectively. Therefore, for each test pixel $\mathbf{x}$ in Figure 6.2, the number of training background pixels is $N_b = 240$; for each test pixel $\mathbf{x}$ in Figure 6.3, the number of training background pixels is $N_b = 200$, which are all greater than the dimension of the spectra $p = 126$.

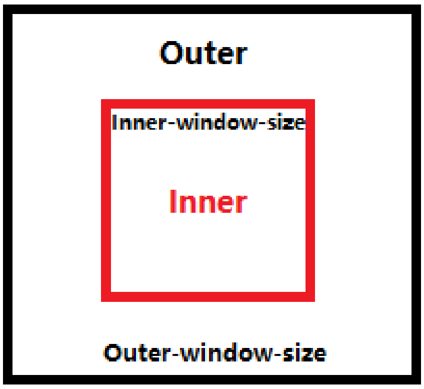


Figure 6.4: An illustration of the dual window adopted for sampling background adaptively.

For each target pixel $\mathbf{x}_i$ in an HSI, we use the mean-centred background samples extracted by double concentric window to compute the covariance matrix $\mathbf{C}_i$, where $i = 1, \ldots, N$ and N is the total number of test pixels in the HSI. Then the columns of the subspace $\mathbf{B}$ are created by the eigen-decomposition of $\mathbf{C}_i$. Since we only have one prior spectrum for each desired target, we subtract the background mean μ_i of the local adaptive background samples around the test pixel $\mathbf{x}_i$ from the target spectrum $\mathbf{m}_t$, i.e. $\mathbf{m}_t - \mu_i$, then normalise $\mathbf{m}_t - \mu_i$ to have a unit l_2-norm as the target subspace $\mathbf{T}$. As a result, the columns in $\mathbf{B}$ and $\mathbf{T}$ all have unit l_2-norms and are independent of each other.

Regarding the variance σ_T of γ defined in MSSD-i (6.37) and the eigenvalue matrix Λ_T of γ defined in MSSD-a (6.46), we set both σ_T and Λ_T to be ∞, since we only have one target spectrum to construct $\mathbf{T}$ and there is no variance can be estimated in the target subspace. It means that in the real application of target detection where only one target spectrum is available, we actually do not shrink the size of abundance γ corresponding to the target basis vector in the H_1 model in both MSSD-i and MSSD-a, and let the projection of a test pixel onto the target basis vector to be as much as possible.

In the conventional MSD to be evaluated on the Hymap image, there is only one unknown parameter to be tuned, which is the number of preserved leading eigenvectors $r_b(r_b \leqslant p)$ for the subspace $\mathbf{B}$. For each desired target, there is only one target spectrum, i.e. $N_t = r_t = 1$. In the propose MSSD-i and MSSD-a, two unknown parameters in (6.16) and (6.25) need to be tuned: the shrinkage parameters θ_0 and θ_1. The optimal values of r_b of MSD, θ_{iso0} and θ_{iso1} of MSSD-i and θ_{aniso0} and θ_{aniso1} of MSSD-a tuned by the training dataset are listed in Table 6.3.

Table 6.3: Parameter settings of MSD, MSSD-i and MSSD-a.

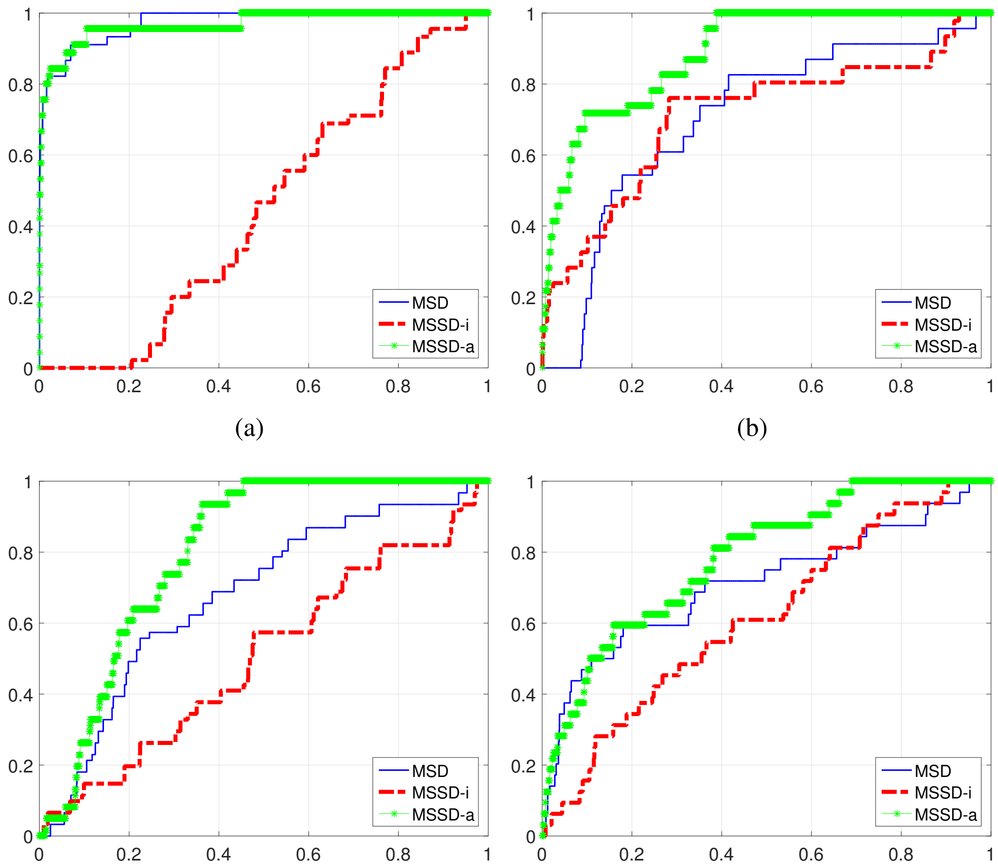
	MSD	MSSD-i		MSSD-a	
	r_b	θ_{iso0}	θ_{iso1}	θ_{aniso0}	θ_{aniso1}
F1	2	1e-09	1e-07	1e-03	1e-03
F2	2	1e-09	3e-07	7e-07	1e-09
F3	14	1e-09	1e-08	1e-08	3
F4	2	1e-09	1e-08	3e-03	3e-03
V1	124	1e-09	1e-09	3e-07	1e-09
V2	6	1e-09	1e-07	1e-07	1e-06
V3	124	1e-09	1e-07	3	5e+1

6.6.2 Detection performance

The detection performances of MSD, MSSD-i and MSSD-a are listed in Table 6.4 and shown in Figure 6.5 and Figure 6.6. Firstly, we can observe that both MSSD-i and MSSD-a can outperform MSD in detecting F2, V1, V2 and V3. Specifically, MSSD-a can improve the detection performance significantly, compared with the conventional MSD method. Among the seven types of targets, MSSD-a improves

Table 6.4: Detection performance of MSD, MSD-i and MSSD-a measured in the AUC statistics. The best performance is indicated in boldface.

	MSD	MSSD-i	MSSD-a
F1	**0.974**	0.662	0.968
F2	0.706	0.713	**0.888**
F3	0.679	0.506	**0.801**
F4	0.711	0.656	**0.784**
V1	0.673	**0.845**	0.726
V2	0.647	0.752	**0.778**
V3	0.643	0.664	**0.676**



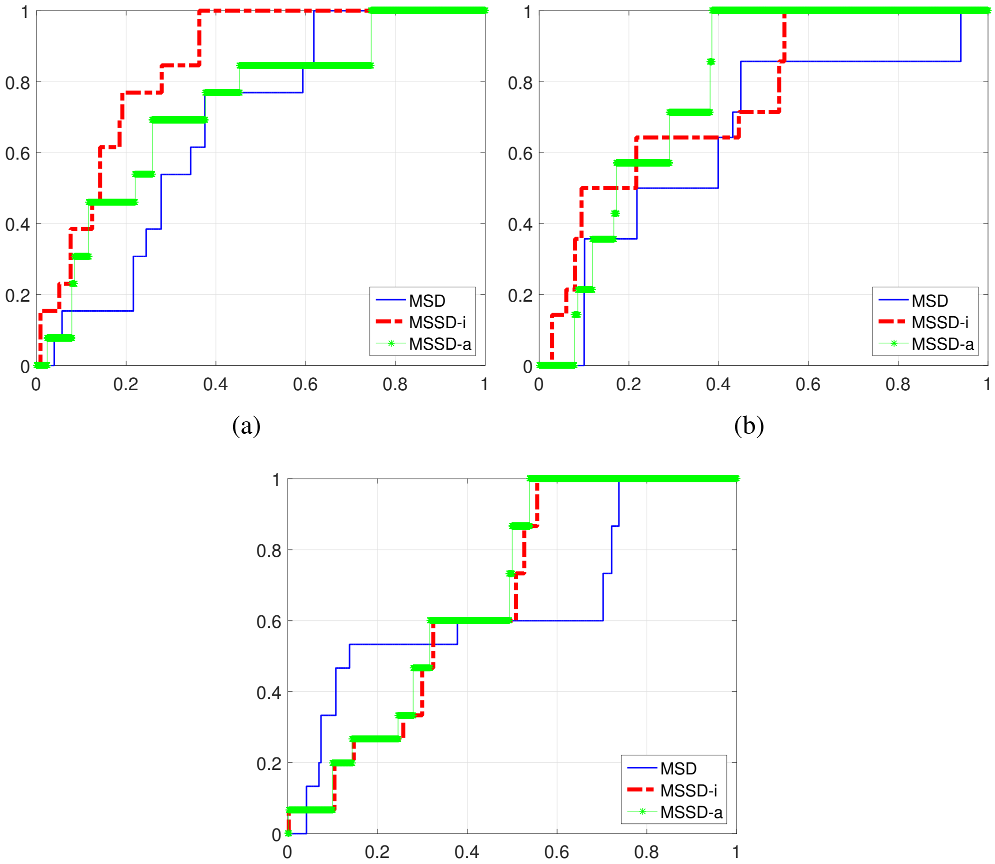
(a)

(b)

(c)

(d)

Figure 6.5: ROC curves of detecting fabric panels: (a) F1; (b) F2; (c) F3; (d) F4. The x-axis and y-axis are false positive rate and true positive rate, respectively.



(a)

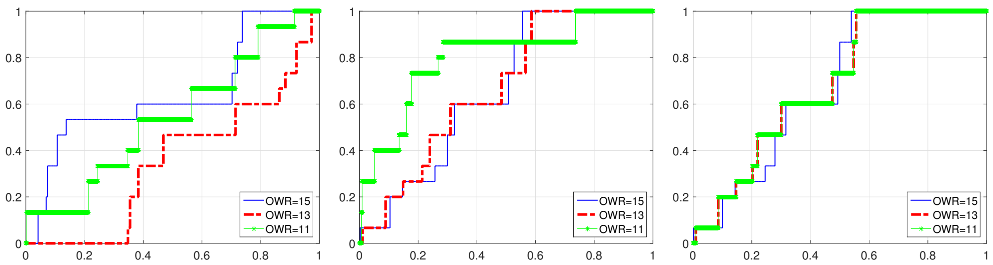
(b)

(c)

Figure 6.6: ROC curves of detecting vehicles: (a) V1; (b) V2; (c) V3. The x-axis and y-axis are false positive rate and true positive rate, respectively.

six of them, F2, F3, F4, V1, V2 and V3, from MSD. Secondly, MSSD-i improves the performance on detecting F2, V1, V2 and V3, compared with MSD. These results suggest that introducing l_2-norm regularisation terms into MSD can improve the detection performance.

We shall note that MSD has better performance on detecting F1 than MSSD-i and MSSD-a. However, MSSD-a still has competitive performance as MSD on detecting F1 (0.9680 vs. 0.9742); it also illustrates that preserving the information from the eigenvalues in the prior distribution of abundance by MSSD-a can have a more stable detection performance than MSSD-i, which assumes an equal variance in the prior distribution.



(a) (b) (c)

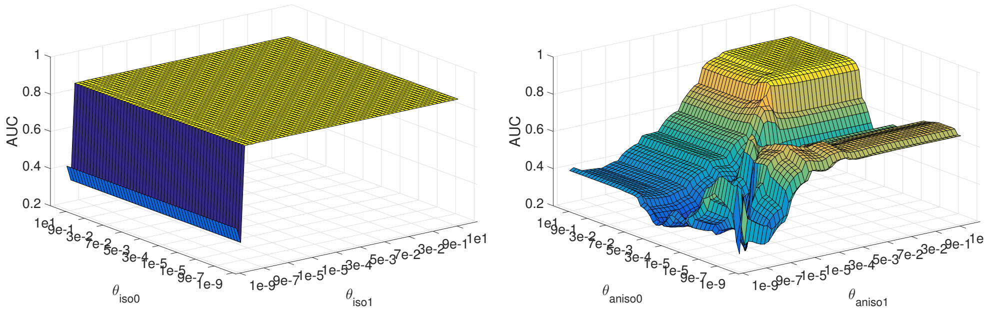
Figure 6.7: Effects of window sizes on detecting V3: (a) MSD; (b) MSSD-i; (c) MSSD-a. The IWR size is fixed to be 5×5, and the OWR size varies from 15×15, 13×13 to 11×11.

6.6.3 Discussion on effects of parameters

We further investigate the effects of parameters on the performances of detectors.

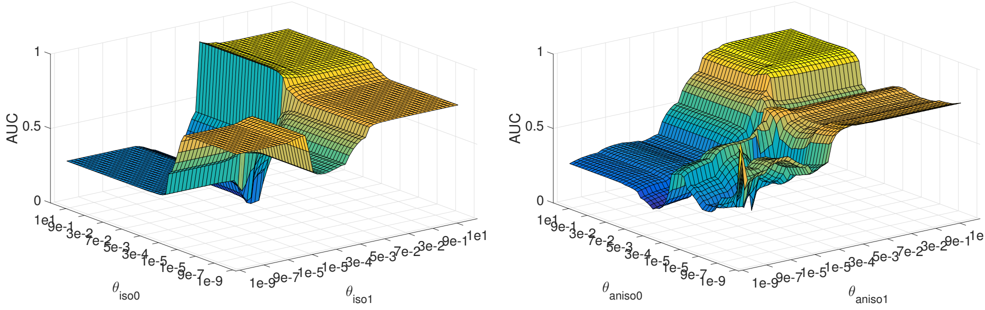
Firstly, the effects of window sizes on the performances of MSD, MSSD-i and MSSD-a for detecting target V3 are illustrated in Figure 6.7; the results for detecting other targets are of a similar pattern. It is true that all parameters, such as window sizes of OWR and IWR and shrinkage parameters θ_0 and θ_1, jointly affect the performances of detectors. Here for simplicity of exploring the effect of window sizes alone, we fix the values of other parameters (r_b, θ_0 and θ_1) of corresponding detectors as those in Table 6.3, and fix the size of IWR. The ROC curves of the detectors under three different sizes of OWR are plotted in Figure 6.7. We can observe that MSD and MSSD-i are sensitive to OWR, whilst MSSD-a is more stable. This indicates that MSSD-a is more robust to the variation of background samples, and preserving variances of the original data is beneficial in terms of the stability of detection performance.

Secondly, we investigate the effects of shrinkage parameters by sweeping the parameter spaces of θ_{iso0} and θ_{iso1} of MSSD-i and θ_{aniso0} and θ_{aniso1} of MSSD-a. Here due to much higher computational complexity for the large number of test pixels, we show the results for the training pixels as illustration. We show the results of MSSD-i and MSSD-a for detecting V3 under two sets of window sizes in Figure 6.8 and Figure 6.9, respectively. Again, the results for detecting other targets are of a similar pattern.



(a) (b)

Figure 6.8: For OWR of size 15 × 15 and IWR of size 5 × 5. (a) MSSD-i: effects of θ_{iso0} and θ_{iso1} on detecting V3; (b) MSSD-a: effects of θ_{aniso0} and θ_{aniso1} on detecting V3.



(a) (b)

Figure 6.9: For OWR of size 11 × 11 and IWR of size 5 × 5. (a) MSSD-i: effects of θ_{iso0} and θ_{iso1} on detecting V3; (b) MSSD-a: effects of θ_{aniso0} and θ_{aniso1} on detecting V3.

We can observe that the AUC surface of MSSD-i is smoother than that of MSSD-a in both sets of window sizes. This pattern is particularly clear in the setting that OWR is of size 15×15 and IWR is of size 5×5, where MSSD-i is not sensitive to θ_{iso0}, as shown in Figure 6.8(a). Technically, the reason for this 'extreme' pattern is because the number of training background pixels $N_b = 200$ is greater than the pixel dimension $p = 126$, which leads to the result that $r_b = p$ and the $p \times p$ matrix $\mathbf{B}$ represents a full space. Therefore for each pixel $\mathbf{x}_j$, the RSS $e_0^{iso}(\mathbf{x}_j)$ in (6.19) can be simplified to

$$\begin{aligned}
e_0^{iso}(\mathbf{x}_j) &= \mathbf{x}_j^T\left(\mathbf{I}-\mathbf{B}\left((1+\theta_{iso0})\mathbf{I}_0\right)^{-1}\mathbf{B}^T\right)\mathbf{x}_j \\
&= \mathbf{x}_j^T\mathbf{x}_j - \frac{1}{1+\theta_{iso0}}\mathbf{x}_j^T\mathbf{B}\mathbf{B}^T\mathbf{x}_j \\
&= \mathbf{x}_j^T\mathbf{x}_j - \frac{1}{1+\theta_{iso0}}\mathbf{x}_j^T\mathbf{x}_j \\
&= \frac{\theta_{iso0}}{1+\theta_{iso0}}\mathbf{x}_j^T\mathbf{x}_j \; .
\end{aligned} \tag{6.55}$$

In (6.55), $e_0^{iso}(\mathbf{x}_j)$ is equivalent to scaling the l_2-norm of every pixel $\mathbf{x}_j$ with a scaler $\frac{\theta_{iso0}}{1+\theta_{iso0}}$. The detection ratio (6.21) is then scaled by $\frac{\theta_{iso0}}{1+\theta_{iso0}}$ as well when θ_{iso1} is fixed. As a result, the AUC of MSSD-i does not depend on θ_{iso0}, as shown in Figure 6.8(a). However, in Figure 6.9(a) when the OWR size reduces to 11 × 11, the number of background samples N_b becomes 96 and thus $N_b < p$, and the AUC becomes dependent on θ_{iso0}, because now $e_0^{iso}(\mathbf{x}_j)$ cannot be simplified to (6.55) and θ_{iso0} affects the AUC.

As a by-product, the above analysis suggests a guideline on the use of MSSD-i: when $N_b < p$, both shrinkage parameters θ_{iso0} and θ_{iso1} should be tuned during the training phase; when $N_b > p$, only θ_{iso1} needs to be tuned and θ_{iso0} can be arbitrary. For example, the values of θ_{iso0} in Table 6.3 are in the case of $N_b > p$ and are not necessary to be 1e-09; instead, they can be any values.

For MSSD-a, the detection performance varies with both θ_{aniso0} and θ_{aniso1}, as shown in Figure 6.8(b) and Figure 6.9(b).

Finally, it is worth discussing why MSSD-a is more favourable than MSSD-i, as indicated by the test results listed in Table 6.4. We believe a big reason for this is that MSSD-a considers both eigenvectors and eigenvalues to preserve the information of the data for the shrinkage, while MSSD-i considers only eigenvectors. MSSD-i essentially assumes an equal variance in the prior distribution of each coefficient in the eigenspace, while MSSD-a assumes different variances for different coefficients based on eigenvalues. Hence the latter preserves the variances of the original data and can adapt to the shrinkage in different directions in the eigenspace better than the former.

6.7 Conclusion

We have proposed a new approached to hyperspectral target detection, called the matched shrunken subspace detector (MSSD), and its two implementations, MSSD-i with isotropic shrinkage and MSSD-a with anisotropic shrinkage. The MSSD introduces the l_2-norm regularisation into the popular matched subspace detector (MSD), seeking more reliable projection for the hypothesis models H_0 and H_1. From the Bayesian perspective, the added regularisation terms preserve non-uniform prior distributions of the coefficient vectors in the models. Both MSSD-i and MSSD-a can reduce the variances of the coefficients and result in more stable estimators. The links among MSD, MSSD-i and MSSD-a have also been discussed in detail, and the two proposed methods have shown superior detection performance compared with the conventional MSD on the real dataset of Hymap.

Chapter 7

HSI Target Detection: Matched Shrunken Cone Detectors (MSCD)

7.1 Introduction

With the help of remote sensors, hyperspectral imaging has become an important scientific tool for various fields of real-world applications. In the analysis of hyperspectral images (HSIs), target detection is a major task, which aims to detect small objects or anomalies in a hyperspectral image. Typical target detection applications include military defence, agricultural management and mineral detection.

Target detection is essentially a binary classification problem, of which the task is to determine if an HSI pixel is a target spectrum or a background spectrum. Hence, target detection can be regarded as a binary hypothesis model with two competing hypotheses: the null hypothesis H_0 for the absence of the target; and the alternative hypothesis H_1 for the presence of the target. Binary hypothesis models for target detection have been nicely reviewed in [13, 14, 15, 16].

Target objects often appear as sub-pixels in an HSI. That is, the spectrum of an HSI pixel can be a mixture of different component spectra of materials. These component spectra are usually termed endmembers. To model the mixture of an HSI pixel, the linear mixing model (LMM) [5] has been widely adopted. The underlying assumption of LMM is that an HSI pixel can be approximated by a linear combination of endmembers with different fractions. When a target pixel presents, its spectrum is decomposed as a linear combination of background endmembers and target endmembers; in contrast, when a background pixel presents, its spectrum is fully represented by background endmembers.

Within the framework of binary hypothesis modelling, researches have explored a variety of techniques and extensions on the basis of LMM. Since it is difficult to obtain comprehensive spectral libraries to serve as the endmembers for all desired targets, many methods focus on extract endmembers directly from HSIs. On the one hand, provided with a large number of background samples, subspace methods have been widely developed for target detection. Typical methods, such as the orthogonal subspace projection detector (OSP) [18] and matched subspace detector (MSD) [7], adopt the leading eigenvectors (with dominant eigenvalues) as the subspace bases and implicitly the endmembers. On the other hand, sparse representation (SR) techniques [10] originating from compressed sensing have been recently studied in the HSI analysis [4]. For HSI target detection, SR-based methods, such as sparse target detection (STD) [23], sparse representation-based binary hypothesis model (SRBBH) [24] and hybrid sparsity and statistics detector [94], model a test HSI pixel as a linear combination of only a few training samples (aka atoms of an over-complete dictionary). It implicitly regards the atoms as endmembers, hence the SR-based methods can be viewed as being developed in the original sample space.

These methods can be further extended to nonlinear mixing models. The kernel methods, which aim to define a model in a high-dimensional feature space associated with a nonlinear mapping of input data, have also been studied for HSI target detection [65, 69, 70]. In [65], subspace methods such as MSD, OSP have been extended to their kernel versions. Kernelisation of the SR-based methods has been also developed, such as kernel-based STD [69] and kernel-based SRBBH [70].

For the sake of physical interpretations, HSIs as instances of natural signals possess *non-negative* properties for both hyperspectral signatures and the abundance coefficients. A number of investigations have focused on the non-negative matrix factorisation (NMF) [49, 52] for HSI unmixing problems. NMF factorises a sample data matrix into two low-dimensional matrices in terms of bases and corresponding coefficients, and explicitly enforces the non-negative constraints on both of them. However, in the past researches of HSI target detection [7, 18, 4, 23, 24, 94, 65, 69, 70], the non-negativity properties have not been considered yet, particularly for the abundance coefficients. If we use the samples directly from HSIs as endmembers, it is desirable to impose the non-negative constraints on the coefficients. In this way, both endmembers and coefficients are non-negative, such that this physical characteristic of hyperspectral signatures are modelled.

Statistically, the estimation of non-negatively-constrained coefficients in the LMM is often termed non-negative least squares (NNLS) [53]. Geometrically, the NNLS estimation induces a cone-shape representation [12]. Suppose that a hyperspectral spectrum $\mathbf{x}$ is a p-dimensional vector, and that there are K types of materials, i.e. K endmembers potentially constituting an HSI pixel, which are represented by $\mathbf{m}_1, \ldots, \mathbf{m}_K$ with each $\mathbf{m}_k$ also a p-dimensional vector. Then the cone-based representation of pixel $\mathbf{x}$ expresses the spectral signature of $\mathbf{x}$ as a *non-negative* linear combination of endmembers $\mathbf{m}_1, \ldots, \mathbf{m}_K$ with corresponding *non-negative* abundance fractions $a_1, \ldots, a_K$, such that $a_k \geq 0$ for $k = 1, \ldots, K$. More specifically, a *convex cone* $\mathbb{C}$ is defined as

$$\mathbb{C} : \left\{ \mathbf{x} | \mathbf{x} = \sum_{k=1}^{K} a_k \mathbf{m}_k = \mathbf{M}\mathbf{a}, a_k \geq 0 \right\}, \tag{7.1}$$

where $\mathbf{M}$ is a $p \times K$ matrix whose columns are the K endmembers spectra $\mathbf{m}_k = [m_{k,1}, \ldots, m_{k,p}]^T$; and $\mathbf{a} = [a_1, \ldots, a_K]^T$ denotes the abundance vector. For the non-negative LMM, an additional noise term is also considered:

$$\mathbf{x} = \mathbf{M}\mathbf{a} + \mathbf{n}, a_k \geq 0, \tag{7.2}$$

where the vector $\mathbf{n}$ is assumed to be the Gaussian white noise, i.e. $\mathbf{n} \sim (\mathbf{0}, \sigma^2 \mathbf{I}_p)$, where $\mathbf{I}_p$ is the $p \times p$ identity matrix.

It is worth noting that, LMM-based methods may suffer from the problem of high variance of coefficients estimations. To this end, shrinkage methods [9] have been developed in statistical learning. Typical shrinkage methods include l_2-norm regularisation, also known as *ridge regression* or Tikhonov regularisation, and l_1-norm regularisation, also known as *lasso*. For the convex cone analysis, these regularisations have also been studied, mainly on the computational efficiency of the algorithms developed based on the NNLS [95, 96, 97, 98].

In this chapter, to account for the non-negativity as well as the shrinkage of the coefficients of the convex cone model (7.2) for HSI target detection, we propose a new approach called the matched shrunken cone detector (MSCD). Specifically, on the cone representations we propose to shrink the abundance coefficients of target endmembers and background endmembers by imposing constraints; we propose two working models with the l_2-norm and l_1-norm regularisations, respectively. We call these two methods MSCD-l_2 and MSCD-l_1. Equally important, we derive the proposed MSCD from the Bayesian perspective, showing that MSCD-l_2 and MSCD-l_1 can be derived if a *multivariate half-Gaussian distribution* [99] and a *multivariate half-Laplace distribution* [100] are assumed as the prior distributions of the coefficient vectors. To our knowledge, it is the first time that the cone representations with the l_2-norm and l_1-norm regularisations are derived from the Bayesian perspective, as well as the prior distributions identified.

The main contributions of this chapter are summarised as follows: 1) we propose a regularised cone-based representation approach called MSCD for HSI target detection; 2) we propose two working models of MSCD, namely MSCD-l_2 and MSCD-l_1, by incorporating l_2-norm and l_1-norm regularisations into the cone-based representation (7.2); 3) we derive the proposed MSCD-l_2 and MSCD-l_1 from the Bayesian perspective, showing they imply a multivariate half-Gaussian distribution and a multivariate half-Laplace distribution as the prior distributions for the coefficients; 4) and we illustrate the superior detection performance of the proposed models, compared with the typical subspace and SR-based methods, on two real hyperspectral datasets for sub-pixel and full-pixel target detections, respectively.

In the rest of the chapter, section 7.2 reviews the binary hypothesis model in terms of the LMM-based likelihood ratio test. Section 7.3 introduces the propose

MSCD. Section 7.4 shows the derivations of the proposed MSCD-l_2 and MSCD-l_1 from the Bayesian perspective with the prior distributions of the coefficients identified. Section 7.5 illustrates the superior performance of MSCD to other subspace and SR-based methods; and section 7.6 gives the conclusion of this work.

7.2 Formulation of LMM-based binary hypothesis models

In the framework of LMM [5] (2.9), a test pixel $\mathbf{x}$ is modelled by a linear combination of target endmembers and background endmembers. Specifically, the LMM-based binary hypthesis models for HSI target detection are constructed as follows:

$$\begin{aligned} &H_0 : \mathbf{x} = \mathbf{M}_B\beta + \mathbf{n}_0, \ \mathbf{x} \text{ is a background pixel,} \\ &H_1 : \mathbf{x} = \mathbf{M}_T\gamma + \mathbf{M}_B\beta + \mathbf{n}_1, \ \mathbf{x} \text{ is a target pixel,} \end{aligned} \tag{7.3}$$

where $\mathbf{M}_T = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}]$ is a $p \times N_t$ matrix whose columns $\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}$ are N_t target spectra; $\mathbf{M}_B = [\mathbf{b}_1, \ldots, \mathbf{b}_{N_b}]$ is a $p \times N_b$ matrix whose columns are N_b background spectra; γ and β are the abundance vectors of $\mathbf{M}_T$ and $\mathbf{M}_B$, respectively; and $\mathbf{n}_0$ and $\mathbf{n}_1$ are assumed to be p-dimensional vectors of Gaussian white noise: $\mathbf{n}_0 \sim \mathcal{N}(\mathbf{0}, \sigma^2_{H_0}\mathbf{I}_p)$ and $\mathbf{n}_1 \sim \mathcal{N}(\mathbf{0}, \sigma^2_{H_1}\mathbf{I}_p)$, where $\mathbf{I}_p$ is the $p \times p$ identity matrix.

For a more convenient representation, we let $\mathbf{M}$ be the concatenated matrix of $\mathbf{M}_T$ and $\mathbf{M}_B$: $\mathbf{M} = [\mathbf{M}_T, \mathbf{M}_B] = [\mathbf{t}_1, \ldots, \mathbf{t}_{N_t}, \mathbf{b}_1, \ldots, \mathbf{b}_{N_b}] \in \mathbb{R}^{p \times (N_t + N_b)}$. Accordingly, we concatenate the abundance vectors γ and β of model H_1 into one vector α: $\alpha = \begin{bmatrix} \gamma \\ \beta \end{bmatrix} = [\gamma_1, \ldots, \gamma_{N_t}, \beta_1, \ldots, \beta_{N_b}]^T \in \mathbb{R}^{(N_t + N_b)}$. Then model H_1 can be rewritten as

$$\begin{aligned} H_1 : \mathbf{x} &= \mathbf{M}_T\gamma + \mathbf{M}_B\beta + \mathbf{n}_1 \\ &= \begin{bmatrix} \mathbf{M}_T & \mathbf{M}_B \end{bmatrix} \begin{bmatrix} \gamma \\ \beta \end{bmatrix} + \mathbf{n}_1 \\ &= \mathbf{M}\alpha + \mathbf{n}_1, \end{aligned} \tag{7.4}$$

and the LMM-based binary hypothesis model becomes

$$\begin{aligned} H_0 &: \mathbf{x} = \mathbf{M}_B\beta + \mathbf{n}_0,\ \mathbf{x} \text{ is a background pixel,} \\ H_1 &: \mathbf{x} = \mathbf{M}\alpha + \mathbf{n}_1,\ \mathbf{x} \text{ is a target pixel,} \end{aligned} \tag{7.5}$$

where now the unknown parameters are β, α, σ_{H_0} and σ_{H_1}.

7.2.1 Derivations of LMM-based GLR

Based on (6.2), the generalised likelihood ratio (GLR) of LMM for target detection is formulated as

$$\begin{aligned} \hat{l}(\mathbf{x}) &= \frac{l(\hat{\alpha}, \hat{\sigma}^2_{H_1}; \mathbf{x})}{l(\hat{\beta}, \hat{\sigma}^2_{H_0}; \mathbf{x})} \\ &= \left(\frac{\hat{\sigma}^2_{H_1}}{\hat{\sigma}^2_{H_0}}\right)^{-p/2} \exp\left\{ -\frac{1}{2\hat{\sigma}^2_{H_1}} \|\hat{\mathbf{n}}_1\|_2^2 + \frac{1}{2\hat{\sigma}^2_{H_0}} \|\hat{\mathbf{n}}_0\|_2^2 \right\}. \end{aligned} \tag{7.6}$$

The MLEs $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$ are equal to $\frac{1}{p}\|\hat{\mathbf{n}}_0\|_2^2$ and $\frac{1}{p}\|\hat{\mathbf{n}}_1\|_2^2$, respectively. Taking the $2/p$ power of (7.6), we have

$$\begin{aligned} L_{LMM}(\mathbf{x}) &= (\hat{l}(\mathbf{x}))^{2/p} \\ &= \left(\frac{\hat{\sigma}^2_{H_1}}{\hat{\sigma}^2_{H_0}}\right)^{-1} = \frac{\hat{\sigma}^2_{H_0}}{\hat{\sigma}^2_{H_1}} \\ &= \frac{\|\hat{\mathbf{n}}_0\|_2^2}{\|\hat{\mathbf{n}}_1\|_2^2} = \frac{\left\|\mathbf{x} - \mathbf{M}_B\hat{\beta}\right\|_2^2}{\|\mathbf{x} - \mathbf{M}\hat{\alpha}\|_2^2}. \end{aligned} \tag{7.7}$$

The MLEs of β and α in (7.7) are given by

$$\begin{aligned} \hat{\beta} &= \underset{\beta}{\operatorname{argmax}} \left\{ f_{\mathbf{x}|H_0}(\mathbf{x}; \beta, \sigma_0^2) \right\} \\ &= \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma^2_{H_0}} \|\mathbf{x} - \mathbf{M}_B\beta\|_2^2 \right\} \end{aligned} \tag{7.8}$$

and

$$\begin{aligned}\hat{\alpha} &= \underset{\alpha}{\operatorname{argmax}} \left\{ f_{\mathbf{x}|H_1}(\mathbf{x}; \alpha, \sigma_{H_1}^2) \right\} \\ &= \underset{\alpha}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma_{H_1}^2} \left\| \mathbf{x} - \mathbf{M}\alpha \right\|_2^2 \right\},\end{aligned} \tag{7.9}$$

and thus

$$\hat{\beta} = (\mathbf{M}_B^T \mathbf{M}_B)^{-1} \mathbf{M}_B^T \mathbf{x} \text{ and} \tag{7.10}$$

$$\hat{\alpha} = (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{x}, \tag{7.11}$$

by least square estimates. Based on solutions (7.10) and (7.11), the residual sums of squares (RSS) e_0 and e_1 for models H_0 and H_1 are computed as

$$\begin{aligned}H_0 : e_0 = \|\hat{\mathbf{n}}_0\|_2^2 &= \left\| \mathbf{x} - \mathbf{M}_B \hat{\beta} \right\|_2^2 \\ &= \mathbf{x}^T (\mathbf{I}_p - \mathbf{M}_B (\mathbf{M}_B^T \mathbf{M}_B)^{-1} \mathbf{M}_B^T) \mathbf{x}\end{aligned} \tag{7.12}$$

and

$$\begin{aligned}H_1 : e_1 = \|\hat{\mathbf{n}}_1\|_2^2 &= \|\mathbf{x} - \mathbf{M}\hat{\alpha}\|_2^2 \\ &= \mathbf{x}^T (\mathbf{I}_p - \mathbf{M}(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T) \mathbf{x},\end{aligned} \tag{7.13}$$

respectively. The final GLR detector of LMM is then

$$D_{LMM}(\mathbf{x}) = \frac{e_0}{e_1} = \frac{\mathbf{x}^T (\mathbf{I}_p - \mathbf{M}_B (\mathbf{M}_B^T \mathbf{M}_B)^{-1} \mathbf{M}_B^T) \mathbf{x}}{\mathbf{x}^T (\mathbf{I}_p - \mathbf{M}(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T) \mathbf{x}} \underset{H_0}{\overset{H_1}{\gtrless}} \nu. \tag{7.14}$$

The value of $D_{LMM}(\mathbf{x})$ is compared to a threshold ν to make the final decision of which hypothesis should be rejected for the test pixel $\mathbf{x}$. It is worth noting that the over-fitting problem may happened in (7.14), and to this end the matched subspace detector (MSD) [7] can be used instead. In MSD, the endmembers of background spectra and target spectra, $\mathbf{M}_B$ and $\mathbf{M}_T$, are represented by the leading eigenvectors of the background and target subspaces, respectively.

We shall note that the derivation of section 7.2.1 is different from that shown in section 6.2.2. Section 6.2.2 derives the GLR-based MSD while in this section the derivation is based on LMM. The main difference is that matrix **B** and **T** are subspaces in section 6.2.2 and the basis vectors of **B** or **T** should be orthogonal to each other; while $\mathbf{M}_B$ and $\mathbf{M}_T$ are the sets of spectra and the spectra are not necessarily orthogonal to each other in either of $\mathbf{M}_B$ or $\mathbf{M}_T$.

7.3 Matched shrunken cone detector (MSCD)

Rather than using an unconstrained LMM, it is desirable to adopt the non-negative linear model for modelling a mixed HSI pixel, so as for a reasonable physical interpretation. On top of that, we also introduce the regularisation to the non-negative representation to control the variance of estimates, and derive the whole new model from the Bayesian perspective. Particularly, we introduce the popular l_2-norm and l_1-norm regularisations to the cone-based representation. We call the proposed approach matched shrunken cone detector (MSCD) with two specific models MSCD-l_2 and MSCD-l_1.

7.3.1 Regularised cone

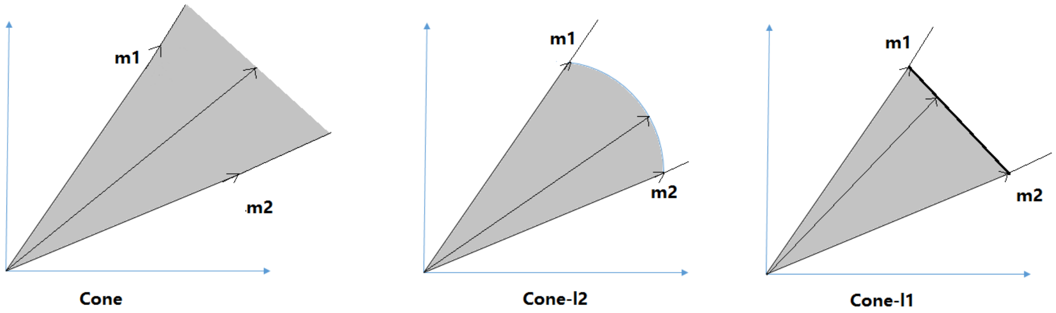


Figure 7.1: Illustration of cone-representation methods in a 2-D case with different constraints on coefficient vector **a**: (a) cone (7.15); (b) cone representation with l_2-norm regularisation (7.16); and (c) cone representation with l_1-norm regularisation (7.17).

The cone representation of a mixed pixel and its l_2-norm and l_1-norm regularised models are formulated as follows.

Cone representation:

$$\underset{\mathbf{a}\geq\mathbf{0}}{\operatorname{argmin}} ||\mathbf{x}-\mathbf{Ma}||_2^2; \tag{7.15}$$

l_2-norm regularised cone representation:

$$\underset{\mathbf{a}\geq\mathbf{0}}{\operatorname{argmin}} \|\mathbf{x}-\mathbf{M}\mathbf{a}\|_2^2+\lambda\,\|\mathbf{a}\|_2^2; \tag{7.16}$$

l_1-norm regularised cone representation:

$$\underset{\mathbf{a}\geq\mathbf{0}}{\operatorname{argmin}} \|\mathbf{x}-\mathbf{M}\mathbf{a}\|_2^2+\lambda\,\|\mathbf{a}\|_1. \tag{7.17}$$

To illustrate the relationship among (7.15), (7.16) and (7.17), we show a two-dimensional cone with different constraints in Figure 7.1. It is easily to see that the non-negative linear combination of two endmembers $\mathbf{m}_1$ and $\mathbf{m}_2$ will always lie in the cone. With additional l_2-norm or l_1-norm regularisations, the regions of the constructed vectors are down-sized to be a fan or a triangle, respectively. In other words, l_2-norm and l_1-norm regularisations shrink the value of the coefficient vector $\mathbf{a}$ for the representation of an HSI pixel.

In the following sections, we shall derive the cone-based binary hypothesis models corresponding to the optimisation problems of (7.15), (7.16) and (7.17), respectively.

7.3.2 Regularised cone-based estimators of coefficient vectors

The cone-based binary hypothesis models for target detection can be formulated as the model in (7.5) but with additional constraints. Then we call such models corresponding to (7.15), (7.16) and (7.17) matched cone detector (MCD), matched shrunken cone detector with l_2-norm regularisation (MSCD-l_2) and matched shrunken cone detector with l_1-norm regularisation (MSCD-l_1), respectively.

MCD: given the non-negative constraints (7.15), the MLEs of β and α for models H_0 and H_1 of (7.5) are given by

$$\hat{\beta}=\underset{\beta\geq\mathbf{0}}{\operatorname{argmin}}\left\{\|\mathbf{x}-\mathbf{M}_B\beta\|_2^2\right\} \text{ and} \tag{7.18}$$

$$\hat{\alpha}=\underset{\alpha\geq\mathbf{0}}{\operatorname{argmin}}\left\{\|\mathbf{x}-\mathbf{M}\alpha\|_2^2\right\}. \tag{7.19}$$

MSCD-l_2: given the l_2-norm regularised cone representation in (7.16), the estimators of β and α of (7.5) are given by

$$\hat{\beta} = \underset{\beta \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}_B \beta\|_2^2 + \lambda_0 \|\beta\|_2^2 \right\} \text{ and} \tag{7.20}$$

$$\hat{\alpha} = \underset{\alpha \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}\alpha\|_2^2 + \lambda_1 \|\alpha\|_2^2 \right\}. \tag{7.21}$$

MSCD-l_1: given the l_1-norm regularised cone representation in (7.17), the estimators of β and α of (7.5) are given by

$$\hat{\beta} = \underset{\beta \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}_B \beta\|_2^2 + \lambda_0 \|\beta\|_1 \right\} \text{ and} \tag{7.22}$$

$$\hat{\alpha} = \underset{\alpha \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}\alpha\|_2^2 + \lambda_1 \|\alpha\|_1 \right\}. \tag{7.23}$$

7.4 Bayesian Derivations of MSCD

Given the cone representation under the null hypothesis H_0 of (7.5) and Bayes' theorem

$$f(\beta|\mathbf{x}) = \frac{f(\mathbf{x}|\beta)f(\beta)}{f(\mathbf{x})}, \tag{7.24}$$

the *maximum a posteriori* (*MAP*) estimate of β is

$$\hat{\beta} = \underset{\beta}{\operatorname{argmax}} f(\beta|\mathbf{x}) = \underset{\beta}{\operatorname{argmax}} f(\mathbf{x}|\beta)f(\beta). \tag{7.25}$$

As the noise $\mathbf{n}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_{H_0}^2 \mathbf{I}_p)$, the likelihood function $f(\mathbf{x}|\beta)$ can be formulated as

$$f(\mathbf{x}|\beta) \propto \exp\left\{ -\frac{1}{2\sigma_{H_0}^2} \|\mathbf{x} - \mathbf{M}_B \beta\|_2^2 \right\}. \tag{7.26}$$

Similarly, the *MAP* estimate of α in the alternative hypothesis model H_1 is

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmax}} f(\alpha|\mathbf{x}) = \underset{\alpha}{\operatorname{argmax}} f(\mathbf{x}|\alpha)f(\alpha), \tag{7.27}$$

and as the noise $\mathbf{n}_1 \sim \mathcal{N}(\mathbf{0}, \sigma_{H_1}^2 \mathbf{I}_p)$, the likelihood function $f(\mathbf{x}|\alpha)$ can be formulated as

$$f(\mathbf{x}|\alpha) \propto \exp\left\{-\frac{1}{2\sigma_{H_1}^2}\|\mathbf{x}-\mathbf{M}\alpha\|_2^2\right\}. \tag{7.28}$$

In the ordinary cone representations (7.18) and (7.19) of the MCD model, improper uniform (non-informative) prior distributions are actually implied for parameters β and α, with $\beta \geq \mathbf{0}$ and $\alpha \geq \mathbf{0}$. However, in the proposed regularised MSCD-l_2 and MSCD-l_1, multivariate folded distributions are in fact utilised as the priors for the estimation of β in (7.20) and (7.22) and α in (7.21) and (7.23), as we shall show below.

7.4.1 Folded distributions

Suppose that the pdf of a random variable Y is $g(y)$ with $y \in \mathbb{R}$. The folding of $g(y)$ over to the non-negative line is accomplished via transform

$$X = |Y|, \tag{7.29}$$

where X is a random variable on the non-negative real line $\mathbb{R}_+ = [0,\infty)$ with pdf $f(x)$ [100]:

$$f(x) = g(x) + g(-x),\ x \in \mathbb{R}_+. \tag{7.30}$$

If we treat coefficients β_i and α_i in (7.5) as random variables, then the non-negative constraints on them imply that their pdf are on $\mathbb{R}_+$. We shall identify that a multivariate folded Gaussian distribution and a multivariate folded Laplace distribution are the prior distributions of coefficients in the proposed MSCD-l_2 and MSCD-l_1, respectively.

7.4.2 Prior distributions of β and α in MSCD-l_2

A univariate half-Gaussian distribution is defined as follows. If $Y \sim N(0,\sigma^2)$ with mean zero, then $X = |Y|$ follows a half-Gaussian distribution

$$f(x) = \frac{2}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right), x \geq 0, \tag{7.31}$$

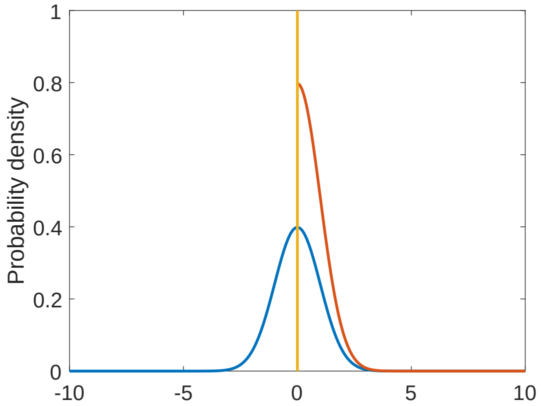


Figure 7.2: Illustration of a half-Gaussian distribution.

with mean

$$E(X) = \sqrt{2/\pi}\sigma, \tag{7.32}$$

and variance

$$var(X) = \sigma^2(1 - 2/\pi). \tag{7.33}$$

An illustration of the half-Gaussian distribution is shown in Figure 7.2. The half-Gaussian distribution is a special case of the folded version of Gaussian distribution $N(\mu, \sigma^2)$ when $\mu = 0$.

We shall identify that, if two *multivariate half-Gaussian distributions* are imposed on the coefficients α and β, respectively, as the prior distributions, then the estimators (7.20) and (7.21) of MSCD-l_2 can be derived in a Bayesian way.

In the model of the null hypothesis H_0 of the proposed MSCD-l_2, let us assume a multivariate half-Gaussian distribution as the prior for the coefficient vector β. Specifically, suppose that a vector $\mathbf{s} = [s_1, \ldots, s_{N_b}]^T$ follows a multivariate Gaussian distribution $N(\mathbf{0}, \sigma_\beta^2 \mathbf{I}_{N_b})$, where $\mathbf{I}_{N_b}$ is the $N_b \times N_b$ identity matrix, then $\beta = [\beta_1, \ldots, \beta_{N_b}]^T$ follows a multivariate half-Gaussian distribution with $\beta_i = |s_i|$ and $\beta_i \geq 0$, where $i = 1, \ldots, N_b$. The expectation of β is

$$E(\beta) = \sqrt{2/\pi}\sigma_\beta \mathbf{1}_{N_b} \in \mathbb{R}^{N_b},$$

where $\mathbf{1}_{N_b} = [1, \ldots, 1]^T$ is an N_b-dimensional vector of all ones; the covariance matrix

$$COV(\beta) = \sigma_\beta^2(1-2/\pi)\mathbf{I}_{N_b} \in \mathbb{R}^{N_b \times N_b},$$

and the pdf is

$$f(\beta) = \frac{1}{(\frac{1}{2}\pi\sigma_\beta^2)^{N_b/2}} \exp\left(-\frac{||\beta||_2^2}{2\sigma_\beta^2}\right). \tag{7.34}$$

In MSCD-l_2, placing the likelihood function $f(\mathbf{x}|\beta)$ (7.26) and the prior distribution $f(\beta)$ (7.34) into the *MAP* estimate $f(\beta|\mathbf{x})$ (7.25) and taking a logarithm, we have

$$\begin{aligned}
\hat{\beta} &= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\beta|\mathbf{x})\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\mathbf{x}|\beta)f(\beta)\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma_{H_0}^2}||\mathbf{x} - \mathbf{M}_B\beta||_2^2 - \frac{1}{2\sigma_\beta^2}||\beta||_2^2 \right\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmin}} \left\{ ||\mathbf{x} - \mathbf{M}_B\beta||_2^2 + \lambda_0 ||\beta||_2^2 \right\},
\end{aligned} \tag{7.35}$$

where $\lambda_0 = \sigma_{H_0}^2/\sigma_\beta^2$. In this way, parameter λ_0 effectively controls the degree of shrinkage via the ratio of two variances $\sigma_{H_0}^2$ and σ_β^2. Equation (7.35) is exactly the same as model (7.20).

Similarly, let us assume the prior distribution of coefficients γ of the target endmembers in the alternative hypothesis H_1 is a multivariate half-Gaussian distribution, with the expectation

$$E(\gamma) = \sqrt{2/\pi}\sigma_\gamma \mathbf{1}_{N_t} \in \mathbb{R}^{N_t},$$

where $\mathbf{1}_{N_t} = [1, \ldots, 1]^T$ is an N_t-dimensional vector; the covariance matrix

$$COV(\gamma) = \sigma_\gamma(1-2/\pi)\mathbf{I}_{N_t} \in \mathbb{R}^{N_t \times N_t},$$

where $\mathbf{I}_{N_t}$ is the $N_t \times N_t$ identity matrix; and the pdf is

$$f(\gamma) = \frac{1}{(\frac{1}{2}\pi\sigma_\gamma^2)^{N_t/2}} \exp\left(-\frac{||\gamma||_2^2}{2\sigma_\gamma^2}\right). \tag{7.36}$$

Then the concatenated α in model H_1 is actually assumed to follow a half-Gaussian distribution with mean

$$E(\alpha) = \sqrt{2/\pi}[\sigma_\gamma, \dots, \sigma_\gamma, \sigma_\beta, \dots, \sigma_\beta]^T \in \mathbb{R}^{(N_t+N_b)}. \tag{7.37}$$

Let Σ denote an $(N_t+N_b)\times(N_t+N_b)$ diagonal matrix equal to diag $\left([\sigma_\gamma, \dots, \sigma_\gamma, \sigma_\beta, \dots, \sigma_\beta]^T\right)$. Then the covariance matrix of α is

$$COV(\alpha) = (1-2/\pi)\Sigma, \tag{7.38}$$

which is an $(N_t+N_b)\times(N_t+N_b)$ matrix; and the pdf is

$$f(\alpha) = \prod_{i=1}^{N_t+N_b} \frac{2}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\alpha_i^2}{2\sigma_i^2}\right), \tag{7.39}$$

where $\sigma_i = \sigma_\gamma$ for $i = 1, \dots, N_t$ and $\sigma_i = \sigma_\beta$ for $i = N_t+1, \dots, N_t+N_b$.

When $\sigma_\gamma = \sigma_\beta$ and we let both of them be σ_α, (7.39) can be simplified to

$$f(\alpha) = \frac{1}{(\frac{1}{2}\pi\sigma_\alpha^2)^{(N_t+N_b)/2}} \exp\left(-\frac{\|\alpha\|_2^2}{2\sigma_\alpha^2}\right). \tag{7.40}$$

Then placing the likelihood function $f(\mathbf{x}|\alpha)$ (7.28) and the prior distribution (7.40) into the *MAP* estimate $f(\alpha|\mathbf{x})$ (7.27), we have

$$\begin{aligned}\hat{\alpha} &= \underset{\alpha\geq\mathbf{0}}{\operatorname{argmax}} \log\{f(\alpha|\mathbf{x})\}\\ &= \underset{\alpha\geq\mathbf{0}}{\operatorname{argmax}} \log\{f(\mathbf{x}|\alpha)f(\alpha)\}\\ &= \underset{\alpha\geq\mathbf{0}}{\operatorname{argmax}} \left\{-\frac{1}{2\sigma_{H_1}^2}\|\mathbf{x}-\mathbf{M}\alpha\|_2^2 - \frac{1}{2\sigma_{\alpha}^2}\|\alpha\|_2^2\right\}\\ &= \underset{\alpha\geq\mathbf{0}}{\operatorname{argmin}} \left\{\|\mathbf{x}-\mathbf{M}\alpha\|_2^2 + \lambda_1\|\alpha\|_2^2\right\},\end{aligned} \tag{7.41}$$

where $\lambda_1 = \sigma_{H_1}^2/\sigma_{\alpha}^2$ is the shrinkage parameter. Equation (7.41) is exactly the same as model (7.21).

We can further generalise (7.41) to a slightly-adaptive shrinkage model:

$$\hat{\alpha} = \underset{\alpha\geq\mathbf{0}}{\operatorname{argmin}} \left\{\|\mathbf{x}-\mathbf{M}\alpha\|_2^2 + \sum_{i=1}^{N_t+N_b} \lambda_{1i}\alpha_i^2\right\}. \tag{7.42}$$

In (7.42), when $i = 1,\ldots,N_t$, we have $\lambda_{1i} = \sigma_{H_1}^2/\sigma_{\gamma}^2$, and when $i = N_t+1,\ldots,N_t+N_b$, we have $\lambda_{1i} = \sigma_{H_1}^2/\sigma_{\beta}^2$.

7.4.3 Prior distributions of β and α in MSCD-l_1

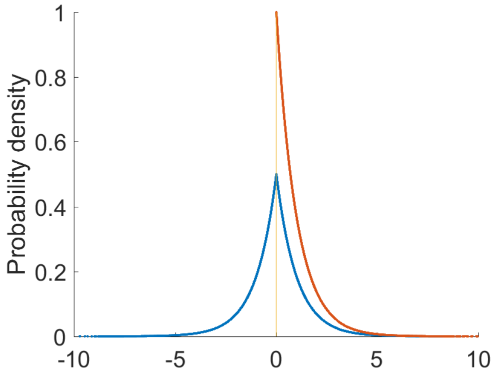


Figure 7.3: Illustration of a half-Laplace distribution.

A Laplace distribution is defined as follows. If a random variable Y has a

Laplace distribution $\mathscr{L}(\mu, b)$, then it has mean μ, variance $2b^2$, and pdf

$$g(y) = \frac{1}{2b}\exp\left(-\frac{|y-\mu|}{b}\right),\ y \in \mathbb{R}. \tag{7.43}$$

A folded Laplace distribution is also accomplished via transform (7.29), and the pdf of the transformed random variable X becomes (7.30). Placing (7.43) in (7.30), we have the pdf of a folded Laplace distribution [100]:

$$f(x) = \frac{1}{b}\begin{cases} \exp(-\frac{\mu}{b})\cosh(\frac{x}{b}) \text{ for } 0 \leqslant x < \mu, \\ \exp(-\frac{x}{b})\cosh(\frac{\mu}{b}) \text{ for } \mu \leqslant x. \end{cases} \tag{7.44}$$

Specifically, when $\mu = 0$, (7.44) reduces to

$$f(x) = \frac{1}{b}\exp\left(-\frac{x}{b}\right),\ x \in \mathbb{R}_+, \tag{7.45}$$

which is the pdf of a half-Laplace distribution with mean b.

We shall also identify that, if two *multivariate half-Laplace distributions* are imposed on the coefficients α and β, respectively, as the prior distributions, then the estimators (7.22) and (7.23) of MSCD-l_1 can be derived in a Bayesian way.

Let a random multivariate vector $\mathbf{v} = [v_1, \dots, v_{N_b}]^T$ have a multivariate Laplace distribution $\mathscr{L}(\mathbf{0}, \varphi_\beta \mathbf{I}_{N_b})$. For model (7.22), coefficient vector $\beta = [\beta_1, \dots, \beta_{N_b}]^T$ follows a multivariate half-Laplace distribution if $\beta_i = |v_i|$ for $i = 1, \dots, N_b$. In this case, the mean of β is $E(\beta) = \varphi_\beta \mathbf{1}_{N_b}$ and the pdf is

$$f(\beta) = \frac{1}{\varphi_\beta^{N_b}}\prod_{i=1}^{N_b}\exp\left(-\frac{\beta_i}{\varphi_\beta}\right),\ \text{for } \beta_i \geq 0. \tag{7.46}$$

Then placing the likelihood function $f(\mathbf{x}|\beta)$ (7.26) and the prior distribution

$f(\beta)$ (7.46) into the *MAP* function $f(\beta|\mathbf{x})$ (7.25) and taking the logarithm, we have

$$\begin{aligned}
\hat{\beta} &= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\beta|\mathbf{x})\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\mathbf{x}|\beta)f(\beta)\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma_{H_0}^2}\|\mathbf{x}-\mathbf{M}_B\beta\|_2^2 - \frac{1}{\varphi_\beta}\sum_{i=1}^{N_b}\beta_i \right\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma_{H_0}^2}\|\mathbf{x}-\mathbf{M}_B\beta\|_2^2 - \frac{1}{\varphi_\beta}\|\beta\|_1 \right\} \\
&= \underset{\beta \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x}-\mathbf{M}_B\beta\|_2^2 + \lambda_0\|\beta\|_1 \right\},
\end{aligned} \tag{7.47}$$

where $\lambda_0 = 2\sigma_{H_0}^2/\varphi_\beta$ controls the degree of shrinkage through the ratio of $2\sigma_{H_0}^2$ and φ_β. Equation (7.47) is exactly the same as model (7.23).

In the same fashion, the prior distribution of coefficients γ of the target endmembers in the alternative model H_1 is also assumed to be a multivariate half-Laplace distribution with pdf

$$f(\gamma) = \frac{1}{\varphi_\gamma^{N_t}}\prod_{i=1}^{N_t}\exp\left(-\frac{\gamma_i}{\varphi_\gamma}\right), \text{ for } \gamma_i \geq 0. \tag{7.48}$$

As a result, the concatenated coefficients α in model H_1 is in fact assumed to follow a multivariate half-Laplace distribution as well, with pdf

$$f(\alpha) = \prod_{i=1}^{N_t+N_b}\frac{1}{\varphi_i}\exp\left(-\frac{\alpha_i}{\varphi_i}\right), \text{ for } \alpha_i \geq 0, \tag{7.49}$$

where $\varphi_i = \varphi_\gamma$ for $i = 1,\ldots,N_t$ and $\varphi_i = \varphi_\beta$ for $i = N_t+1,\ldots,N_t+N_b$.

As with the derivations in section 7.4.2, when we have $\varphi_\gamma = \varphi_\beta$ and let both of them to be φ_α, (7.49) can be rewritten as

$$f(\alpha) = \frac{1}{\varphi_\alpha^{N_t+N_b}}\exp\left(-\frac{||\alpha||_1}{\varphi_\alpha}\right), \text{ for } \alpha_i \geq 0. \tag{7.50}$$

Then placing the likelihood function $f(\mathbf{x}|\alpha)$ (7.28) and the prior distribution

$f(\alpha)$ (7.50) into the *MAP* estimate of α (7.27) and taking the logarithm, we have

$$\begin{aligned}\hat{\alpha} &= \underset{\alpha \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\alpha|\mathbf{x})\} \\ &= \underset{\alpha \geq \mathbf{0}}{\operatorname{argmax}} \log\{f(\mathbf{x}|\alpha) f(\alpha)\} \\ &= \underset{\alpha \geq \mathbf{0}}{\operatorname{argmax}} \left\{ -\frac{1}{2\sigma_{H_1}^2} \|\mathbf{x} - \mathbf{M}\alpha\|_2^2 - \frac{1}{\varphi_\alpha} \|\alpha\|_1 \right\} \\ &= \underset{\alpha \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}\alpha\|_2^2 + \lambda_1 \|\alpha\|_1 \right\},\end{aligned} \tag{7.51}$$

where λ_1 is a shrinkage parameter equal to $2\sigma_{H_1}^2/\varphi_\alpha$. Equation (7.51) is exactly the same as model (7.23).

Again, (7.51) can be generalised as

$$\hat{\alpha} = \underset{\alpha \geq \mathbf{0}}{\operatorname{argmin}} \left\{ \|\mathbf{x} - \mathbf{M}\alpha\|_2^2 + \sum_{i=1}^{N_t+N_b} \lambda_{1i}\alpha_i \right\}, \tag{7.52}$$

where $\lambda_{1i} = 2\sigma_{H_1}^2/\varphi_\gamma$ for $i = 1, \ldots, N_t$ and $\lambda_{1i} = 2\sigma_{H_1}^2/\varphi_\beta$ for $i = N_t + 1, \ldots, N_t + N_b$.

It is worth noting that there is often only one target spectrum available in practice for HSI target detection. In such case, the target training sample $\mathbf{M}_T$ is a $p \times 1$ single vector instead of a $p \times N_t$ matrix. Then the variance σ_γ defined in MSCD-l_2 and the diversity φ_γ in MSCD-l_1 are both have to be set as ∞, since there is no σ_γ and ϕ_γ can be estimated from the target samples. In other words, we actually do not shrink the coefficient $\gamma \in \mathbb{R}$ for the target subset $\mathbf{M}_T$ so long as $N_t = 1$, and let non-negative projection of a test HSI pixel $\mathbf{x}$ onto the target endmember to be as much as possible.

7.4.4 Regularisation and prior distributions of MSCD

To adjust (and often improve) the performance of a statistical model like MSD or MCD, some prior domain knowledge about the model, particularly the coefficients, can be incorporated by imposing regularisation (a frequentist fashion) or assuming the prior distributions (a Bayesian fashion). These two ways, although from different statistical schools of thinking and inference, can often achieve the same effect, in particular if we can find the pair of a regularisation term and a prior distribution. That is, deriving the corresponding prior distribution to a regularisation term can not only provide a theoretical justification of the latter, but also assist a deeper understanding of the latter; and vice versa. This inspires our derivation of MSCD from the Bayesian perspective.

Specifically, the benefit from proposing MSCD-l_2 and MSCD-l_1 can be understood from both regularisation and Bayesian points of view.

In MSCD-l_2, an l_2-norm regularisation term is added to impose constraints on the combination coefficients in the model of MCD. This will shrink the value of the coefficients and thus reduce the variances of the estimated coefficients, as usually achieved by a shrinkage methods [9]. From the Bayesian perspective, as the coefficients are non-negative, such an l_2-norm regularisation can be derived as corresponding to a multivariate half-Gaussian prior distribution for the coefficients, as we have shown in section 7.4.2. Equivalently, using such a prior will reduce the posterior variances of the coefficients, in a Bayesian sense. On the one hand, the original MCD models (7.18) and (7.19) are equivalent to (7.35) and (7.41) when λ_0 and λ_1 are zeros, which implies the use of prior distributions of infinite prior variance. In contrast, the non-zero shrinkage parameters λ_0 and λ_1 in (7.35) and (7.41) imply a finite prior variances for the coefficients. On the other hand, with such a prior, the posterior variance of a coefficient will be smaller than the variance of the estimator inferred from the likelihood only. Provided with the lower variance, MSCD-l_2 can provide more stable classification performance than MCD.

The case of MSCD-l_1 is similar to MSCD-l_2, in terms of shrinkage, though the l_1-norm regularisation on the coefficients of the cone representation-based MCD implies a multivariate half-Laplace prior distribution for the coefficients, as we have shown in section 7.4.3. In fact, as well known, l_1-norm regularisation (like lasso) or a Laplace prior distribution can induce not only shrinkage of the values of the coefficients, but also zero values of some coefficients, i.e. the sparsity of the coefficient vectors. This actually implies an endmember selection in the cone representation for HSI target detection.

7.5 Experimental studies

We conduct target detection experiments on two real hyperspectral datasets for sub-pixel target detection and full-pixel target detection, respectively. For sub-pixel target detection, a target appearing in an HSI is smaller than an HSI pixel. In this case we compare the target detection methods on the Hymap dataset [3] which has been used in Chapter 5 and Chapter 6. For full-pixel target detection, a target appearing in an HSI can occupy more than one HSI pixel. We use the dataset collected by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) from San Diego, CA, USA to evaluate the performance of detecting the full-pixel targets.

We compare the proposed methods MSCD-l_2 and MSCD-l_1 with three types of target detectors: 1) the cone representation-based detector MCD in (7.18) and (7.19); 2) the subspace detectors OSP (2.12) [18] and MSD (2.10) [7]; and 3) the sparse representation-based detectors STD (2.16) [23] and SRBBH (2.20) [24]. For the proposed MSCD-l_2 and MSCD-l_1, we adopt the MATLAB codes provided by [97] and on `http://www.yelab.net/software/SLEP/` to solve the l_2-norm regularised cone model (7.16) and the l_1-norm regularised cone model (7.17), respectively.

As with [15, 23, 24, 83] and our work shown in section 6.6 of Chapter 6, we adopt the dual window scheme to obtain the background pixels for each test HSI pixel in a local approach. An illustration of a dual window is shown in Figure 6.4, which separates a local area of a test HSI pixel into two regions: an inner window region (IWR) and an outer window region (OWR). Also as with [24, 93, 94], we empirically set OWR and IWR to be 15×15 and 9×9 respectively for all compared methods, in order to detect targets appearing in both of Hymap and AVIRIS datasets.

7.5.1 The Hymap dataset

7.5.1.1 Data description

The descriptions of the Hymap dataset have been detailed in section 5.4.2 of Chapter 5. In this experiment, we adopt the same data settings in section 5.4.2 for evaluating the proposed MSCD. The data settings are summarised as follows:

- only one target pixel of each desired target is assumed to be in the HSI;
- the seven types of targets and their central coordinates of ROIs are shown in Table 5.3;
- a spatial size of 100×300 sub-image is cropped for evaluation as shown in Figure 5.9(a);
- the ROIs of seven types of target are shown in Figure 5.9(b);
- the spectrum of each desired target (F1-F4 and V1-V3) are plotted in Figure 5.10(a) and Figure 5.10(b), respectively; and the sampled spectral signature with each ROI of target in the scene are shown in Figure 5.11(a) and Figure 5.11(b), respectively.

7.5.1.2 Experimental settings

The ROIs mean that a target pixel may appear in any coordinates within the ROIs, and the exact number of pixels of a type of target is unknown. As with the experimental settings in [2, 84], the criterion for measuring the correct detection is that if at least one pixel in the ROIs is identified as target, then this detection is regarded as a correction detection. Moreover, since the predefined threshold of each compared detector is unknown, we also adopt the *false alarm rate (FAR)* defined in [2, 84] for measuring the detection performance. The FAR is equal to the number of pixels that are not in the target ROIs but have the test values equal to or greater than the highest test value of pixels within the ROIs, over the total number of pixels in the Hymap HSI, i.e. 30,000 in the example of Figure 5.9. Hence we expect to see the lower the FAR, the better the detection performance.

Parameters of the compared methods should be determined. For the subspace methods OSP and MSD, parameter r_b, which is the number of leading eigenvectors of background subspace, should be determined. For the sparse representation methods STD and SRBBH, parameter L, which is the sparsity level, should be determined. We shall also determine the parameter λ_0 and λ_1, which are the shrinkage parameters of models H_0 and H_1, respectively, for both the proposed MSCD-l_1

Table 7.1: Parameter settings: the number r_b of leading eigenvectors of OSP and MSD; and the sparsity level L of STD and SRBBH.

Target	r_b		L	
	OSP	MSD	STD	SRBBH
F1	104	116	10	7
F2	119	76	10	10
F3	117	117	4	7
F4	119	116	8	5
V1	119	1	10	4
V2	119	8	4	4
V3	119	7	5	7

Table 7.2: Parameter settings: λ_0 and λ_1 of MSCD-l_1 and MSCD-l_2.

Target	MSCD-l_1		MSCD-l_2	
	λ_0	λ_1	λ_0	λ_1
F1	1e-04	1e-04	1e-03	1e-01
F2	1e-04	1e-04	1e-02	1e-01
F3	1e-01	1e-03	1e-02	1e-02
F4	1e-00	1e-01	1e-00	1e-00
V1	1e-03	1e-04	1e-02	1e-00
V2	1e-03	1e-04	1e-02	1e-00
V3	1e-01	1e-01	1e+01	1e-00

and MSCD-l_2. Due to the limited size of training samples, we are unable to do cross-validation for tuning parameters. Specifically, we have only one ground-truth spectrum of each type target and we do not even have the ground-truth spectra of background samples within the Hymap HSI. Therefore for illustration purposes, we manually tune the parameters of each compared method to their optimal values when the FARs of each method are the lowest, as done by most published works on the Hymap dataset [2, 81, 82]. The range of r_b is [1, 119]; the range of L is [1, 30]. For the proposed MSCD-l_1 and MSCD-l_2, we also manually tune the parameters λ_0 and λ_1 to their optimal values by sweeping the value in [1e-05, 1e-04, 1e-03, 1e-02, 1e-01, 1e-00, 1e+10, 1e+02]. The optimal values of r_b for OSP and MSD and of L for STD and SRBBH are listed in Table 7.1. The optimal values of λ_0 and λ_1 for the proposed MSCD-l_1 and MSCD-l_2 are listed in Table 7.2.

Table 7.3: False alarm rate (FAR) of compared methods for the Hymap dataset. The OWR and IWR are set to be 15×15 and 9×9, respectively, for OSP, MSD, STD, SRBBH, MSCD, MSCD-l_1 and MSCD-l_2. The minimum FARs are in boldface.

Target	Subspace representation		Sparse representation		Cone representation		
	OSP	MSD	STD	SRBBH	MCD	MSCD-l_1	MSCD-l_2
F1	5.64e-02	1.64e-02	0.04e-02	**0.00e-02**	**0.00e-02**	**0.00e-02**	**0.00e-02**
F2	0.56e-02	**0.01e-02**	0.30e-02	0.05e-02	0.24e-02	0.31e-02	0.19e-02
F3	2.53e-02	**0.00e-02**	1.84e-02	0.34e-02	0.79e-02	**0.00e-02**	0.02e-02
F4	0.13e-02	0.26e-02	0.32e-02	0.11e-02	0.45e-02	0.35e-02	**0.04e-02**
F1-F4	8.86e-02	1.91e-02	2.5e-02	0.5e-02	1.48e-02	0.66e-02	**0.25e-02**
V1	0.52e-02	1.17e-02	0.89e-02	0.36e-02	0.04e-04	0.03e-02	**0.02e-02**
V2	3.50e-02	6.73e-02	**1.14e-02**	2.04e-02	5.18e-02	5.22e-02	3.00e-02
V3	7.74e-02	2.75e-02	**0.66e-02**	9.56e-02	21.9e-02	7.26e-02	1.85e-02
V1-V3	11.76e-02	10.65e-02	**2.69e-02**	11.95e-02	27.12-02	12.51e-02	4.87e-02
Sum	20.63e-02	12.56e-02	5.19e-02	12.46e-02	28.60e-02	13.17e-02	**5.12e-02**

7.5.1.3 Experimental results and analysis

The FARs of all compared methods for detecting each type of targets are listed in Table 7.3. Firstly, for the cone-based detectors, MCD, MSCD-l_2 and MSCD-l_1, we can observe that the proposed MSCD-l_2 (FAR 5.12e-02) and MSCD-l_1 (13.17e-02) outperform MCD (28.60e-02) for detecting different types of targets. This illustrates the effectiveness of incorporating the regularisations into the optimisation of non-negative problems. Furthermore, MSCD-l_2 performs significantly better than MSCD-l_1, which implies that the l_2-norm regularised cone representation is more effective than the l_1-norm regularised cone representation for detecting the targets in the Hymap dataset.

Secondly, comparing all the methods listed in Table 7.3, we can clearly see that our proposed MSCD-l_2 outperforms OSP, MSD, STD, SRBBH, MCD and MSCD-l_1 for detecting targets F1, F4 and V1, and it performs the best in terms of the sum of FARs of detecting fabric targets F1-F4 with FAR as 0.25e-02. More importantly, MSCD-l_2 also outperforms others in detecting all types of targets, i.e. F1-F4 and V1-V3, with the smallest sum of FARs as 5.12e-02. This indicates that the proposed MSCD-l_2 is more effective than the subspace and sparse representation methods.

Last but not least, we shall note that, among the compared methods, the subspace method MSD and the sparse representation method STD perform relatively better than each of their cohort methods, i.e. MSD is better than OSP and STD is better than SRBBH in terms of the sum of FARs of all targets. STD also has competitive performance for detecting the vehicle targets, particularly V2 and V3. However, both of MSD and STD are not as good as the proposed MSCD-l_2 in terms of the sum of FARs for detecting all targets. This also implies that MSCD-l_2 is more stable than other methods, whatever the types of targets and the sizes of them.

To further illustrate the detection performances of the compared methods, we display the prediction maps of all methods in Figure 7.4 for detecting target F4. Figure 7.4(b) shows the ground-truth map of target F4. The value of each pixel shown in Figure 7.4(c)-7.4(i) represents the test statistic value of the pixel: the brighter the pixel, the higher the test statistic value, and thus the more likely a target. That is, we expect a good prediction map to show a clear pattern for detecting F4 that the brightnesses of the pixels located within the ROIs of F4 are higher than those outside. From these prediction maps, we can visually observe that 1) OSP (Figure 7.4(c)) and MSD (Figure 7.4(d)) have no such a clear pattern; 2) STD (Figure 7.4(e)), SRBBH (Figure 7.4(f)), MCD (Figure 7.4(g)) and MSCD-l_1(Figure 7.4(h)) look better, but we can easily spot many outside pixels brighter than the pixels within the ROIs of F4; 3) among all the maps, MSCD-l_2 in Figure 7.4(i) looks the best, though it still does not provide a zero FAR (FAR = 0.04e-02 in Table 7.3), where the bright pixels largely stick around the ground-truth of F4, rather than spread over the scene as in other prediction maps.

7.5.2 The AVIRIS dataset

7.5.2.1 Data description

The AVIRIS data was captured at an airport in the San Diego, CA, USA with the planes as targets. We select an sub-image that spatially covers a region of 100×100. As with [24, 94], we remove some bad spectral bands and preserve 189 spectral bands for evaluation. In the AVIRIS scene, there are three planes need to be detected, consisting of 58 HSI pixels that are labelled as target pixels. The hyperspectral image scene and the ground-truth maps are shown in Figure 7.5(a) and

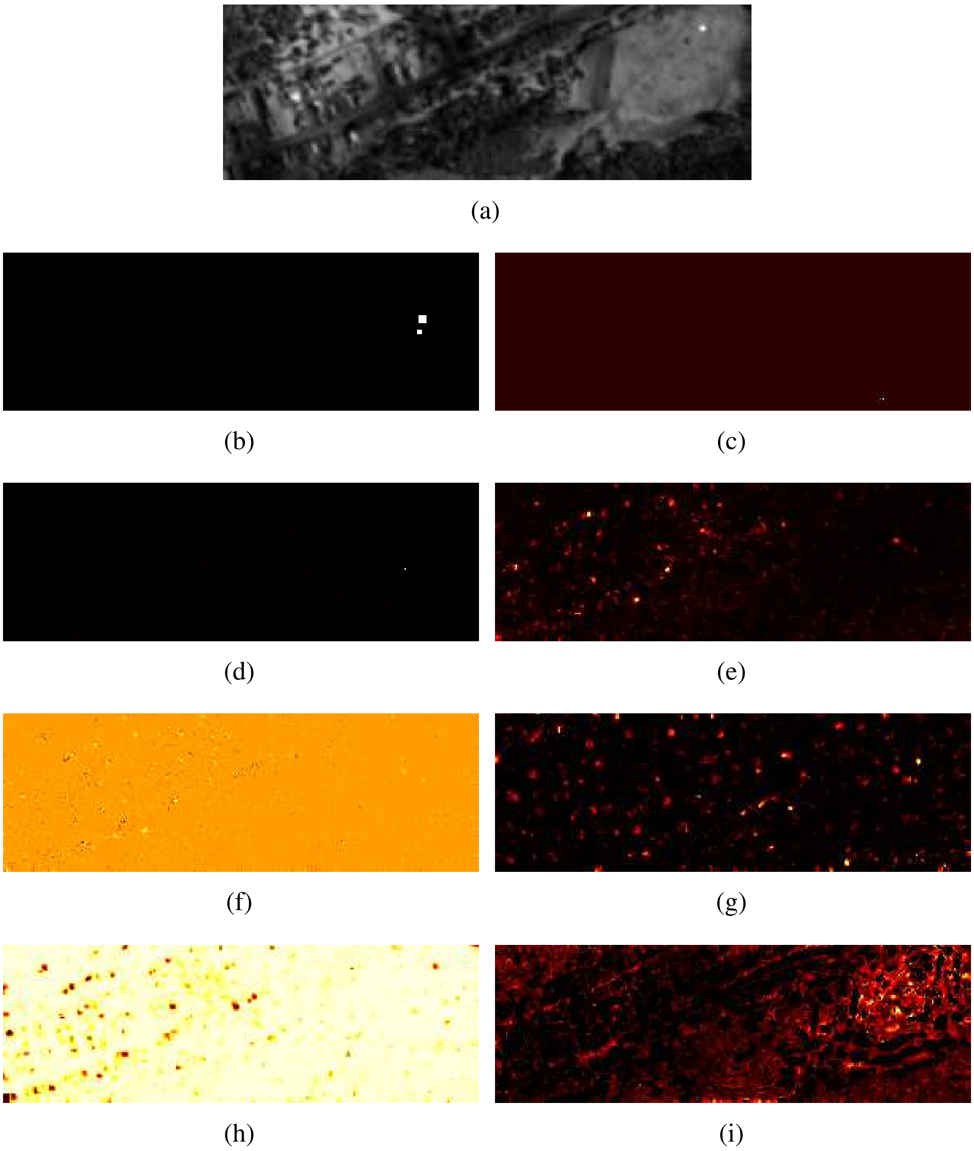


Figure 7.4: Prediction maps of test statistics for detecting F4 in the Hymap image. (a) The Hymap HSI of the 33rd spectral band; (b) ground-truth labels of F4; (c) OSP, FAR = 0.13e-02; (d) MSD, FAR = 0.26e-02; (e) STD, FAR = 0.32e-02; (f) SRBBH, FAR = 0.11e-02; (g) MCD, FAR = 0.45e-02; (h) MSCD-l_1, FAR = 0.35e-02; (i) MSCD-l_2, FAR = 0.04e-02.

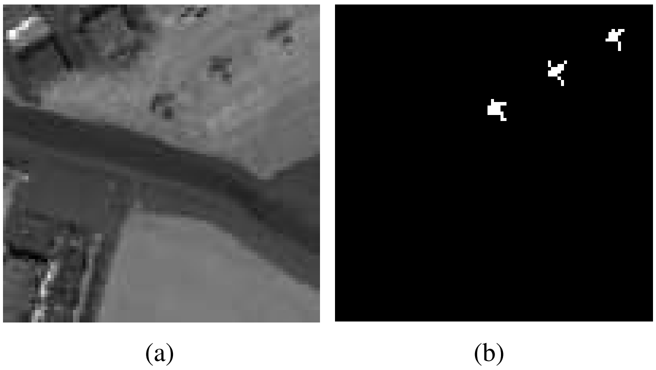


Figure 7.5: (a) The AVIRIS sub-image (100 × 100) of the 45th spectral band; (b) the ground-truth labels of targets including 58 target pixels.

Figure 7.5(b), respectively. It is clear that each target plane covers more than one HSI pixel. Hence the AVIRIS dataset adopted here is suitable for evaluating the full-pixel target detection performance of the compared methods.

7.5.2.2 Experimental settings

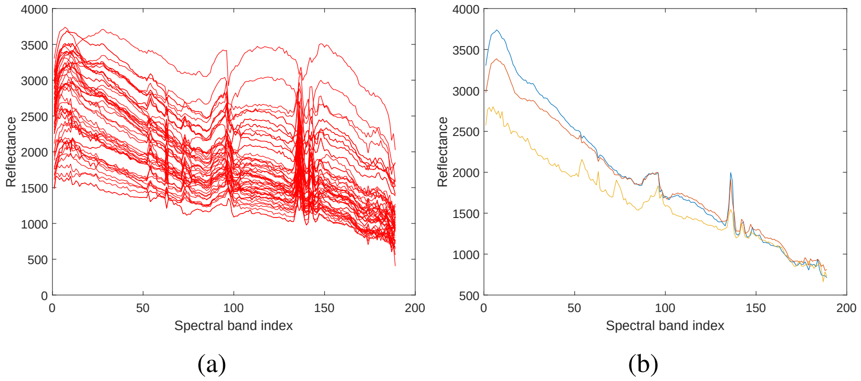


Figure 7.6: Spectra of targets in the AVIRIS dataset: (a) all target spectra in the hyperspectral scene; (b) spectra of three training target pixels, which are the central pixels of the three planes, respectively.

Because the labels for individual HSI pixels are available in the AVIRIS dataset, we select the three central HSI pixels of each plane as the prior spectra of target signatures, as with [24] and [94]. The rest of target HSI pixels are used to evaluate the detection performances of methods. The 58 target spectra and the three training target spectra are shown in Figure 7.6(a) and Figure 7.6(b), respectively. We can observe that the spectra of the target HSI pixels still look different from each other. However, compared with Figure 5.10 and Figure 5.11 for the Hymap dataset, the spectral pattern of the AVIRIS targets may be clearer and the targets may be easier to be detected, as the training target pixels are from the HSI rather than from spectral libraries.

As with [24] and [94], we use the *receiver operating characteristic (ROC)* curves to measure the detection performances for the AVIRIS dataset. The reason of using ROC instead of FAR is that now we have the labelling information for every single target HSI pixel, instead of the only available ROIs in the Hymap dataset. We expect that an ROC curve goes to the top left of the plot, if the detection performance of a method is good. Additionally, we adopt the area under curve (AUC) statistics to quantitatively measure the detection performance in pair with the ROC curves.

Table 7.4: Parameters and AUC statistics of the compared methods for the AVIRIS dataset. The OWR and IWR are set to be 15 × 15 and 9 × 9, respectively for OSP, MSD, STD, SRBBH, MSCD, MSCD-l_1 and MSCD-l_2. The maximal AUC is in boldface.

	Subspace representation		Sparse representation		Cone representation				
Detector	OSP	MSD	STD	SRBBH	MCD	MSCD-l_1		MSCD-l_2	
Parameter	r_b	r_b	L	L	NA	λ_0	λ_1	λ_0	λ_1
	157	7	9	11	NA	1e-03	1e-02	1e-04	1e-02
AUC	0.9527	0.9091	0.9647	0.9547	0.9616	**0.9713**		0.9632	

Similarly, the parameters of each compare method should be determined: the number of leading eigenvectors r_b for the subspace methods OSP and MSD; the sparsity level L for the SR-based methods; and the shrinkage parameters λ_0 and λ_1 for both of the proposed MSCD-l_1 and MSCD-l_2. Again, for illustration purposes, the parameters are empirically determined and the values are listed in Table 7.4, with the same tuning ranges of values as for the Hymap dataset.

7.5.2.3 Experimental results and analysis

The ROC curves of all the compared methods are shown in Figure 7.7 and the corresponding AUC statistics are listed in Table 7.4. Once again, we can observe that the proposed MSCD-l_1 and MSCD-l_2 both outperform MCD, which indicates the benefit of incorporating the l_1-norm and l_2-norm regularisations into the cone-based

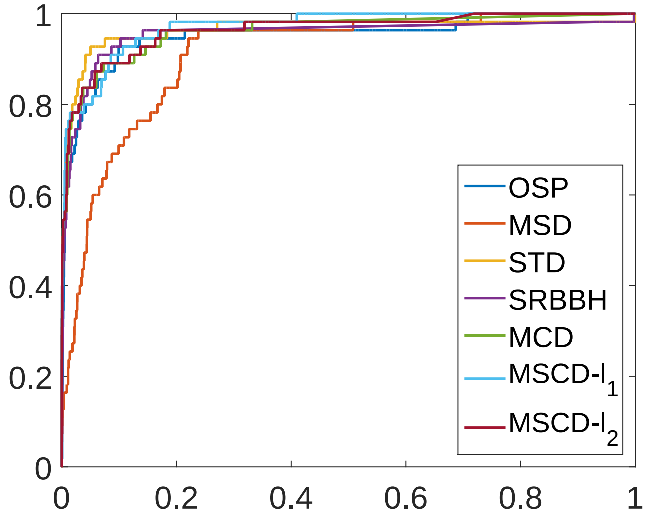


Figure 7.7: The ROC curves of the compared methods: OSP, MSD, STD, SRBBH, MCD, MSCD-l_1 and MSCD-l_2.

representation for HSI target detection. Moreover, the proposed MSCD-l_1 has the best performance among all the compared method. This implies that, for detecting full-size target HSI pixels, introducing the sparsity constraints on the coefficients into the MCD can achieve better performance than the l_2-norm constraints on the coefficients. Generally speaking, the cone representation methods are better than the sparse representation methods; and the sparse representation methods are better than the subspace methods for detecting full-size target HSI pixels in the AVIRIS dataset.

We also plot the prediction maps for all the methods and display them in Figure 7.8. It can be seen that the cone representation methods, i.e. MCD (Figure 7.8(f)), MCD-l_1 (Figure 7.8(g)) and MCD-l_2 (Figure 7.8(h)), look relatively better than the others. The difference among these three prediction maps are not so much. Among the other four methods (OSP, MSD, STD and SRBBH), MSD (Figure 7.8(c)) looks the worst, as it is badly affected by the dual window scheme (Figure 6.4); and STD looks better than OSP, MSD and SRBBH. However, the colour contrast in Figure 7.8(d) of STD is not as large as those in Figure 7.8(f)-7.8(h) of the cone-representation methods. This means that the test statistics of background pixels and target pixels of MCD, MSCD-l_1 and MSCD-l_2 are more different than those of STD, which further illustrates the stable performances of the cone-based methods for detecting targets in the AVIRIS dataset.

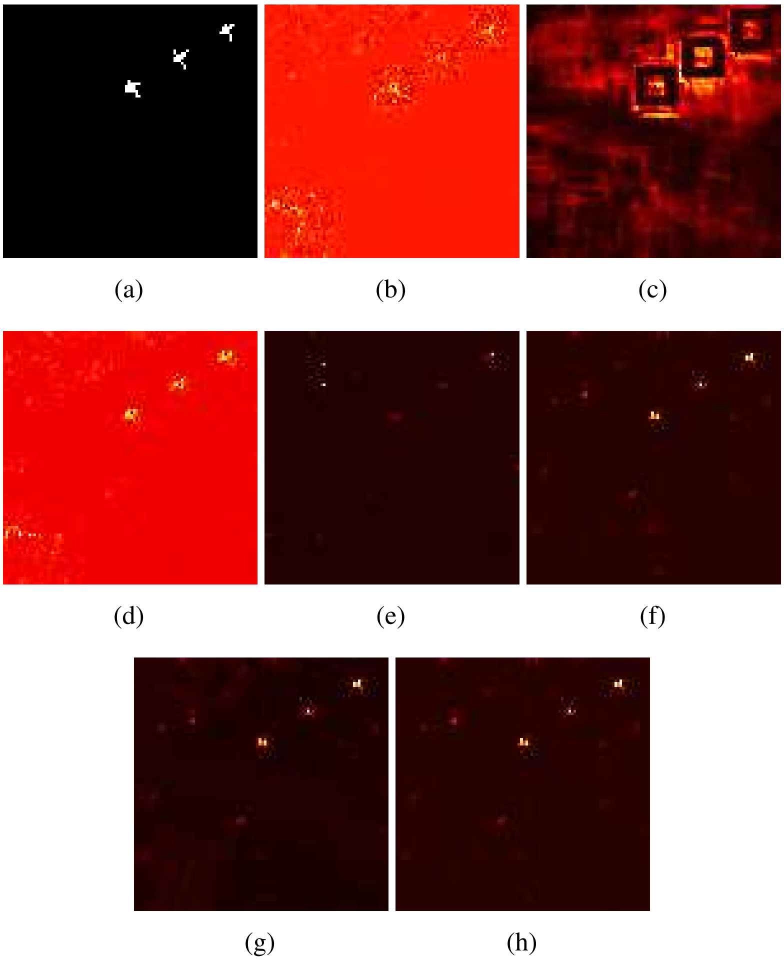


Figure 7.8: Prediction maps for detecting planes in the AVIRIS image. The brighter the pixels, the more likely to be targets. (a) Ground-truth labels of targets; (b) OSP, AUC = 0.9527; (c) MSD, AUC = 0.9091; (d) STD, AUC = 0.9647; (e) SRBBH, AUC = 0.9547; (f) MCD, AUC = 0.9616; (g) MSCD-l_1, AUC = 0.9713; (h) MSCD-l_2, AUC = 0.9632.

7.6 Conclusion

In this chapter, we have proposed a new approach called matched shrunken cone detector (MSCD) for hyperspectral target detection. Two working models of MSCD, namely MSCD-l_2 and MSCD-l_1, have also been proposed, with the l_2-norm and l_1-norm regularisations incorporated into the MSCD, respectively. Geometrically, we have analysed the underlying effectiveness of MSCD. The values of the coefficients are shrunken within a cone either by the l_2-norm regularisation or the l_1-norm regularisation, which form two different constrained regions for the coefficients. Statistically, we have derived MSCD from the Bayesian perspective. We have shown that if a multivariate half-Gaussian distribution or a multivariate half-Laplace distribution is assumed as the prior distribution of the coefficients, then MSCD-l_2 or MSCD-l_1 can be derived. In our experiments, cases studies on two real hyperspectral datasets have been conducted, with the Hymap dataset to illustrate the sub-pixel target detection and the AVIRIS dataset to illustrate the full-pixel target detection. We have compared three categories detectors including the subspace methods, the sparse representation methods and the cone representation methods. Experimental results on both of the two datasets have showed the superior performance of the proposed MSCD.

We would like to make two further notes about the Bayesian derivations. One the one hand, in the Bayesian paradigm, the half-Gaussian or half-Laplace prior distribution can be assumed on the basis of our prior knowledge that the model coefficients are positive. In principle any distribution of a positive random variable can be assumed as the prior for such a coefficient; in our case, half-Gaussian and half-Laplace distributions match the l_2-norm and l_1-norm regularisations, respectively. That is, the half-Gaussian and half-Laplace priors provide us with a principled Bayesian interpretation of the two regularised models. On the other hand, if the practitioners hold some specific prior domain knowledge which prefers to be modelled by other positive prior distributions, such as log-normal distributions or gamma distributions, a Bayesian derivation like ours can open a door to different new regularised models, which fit their practice better. This can be an interesting and practically valuable direction to further our principled work presented in this chapter.

Chapter 8

Conclusions and Future Work

In this thesis, we have discussed on the problems of HSI classification and HSI target detection. Five new methods covered in Chapter 3-7 have been proposed to solve the corresponding problems. For the HSI classification, we focus on the joint sparse model (JSM), and have proposed a dictionary learning method called JSM-based discriminative K-SVD (JSM-DKSVD) (Chapter 3) and a cone-based JSM, called C-JSM (Chapter 4), respectively. For the HSI target detection which is a special case of HSI classification, we have developed three new works based on the linear mixing model (LMM) from the statistical point of view. Specifically, we first tackle the mixing problem in the target detection, and have proposed a method called matched subspace detector with interaction effects (MSDinter) (Chapter 5). Secondly, to solve the problems of high variances of coefficients estimations, we have proposed two new methods called the matched shrunken subspace detector (MSSD) (Chapter 6) and the matched shrunken cone detector (MSCD) (Chapter 7), respectively to shrink the coefficients in the linear model by imposing constraints. Equally important, we have derived both of MSSD and MSCD from the Bayesian perspective and have showed that the certain prior distributions are in fact assumed for the coefficients.

8.1 Relation between chapters

The five works in Chapter 3-7 can be categorised into three groups according to general concepts: JSM, LMM and cone-representation-based methods. The relation between chapters has been briefly illustrated in Figure. 1.1.

8.1.1 JSM-based approaches (Chapter 3 and Chapter 4)

The two works proposed for HSI classification are based on the JSM. Typical classification problem consists of two phases: the training phase and the test phase. The proposed JSM-DKSVD focuses on the training phase, i.e. to train a powerful dictionary for classification. The proposed C-JSM focuses on the test phase, i.e. modelling a test HSI pixel so long as a pre-defined dictionary is given. On the one hand, the proposed C-JSM shares the same application of the JSM, i.e. to model an HSI pixel. On the other hand, C-JSM can also be incorporated in the dictionary learning process, so as to learn a dictionary with additionally rich spatial information.

8.1.2 LMM-based approaches (Chapter 5, Chapter 6 and Chapter 7)

The three works proposed for HSI target detection are based on the LMM. Let $\mathbf{E}_B$ and $\mathbf{E}_T$ be the sets of background endmembers and target endmembers, respectively. Then no matter we use spectral signatures or eigenvectors to represent the endmembers, all three works can be regarded as the developments of the following model:

$$\begin{aligned} H_0 &: \mathbf{x} = \mathbf{E}_B \beta + \mathbf{n}_0, \\ H_1 &: \mathbf{x} = \mathbf{E}_T \gamma + \mathbf{E}_B \beta + \mathbf{n}_1. \end{aligned} \tag{8.1}$$

Thus the proposed MSDinter (Chapter 5), MSSD (Chapter 6) and MSCD (Chapter 7) in fact make different contributions to (8.1) with respect to different factors in (8.1):

- **MSDinter**: Interaction effects

$$\begin{aligned} H_0 &: \mathbf{x} = \mathbf{E}_B \beta + \mathbf{n}_0, \\ H_1 &: \mathbf{x} = \mathbf{E}_T \gamma + \mathbf{E}_B \beta + \mathbf{H}\eta + \mathbf{n}_1. \end{aligned}$$

- **MSSD**: l_2-norm regularisation on coefficient vectors β and γ

$$
\begin{aligned}
H_0 &: \mathbf{x} = \mathbf{E}_B\beta + \mathbf{n}_0, \\
H_1 &: \mathbf{x} = \mathbf{E}_T\gamma + \mathbf{E}_B\beta + \mathbf{n}_1.
\end{aligned}
$$

- **MSCD**: Non-negativity and l_1, l_2-norm regularisation constraints on coefficient vectors β and γ

$$
\begin{aligned}
H_0 &: \mathbf{x} = \mathbf{E}_B\beta + \mathbf{n}_0, \\
H_1 &: \mathbf{x} = \mathbf{E}_T\gamma + \mathbf{E}_B\beta + \mathbf{n}_1.
\end{aligned}
$$

MSDinter aims to change the linearity of model (8.1) to take into account the interaction effects between the target and its surrounding background. MSSD and MSCD on the other hand, aim to make additional constraints on the coefficient vectors γ and β. Specifically, MSSD conducts the regularisations on the subspace-representation-based method, i.e. MSD, while MSCD conducts the regularisations on the cone-representation-based linear model. All in all, the proposed three methods focus on improvment of the detection performance.

8.1.3 Cone-representation-based approaches (Chapter 4 and Chapter 7)

For the sake of physical interpretations, HSIs as examples of natural signals have the non-negative properties for both the hyperspectral signature and the abundance coefficients of linear models. Geometrically, the non-negativity constraints on JSM and LMM induce a cone-shape representation. The proposed C-JSM (Chapter 4) and MSCD (Chapter 7) incorporate the non-negativity constraints in the conventional models JSM and LMM for HSI classification and HSI target detection, respectively. C-JSM and MSCD aim to improve the performances of JSM and LMM by giving more reasonable assumptions with respect to physical interpretations.

8.2 Future work

8.2.1 C-JSM based dictionary learning

Despite the dictionaries learned in the baseline method D-KSVD [6] as well as the proposed JSM-DKSVD (Chapter 3) can provide discriminative power for classification, the learned atoms lose the semantic information. Specifically, labels of learned atoms are lost and the resultant synthetic learned atoms cannot be used to represent real hyperspectral pixels, even materials. In addition, the learned atom vectors may also have negative values, which violates the physical properties, i.e. the non-negativity of hyperspectral signals. Therefore it is desirable to learn a dictionary with clear physical meaning, where the atoms have labels and are non-negative. To achieve these goals, the combination of cone-based JSM and the discriminative dictionary learning will be a promising direction.

8.2.2 Extension of MSSD

In the proposed MSSD (Chapter 6), we have incorporated the l_2-norm regularisation in MSD [7] with two implementations: MSSD with isotropic shrinkage and MSSD with anisotropic shrinkage. As with the work we have done in MSCD (Chapter 7), it is interested to investigate the l_1-norm regularised MSD. The work in [9] shows that incorporating the l_1-norm regularisation in fact assumes a multivariate Laplacian prior distribution on the coefficient vectors. The l_1-norm regularised MSD is essentially a sparse-representation version of LMM but in the eigenspace. The sparseness constraints will be incorporated on top on the MSD, which induces an eigenvector-based sparse representation. It is worth studying the detection performance of sparse-based MSD, compared with those of STD [23] and SRBBH [24], which are typical sparse-representation-based methods but adopt the original spectral signatures for dictionaries.

8.2.3 Some other directions

Beyond the topics we discussed in this thesis, we realise that there are several other approaches that can be developed in terms of optimisation problems that we proposed, such as the estimation of dictionary in the JSM-DKSVD model (chapter 3) and the estimations of coefficient vectors in the proposed MSSD (chapter 6) and MSCD (chapter 7). Instead of the constrained optimisation approach that we used to solve the coefficients, the unconstrained optimisation approach is also our interest in the future.

Regarding the statistical distributions involved in the thesis, we have discussed about the prior distributions of coefficient vectors in the proposed MSSD (chapter 6) and MSCD (chapter 7). We are also interested in the distribution of the response in the LMM, i.e. the test HSI pixel $\mathbf{x}$. To study the distribution of the HSI pixels in terms of different classes can help us to understand deeper about the properties of data and therefore to develop better classifier/detector in terms of classification/detection performance.

Bibliography

[1] Dimitris Manolakis, David Marden, and Gary A Shaw. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1):79–116, 2003.

[2] Lefei Zhang, Liangpei Zhang, Dacheng Tao, Xin Huang, and Bo Du. Hyperspectral remote sensing image subpixel target detection based on supervised metric learning. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4955–4965, 2014.

[3] David Snyder, John Kerekes, Ian Fairweather, Robert Crabtree, Jeremy Shive, and Stacey Hager. Development of a web-based application to evaluate target finding algorithms. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 2, pages II–915. IEEE, 2008.

[4] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Hyperspectral image classification using dictionary-based sparse representation. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(10):3973–3985, 2011.

[5] Dimitris Manolakis, Christina Siracusa, and Gary Shaw. Hyperspectral subpixel target detection using the linear mixing model. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(7):1392–1409, 2001.

[6] Qiang Zhang and Baoxin Li. Discriminative K-SVD for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.

[7] Louis L Scharf and Benjamin Friedlander. Matched subspace detectors. *Signal Processing, IEEE Transactions on*, 42(8):2146–2157, 1994.

[8] Joel A Tropp, Anna C Gilbert, and Martin J Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer, Berlin, 2001.

[10] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[11] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[12] David G Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1969.

[13] Dimitris Manolakis, Eric Truslow, Michael Pieper, Thomas Cooley, and Michael Brueggeman. Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms. *Signal Processing Magazine, IEEE*, 31(1):24–33, 2014.

[14] Nasser M Nasrabadi. Hyperspectral target detection: An overview of current and future challenges. *Signal Processing Magazine, IEEE*, 31(1):34–44, 2014.

[15] Stefania Matteoli, Marco Diani, and Giovanni Corsini. A tutorial overview of anomaly detection in hyperspectral images. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):5–28, 2010.

[16] Stefania Matteoli, Marco Diani, and James Theiler. An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2317–2336, 2014.

[17] Daniel C Heinz and Chein-I Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 39(3):529–545, 2001.

[18] Joseph C Harsanyi and Chein-I Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *Geoscience and Remote Sensing, IEEE Transactions on*, 32(4):779–785, 1994.

[19] Qian Du, Hsuan Ren, and Chein-I Chang. A comparative study for orthogonal subspace projection and constrained energy minimization. *IEEE Transactions on Geoscience and Remote Sensing*, 41(6):1525–1529, 2003.

[20] Jeff Settle. On constrained energy minimization and the partial unmixing of multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 40(3):718–721, 2002.

[21] Shawn Kraut and Louis L Scharf. The CFAR adaptive subspace detector is a scale-invariant GLRT. *Signal Processing, IEEE Transactions on*, 47(9):2538–2541, 1999.

[22] Shawn Kraut, Louis L Scharf, and L Todd McWhorter. Adaptive subspace detectors. *Signal Processing, IEEE Transactions on*, 49(1):1–16, 2001.

[23] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Sparse representation for target detection in hyperspectral imagery. *Selected Topics in Signal Processing, IEEE Journal of*, 5(3):629–640, 2011.

[24] Yuxiang Zhang, Bo Du, and Liangpei Zhang. A sparse representation-based binary hypothesis model for target detection in hyperspectral images. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(3):1346–1354, 2015.

[25] Hongyan Zhang, Jiayi Li, Yuancheng Huang, and Liangpei Zhang. A nonlocal weighted joint sparse representation classification method for hyperspectral imagery. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2056–2065, 2014.

[26] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[27] Yuan Yan Tang, Haoliang Yuan, and Luoqing Li. Manifold-based sparse representation for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(12):7606–7618, 2014.

[28] Leyuan Fang, Shutao Li, Xudong Kang, and Jon Atli Benediktsson. Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(8):4186–4201, 2015.

[29] Jiayi Li, Hongyan Zhang, and Liangpei Zhang. Efficient superpixel-level multitask joint sparse representation for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(10):5338–5351, 2015.

[30] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011.

[31] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[32] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[33] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, 2013.

[34] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791–804, 2012.

[35] Ali Soltani-Farani, Hamid R Rabiee, and Seyyed Abbas Hosseini. Spatial-aware dictionary learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):527–541, 2015.

[36] Zhaowen Wang, Nasser M Nasrabadi, and Thomas S Huang. Spatial–spectral classification of hyperspectral images using discriminative dictionary designed by learning vector quantization. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(8):4808–4822, 2014.

[37] Xiaoxia Sun, Nasser M Nasrabadi, and Trac D Tran. Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(8):4457–4471, 2015.

[38] Zhangyang Wang, Nasser M Nasrabadi, and Thomas S Huang. Semisupervised hyperspectral classification using task-driven dictionary learning with Laplacian regularization. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(3):1161–1173, 2015.

[39] Mairal Julien. SPAMS toolbox. `http://spams-devel.gforge.inria.fr/`.

[40] Purdue Research Foundation. A freeware multispectral image data analysis system. `https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html`, 2014. [Online; accessed 22-July-2014].

[41] John A Richards and Xiuping Jia. *Remote Sensing Digital Image Analysis: An Introduction*. New York: Springer-Verlag, 2006.

[42] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[43] Ziyu Wang, Jianxiong Liu, and Jing-Hao Xue. Joint sparse model-based discriminative K-SVD for hyperspectral image classification. *Signal Processing*, 133:144–155, 2017.

[44] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Hyperspectral image classification via kernel sparse representation. *IEEE Transactions on Geoscience and Remote sensing*, 51(1):217–231, 2013.

[45] Jianjun Liu, Zebin Wu, Zhihui Wei, Liang Xiao, and Le Sun. Spatial-spectral kernel sparse representation for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(6):2462–2471, 2013.

[46] V Paul Pauca, Jon Piper, and Robert J Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and Its Applications*, 416(1):29–47, 2006.

[47] Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, 2007.

[48] Xuesong Liu, Wei Xia, Bin Wang, and Liming Zhang. An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data.

IEEE Transactions on Geoscience and Remote Sensing, 49(2):757–772, 2011.

[49] Zuyuan Yang, Guoxu Zhou, Shengli Xie, Shuxue Ding, Jun-Mei Yang, and Jun Zhang. Blind spectral unmixing based on sparse nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 20(4):1112–1125, 2011.

[50] Junmin Liu, Jiangshe Zhang, Yuelin Gao, Chunxia Zhang, and Zhihua Li. Enhancing spectral unmixing by local neighborhood weights. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5):1545–1552, 2012.

[51] Nan Wang, Bo Du, and Liangpei Zhang. An endmember dissimilarity constrained non-negative matrix factorization method for hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(2):554–569, 2013.

[52] Cédric Févotte and Nicolas Dobigeon. Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 24(12):4810–4819, 2015.

[53] Charles Lawson and Richard Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[54] Rasmus Bro and Sijmen De Jong. A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401, 1997.

[55] Alfred M Bruckstein, Michael Elad, and Michael Zibulevsky. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, 54(11):4813–4820, 2008.

[56] Mehrdad Yaghoobi, Di Wu, and Mike E Davies. Fast non-negative orthogonal matching pursuit. *IEEE Signal Processing Letters*, 22(9):1229–1233, 2015.

[57] Qian Shi, Bo Du, and Liangpei Zhang. Spatial coherence-based batch-mode active learning for remote sensing image classification. *IEEE Transactions on Image Processing*, 24(7):2037–2050, 2015.

[58] Mark H Van Benthem and Michael R Keenan. Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics*, 18(10):441–450, 2004.

[59] Dany Leviatan and Vladimir N Temlyakov. Simultaneous approximation by greedy algorithms. *Advances in Computational Mathematics*, 25(1-3):73–90, 2006.

[60] Shane F Cotter, Bhaskar D Rao, Kjersti Engan, and Kenneth Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, 53(7):2477–2488, 2005.

[61] Jiayi Li, Hongyan Zhang, Yuancheng Huang, and Liangpei Zhang. Hyperspectral image classification by nonlocal joint collaborative representation with a locally adaptive dictionary. *IEEE Transactions on Geoscience and Remote Sensing*, 52(6):3707–3719, 2014.

[62] Martin Slawski and Matthias Hein. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

[63] Dimitris G Manolakis, Gary A Shaw, and Nirmal Keshava. Comparative analysis of hyperspectral adaptive matched filter detectors. In *AeroSense 2000*, pages 2–17. International Society for Optics and Photonics, 2000.

[64] Daniel R Fuhrmann, Edward J Kelly, and Ramon Nitzberg. A CFAR adaptive matched filter detector. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(1):208–216, 1992.

[65] Heesung Kwon and Nasser M Nasrabadi. A comparative analysis of kernel subspace target detectors for hyperspectral imagery. *EURASIP Journal on Applied Signal Processing*, 2007(1):193–193, 2007.

[66] Shuo Yang, Zhenwei Shi, and Wei Tang. Robust hyperspectral image target detection using an inequality constraint. *IEEE Transactions on Geoscience and Remote Sensing*, 53(6):3389–3404, 2015.

[67] Zhengxia Zou and Zhenwei Shi. Hierarchical suppression method for hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):330–342, 2016.

[68] Shuo Yang and Zhenwei Shi. Hyperspectral image target detection improvement based on total variation. *IEEE Transactions on Image Processing*, 25(5):2249–2258, 2016.

[69] Yi Chen, Nasser M Nasrabadi, and Trac D Tran. Kernel sparse representation for hyperspectral target detection. In *SPIE Defense, Security, and Sensing*, pages 839005–839005–9. International Society for Optics and Photonics, 2012.

[70] Yuxiang Zhang, Liangpei Zhang, Bo Du, and Shugen Wang. A nonlinear sparse representation-based binary hypothesis model for hyperspectral target detection. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 8(6):2513–2522, 2015.

[71] Wei Li and Qian Du. A survey on representation-based classification and detection in hyperspectral remote sensing imagery. *Pattern Recognition Letters*, 83:115–123, 2016.

[72] Wei Li, Qian Du, and Bing Zhang. Combined sparse and collaborative representation for hyperspectral target detection. *Pattern Recognition*, 48(12):3904–3916, 2015.

[73] José MP Nascimento and José M Bioucas-Dias. Nonlinear mixture model for hyperspectral unmixing. In *SPIE Europe Remote Sensing*, pages 74770I–74770I. International Society for Optics and Photonics, 2009.

[74] Wenyi Fan, Baoxin Hu, John Miller, and Mingze Li. Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data. *International Journal of Remote Sensing*, 30(11):2951–2962, 2009.

[75] Abderrahim Halimi, Yoann Altmann, Nicolas Dobigeon, and Jean-Yves Tourneret. Nonlinear unmixing of hyperspectral images using a generalized bilinear model. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(11):4153–4162, 2011.

[76] Terrill W Ray and Bruce C Murray. Nonlinear spectral mixing in desert vegetation. *Remote sensing of environment*, 55(1):59–64, 1996.

[77] Xuexia Chen and Lee Vierling. Spectral mixture analyses of hyperspectral data acquired using a tethered balloon. *Remote Sensing of Environment*, 103(3):338–350, 2006.

[78] Ben Somers, Kenneth Cools, Stephanie Delalieux, Jan Stuckens, Dimitry Van der Zande, Willem W Verstraeten, and Pol Coppin. Nonlinear hyperspectral mixture analysis for tree cover estimates in orchards. *Remote Sensing of Environment*, 113(6):1183–1193, 2009.

[79] Rob Heylen and Paul Scheunders. A multilinear mixing model for nonlinear spectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):240–251, 2016.

[80] Bo Du and Liangpei Zhang. Target detection based on a dynamic subspace. *Pattern Recognition*, 47(1):344–358, 2014.

[81] Lefei Zhang, Liangpei Zhang, Dacheng Tao, and Xin Huang. Sparse transfer manifold embedding for hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1030–1043, 2014.

[82] Lianru Gao, Bin Yang, Qian Du, and Bing Zhang. Adjusted spectral matched filter for target detection in hyperspectral imagery. *Remote Sensing*, 7(6):6611–6634, 2015.

[83] Wei Li and Qian Du. Collaborative representation for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, 53(3):1463–1474, 2015.

[84] Yuxiang Zhang, Bo Du, Liangpei Zhang, and Shugen Wang. A low-rank and sparse matrix decomposition-based Mahalanobis distance method for hyperspectral anomaly detection. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1376–1389, 2016.

[85] Heesung Kwon and Nasser M Nasrabadi. A comparative analysis of kernel subspace target detectors for hyperspectral imagery. *EURASIP Journal on Advances in Signal Processing*, 2007(1):1–13, 2006.

[86] Yanfeng Gu, Yuting Wang, He Zheng, and Yue Hu. Hyperspectral target detection via exploiting spatial-spectral joint sparsity. *Neurocomputing*, 169:5–12, 2015.

[87] Andrei Nikolaevich Tikhonov, AV Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical Methods for the Solution of Ill-posed Problems*, volume 328. Springer Science & Business Media, 2013.

[88] Wei Li, Qian Du, and Mingming Xiong. Kernel collaborative representation with Tikhonov regularization for hyperspectral image classification. *Geoscience and Remote Sensing Letters, IEEE*, 12(1):48–52, 2015.

[89] Yuxiang Zhang, Bo Du, and Liangpei Zhang. Regularization framework for target detection in hyperspectral imagery. *Geoscience and Remote Sensing Letters, IEEE*, 11(1):313–317, 2014.

[90] Nasser M Nasrabadi. Regularized spectral matched filter for target recognition in hyperspectral imagery. *Signal Processing Letters, IEEE*, 15:317–320, 2008.

[91] Steven M Kay. *Fundamentals of Statistical Signal Processing: Detection Theory, Vol. 2*. Prentice Hall, NJ, USA, 1998.

[92] Bo Du, Liangpei Zhang, Dacheng Tao, and Dengyi Zhang. Unsupervised transfer learning for target detection from hyperspectral images. *Neurocomputing*, 120:72–82, 2013.

[93] Ting Wang, Bo Du, and Liangpei Zhang. A background self-learning framework for unstructured target detectors. *IEEE Geoscience and Remote Sensing Letters*, 10(6):1577–1581, 2013.

[94] Bo Du, Yuxiang Zhang, Liangpei Zhang, and Dacheng Tao. Beyond the sparsity-based target detector: A hybrid sparsity and statistics-based detector for hyperspectral images. *IEEE Transactions on Image Processing*, 25(11):5345–5357, 2016.

[95] Ming Yuan and Yi Lin. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

[96] Martin Slawski and Matthias Hein. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

[97] Rafal Zdunek. Regularized NNLS algorithms for nonnegative matrix factorization with application to text document clustering. In *Computer Recognition Systems 4*, pages 757–766. Springer, 2011.

[98] Ernie Esser, Yifei Lou, and Jack Xin. A method for finding structured sparse solutions to nonnegative least squares problems with applications. *SIAM Journal on Imaging Sciences*, 6(4):2010–2046, 2013.

[99] Dennis Aigner, CA Knox Lovell, and Peter Schmidt. Formulation and estimation of stochastic frontier production function models. *Journal of Econometrics*, 6(1):21–37, 1977.

[100] Yucui Liu and Tomasz J Kozubowski. A folded Laplace distribution. *Journal of Statistical Distributions and Applications*, 2(1):1, 2015.